PSAP Simulator Software Requirements Specification

For Software Version 1.0.0

Table of Contents

1 Introduction

This document is the Software Requirements Specification (SRS) for the PSAP Simulator application.

The PSAP Simulator application is a test program. The intended uses of this application are:

1. Assist in interoperability testing of Next Generation 9-1-1 (NG9-1-1) call Emergency Services IP Network (ESInet) functional elements that deliver NG9-1-1 calls to NG9-1-1 capable Public Safety Answering Points (PSAPs).
2. Provide a way to perform integration testing of the various NG9-1-1 interfaces that have been implemented in the SipLib, Ng911Lib, EidoLib and Ng911CadIfLib open source class libraries.
3. Provide a proof of concept implementation of the less commonly implemented interfaces specified in the most recent version of the NENA i3 Standard for Next Generation 9-1-1 Standard (NENA-STA-010.3b [1]).

This application shall be a simplified PSAP call handling functional element. It will be a single call taker position application that can handle multiple calls simultaneously, but the call taker can only communicate with a single caller at a time. There will be no centralized PSAP call controller so functions such as automatic call distribution, call queue pickup, call takeover, barge-in, local transfers (within the same PSAP), local conferences, administrative (non-emergency, i.e. an interface to an agency's PBX) call handling and other functions that are normally expected in a PSAP application will not be available.

1.1 References

This section lists the standards to be used for the development of this application by category. The fact that a standard is listed as a reference here does not imply that all requirements in that standard will be implemented. Sections 2, 3 and 4 present specific requirements from these standards that shall be implemented.

1.1.1 NENA Standards

1. NENA i3 Standard for Next Generation 9 1 1, NENA, NENA STA-010.3f-2021, October 7, 2021.
2. NENA Standard for Emergency Incident Data Object (EIDO), NENA, NENA-STA-021.1a-2022, April 19, 2022.
3. NENA Standard for the Conveyance of Emergency Incident Data Objects (EIDOs) between Next Generation (NG9-1-1) Systems and Applications, NENA, NENA-STA-024.1a-2023, January 10, 2023.

1.1.2 General SIP RFCs

4. SIP: Session Initiation Protocol, IETF, RFC 3261, June 2002.
5. Reliability of Provisional Responses in the Session Initiation Protocol (SIP), IETF, RFC 3262, June 2002.
6. An Offer/Answer Model with the Session Description Protocol (SDP), IETF, RFC 3264, June 2002.
7. A Presence Event Package for the Session Initiation Protocol (SIP), IETF, RFC 3856, August 2004.
8. Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP), IETF, RFC 3605, October 2003.

9. An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing, IETF, RFC 3581, August 2003.
10. The Session Initiation Protocol Replaces Header, IETF, RFC 3891, September 2004.
11. Session Initiation Protocol (SIP)-Specific Event Notification, IETF, RFC 6665, July 2012.
12. A Session Initiation Protocol (SIP) Event Package for Conference State, IETF, RFC 4575, August 2006.
13. SDP : Session Description Protocol, IETF, RFC 4566, July 2006.
14. RTP: A Transport Protocol for Real-Time Applications, IETF, RFC 3550, July 2003.
15. RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals, IETF, RFC 4733, December 2006.
16. The Session Initiation Protocol (SIP) Refer Method, IETF, RFC 3515, 2003.
17. Session Initiation Protocol (SIP) Call Control – Conferencing for User Agents, IETF, RFC 4579, August 2006.
18. The Session Description Protocol (SDP) Label Attribute, IETF, RFC 4575, August 2006.

1.1.3 NG9-1-1 Specific RFCs/Standards

19. Additional Data Related to an Emergency Call, IETF, RFC 7852, July 2016.
20. Framework for Emergency Calling Using Internet Multimedia, IETF, RFC 6443, December 2011.
21. Best Current Practice for Communications Services in Support of Emergency Calling, IETF, RFC 6881, March 2013.
22. Location Conveyance for the Session Initiation Protocol, IETF, RFC 6442, December 2011.
23. Next-Generation Pan-European eCall, IETF, RFC 8147, May 2017.
24. Next-Generation Vehicle-Initiated Emergency Calls, IETF, RFC 8148, May 2017.
25. Advanced Automatic Collision Notification (AACN) Vehicle Emergency Data Set (VEDS), NENA/APCO, APCO/NENA Candidate ANS 2.102.1.2022.
26. Common Alerting Protocol Version 1.2, OASIS, July 2010.
27. HTTP-Enabled Location Delivery (HELD), IETF, RFC 5985, September 2010.
28. Use of Device Identity in HTTP-Enabled Location Delivery (HELD), IETF, RFC 6155, March 2011.
29. LoST: A Location-to-Service Translation Protocol, IETF, RFC 5222, August 2008.
30. Negotiating Human Language in Real-Time Communications, IETF, RFC 8373, May 2018.
31. An Extension to the Session Description Protocol (SDP) and Real-time Transport Protocol (RTP) for Media Loopback, IETF, RFC 6849, February 2013.
32. Non-interactive Emergency Calls, IETF, RFC 8876, September 2020.

1.1.4 Security Related RFCs

33. The Secure Real-time Transport Protocol (SRTP), IETF, RFC 3711, March 2004.
34. Session Description Protocol (SDP) Security Descriptions for Media Streams, IETF, RFC 4568, July 2006.
35. The Use of AES-192 and AES-256 in Secure RTP, IETF, RFC 6188, March 2011.
36. Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS), IETF, RFC 5763, May 2010.
37. Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP), IETF, RFC 5764, May 2010.

1.1.5 MSRP RFCs

38. The Message Session Relay Protocol (MSRP), IETF, RFC 4975, September 2007.
39. Multi-party Chat Using the Message Session Relay Protocol (MSRP), IETF, RFC 7701, December 2015.
40. Common Profile for Instant Messaging (CPIM), IETF, RFC 3860, August 2004.
41. TCP-Based Media Transport in the Session Description Protocol (SDP), IETF, RFC 4145, September 2005.
42. An Alternative Connection Model for the Message Session Relay Protocol (MSRP), IETF, RFC 6135, February 2011.

1.1.6 RTT RFCs

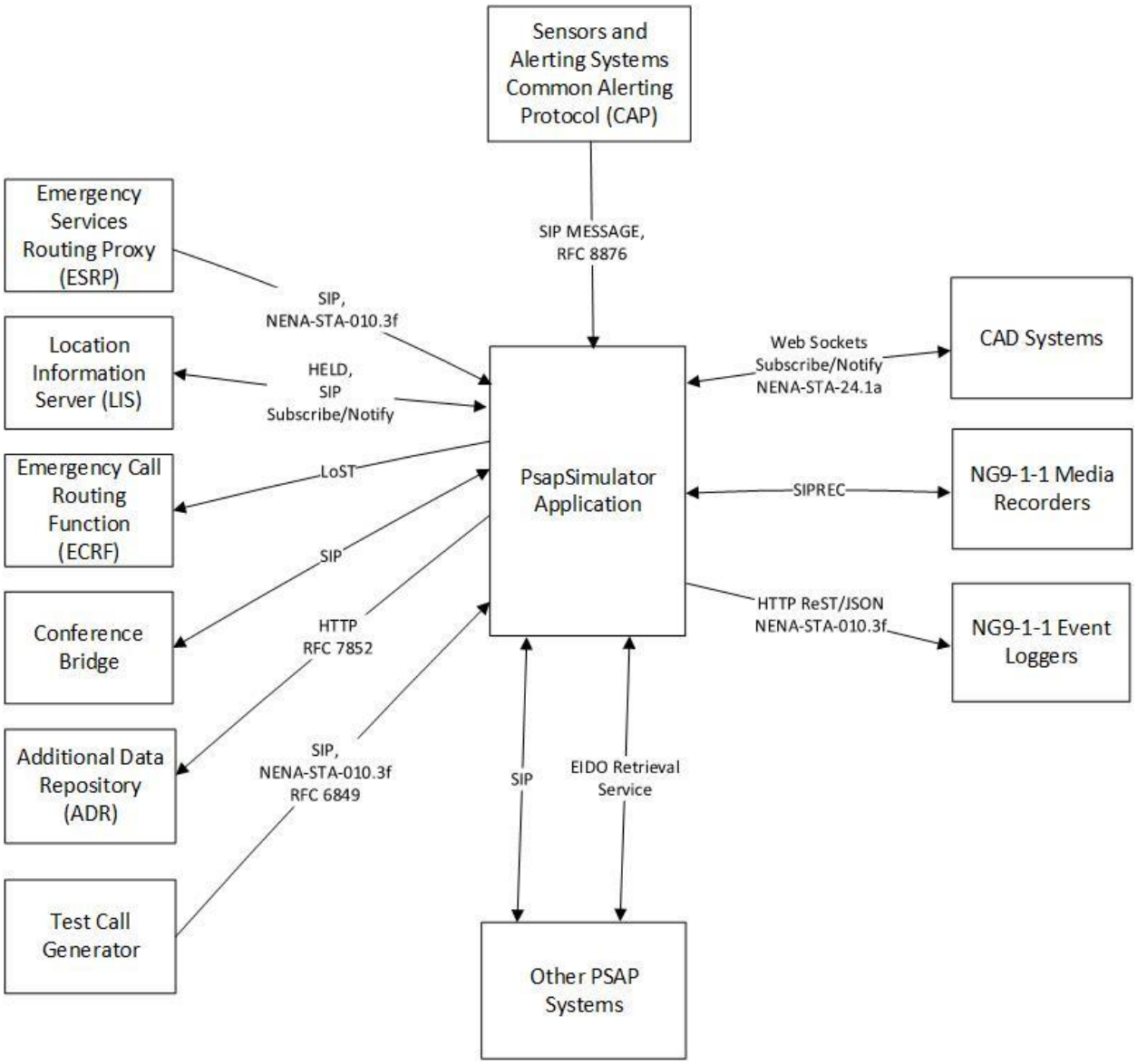
43. RTP Payload for Text Conversation, IETF, RFC 4103, June 2005.
44. RTP-Mixer Formatting of Multiparty Real-Time Text, IETF, RFC 9071, July 2021.

1.1.7 SIPREC RFCs

45. An Architecture for Media Recording Using the Session Initiation Protocol, IETF, RFC 7245, May 2014.
46. Session Initiation Protocol (SIP) Recording Metadata, IETF, RFC 7865, May 2016.
47. Session Recording Protocol, IETF, RFC 7866, May 2016.
48. Session Initiation Protocol (SIP) Recording Call Flows, IETF, RFC 8068, February 2017.

2 General Requirements

The following figure shows the functional elements that the PsapSimulator application will interface to.



2.1 Operational Environment

2.1.1 Windows Operating Systems

The application shall be designed to work on Windows 10 Professional or later.

The application is not required to run on the Home editions of the Windows operating systems.

2.1.2 Minimum PC Requirements

TBD.

2.1.3 Application Installation Requirements

The basic installation technology shall be a Windows setup MSI package. The setup program shall automatically install any prerequisites (such as .NET).

A self-extracting EXE installation program shall be developed.

The setup program shall configure Windows to run this application as a Windows administrator.

The installation program shall install a short icon on the user's desktop.

2.2 Network Configurations

The application shall be capable of operating in the following network environments.

1. IPv4 and IPv6
2. IPv4 only
3. IPv6 only

The network configuration shall be configurable.

For each network type, the application shall listen on a single configurable IP address.

2.3 SIP Transport Protocols

The application shall be capable of listening on one or all of the following SIP transport protocols.

1. UDP
2. TCP
3. TLS

The transport protocol selections shall be configurable.

2.4 Media Support

The application shall support the following media types.

1. Voice
2. Video
3. Real Time Text (RTT)
4. Message Session Relay Protocol (MSRP)

2.4.1 Media Security Requirements

The application shall support media encryption using SDES-SRTP and DTLS-SRTP for all media types that are transported using RTP (audio, RTT and video).

The application shall support MSRP media using Transport Layer Security (TLS).

2.4.1.1 Security Descriptors for SRTP (SDES-SRTP)

The application shall support Secure RTP using security descriptors as specified in RFC 3711 and RFC 4568.

The application shall support the following crypto-suites.

1. AES_CM_128_HMAC_SHA1_80 (RFC 4568)
2. AES_CM_128_HMAC_SHA1_32 (RFC 4568)
3. F8_128_HMAC_SHA1_80 (RFC 4568)
4. AES_192_CM_HMAC_SHA1_80 (RFC 6188)
5. AES_192_CM_HMAC_SHA1_32 (RFC 6188)
6. AES_256_CM_HMAC_SHA1_80 (RFC 6188)
7. AES_256_CM_HMAC_SHA1_32 (RFC 6188)

If the caller offers multiple SDES-SRTP crypto-suites, the application shall pick the most secure crypto suite in the answer SDP that it responds with.

If the caller offers SDES-SRTP but does not offer a crypto-suite that the application supports, it shall answer with an SDP response that does not use media encryption.

The application shall have a configuration setting that determines if it will offer SDES-SRTP for outgoing calls. If configured to offer SDES-SRTP encryption, the application shall offer the following crypto suites for each RTP media type that it offers.

1. AES_CM_128_HMAC_SHA1_80 (RFC 4568)
2. AES_256_CM_HMAC_SHA1_80 (RFC 6188)

2.4.1.2 Datagram Transport Layer Security for SRTP (DTLS-SRTP)

The application shall support DTLS-SRTP as specified in RFC 5763 and RFC 5764, as specified in Section 3.1.9 of NENA-STA-010.3b.

The application shall support the following cryptographic profiles for DTLS-SRTP (see Section 4.1.2 of RFC 5764).

1. SRTP_AES128_CM_HMAC_SHA1_80
2. SRTP_AES128_CM_HMAC_SHA1_32
3. SRTP_NULL_HMAC_SHA1_80
4. SRTP_NULL_HMAC_SHA1_32

If the application is the server for an incoming INVITE request, it shall always assume the role of the DTLS-SRTP server. For outgoing calls, the application shall always assume the role of the DTLS-SRTP client.

The application shall have a configuration setting that determines if it will offer DTLS-SRTP for outgoing calls. If configured to offer DTLS-SRTP, the application shall offer the following cryptographic profile in the outgoing INVITE request for each for each RTP media type that it offers.

1. SRTP_AES128_CM_HMAC_SHA1_80

2.4.1.3 MSRP over TLS

The application shall support MSRP over TLS (MSRPS).

For outgoing calls, the application shall have a configuration setting that specifies whether to offer MSRPS for outgoing calls.

2.4.2 Audio and Video Codecs

The application shall support the following audio codecs.

1. G.711 Mu-Law
2. G.711 A-Law
3. G.722
4. G.729
5. AMR-WB

The application shall support the following video codecs.

1. H.264
2. VP8

2.4.3 MSRP Connection Mode Requirements

The application shall accept the active, passive and active/passive connection modes for incoming MSRP calls as specified in RFC 4145 and RFC 6135.

For outgoing calls, there shall be a configuration setting that determines the connection mode to offer. The default setting shall be active.

2.4.4 Quality of Service DSCP Requirements for Media

The application shall apply the Quality of Service (QOS) Differentiated Service Code Point IP packet markings to outgoing media packets (audio, video, RTT and MSRP) as specified in Section 2.7 of NENA-STA-010.3b.

This requirement applies to both IPv4 and IPv6 networks.

This requirement applies to the following types of calls.

1. Incoming calls from the ESInet
2. Outgoing callback calls made via the OCIF.
3. SIPREC calls to media recorders.

2.4.5 Language Support

As specified in Section 4.6.2 of NENA-STA-010.3b, the application shall support the language negotiation SDP attributes defined in RFC 8373. These SDP media attributes are called “hlang-send” and “hlang-recv” and indicate the languages that the caller is capable of communicating with.

The application shall accept all languages offered for each media type in the response to an INVITE request. It shall be capable of displaying the offered languages to the call taker.

3 NG9-1-1 Functional Element Interfaces

Section 4.6 of NENA-STA-010.3b specifies which functional element interfaces that the PSAP call handling functional element must support.

The following table specifies which interfaces and the degree of support that this application must meet. The degree of support is indicated in the "Supported?" column.

NENA-STA-010.3b Section	Supported?	Related Section
4.6.1 SIP Call Interface	Full support for incoming calls and outgoing callback calls	3.1
4.6.2 Media	Full support	2.4
4.6.3 LoST Interface	Full support	3.2
4.6.4 LIS Interfaces	Full support	3.3
4.6.5 Bridge Interface	No. This section of NENA-STA-010.3b states that the PSAP MAY provide its own conference bridge. The application shall support call conferencing and call transfers via an external conference aware user agent (i.e. a conference bridge).	3.13
4.6.6 Element State	Full support	3.4
4.6.7 Service State	Full support	3.5
4.6.8 Abandoned Call Event	Future	
4.6.9 De-queue Registration	Full support	3.6
4.6.10 Queue State	Full support	3.7
4.6.11 SI	No. Support for the Spatial Interface of a GIS server is optional in NENA-STA-010.3b.	
4.6.12 Logging Service	Full Support	3.9
4.6.13 Security Posture	Full Support	
4.6.14 Policy	No. Support for the Policy Store is optional in NENA-STA-010.3b.	
4.6.15 Additional Data Dereference	Full Support	
4.6.16 Time Interface	Yes, via the Windows NTP Interface	
4.6.17 Test Call	Full Support	3.10
4.6.18 Testing of Policy Rules	No. Support for this function appears to be optional in NENA-STA-010.3b.	
4.6.19 Call Diversion	Yes, because De-Queue Registration will be supported.	
4.6.20 Incidents	Incident merging is deemed to be a Computer Aided Dispatch (CAD) function and it does not need to be supported by this application.	

3.1 SIP Call Interface

The application shall support the Session Initiation Protocol (SIP) for incoming NG9-1-1 calls and for performing callback calls. The application also needs to support an outgoing SIP interface to SIPREC media recorders.

The following table specifies which Session Initiation Protocol (SIP) methods the application must support.

SIP Method	NENA-STA-010.3b Requirement	Supported?	Comments
INVITE	Mandatory	Yes	
REFER	Mandatory	Yes	
BYE	Mandatory	Yes	
CANCEL	Mandatory	Yes	
UPDATE	Mandatory	Yes	Used for sending updated SIPREC metadata to SIPREC recorders only. Not used for incoming calls.
OPTIONS	Mandatory	Yes	
ACK	Mandatory	Yes	
PRACK	Mandatory	No	Future
MESSAGE	Mandatory	Yes	Used for incoming non-interactive Common Alerting Protocol (CAP) calls only.
INFO	Mandatory (but must not be used to convey DTMF digits)	No	The INFO method is used in two cases in NENA-STA-010.3b. It is used by a Legacy Network Gateway (LNG) and for requesting Intra-Frame refresh request (RFC 5168) for video media. This application does not need to support any LNG functions and RFC 5168 deprecates this method of frame refresh as specified in the Abstract of this RFC is currently deprecated.
REGISTER	Must not support	No	
SUBSCRIBE	Mandatory	Yes	
NOTIFY	Mandatory	Yes	
PUBLISH	Optional	No	Not used for NG9-1-1 applications

3.1.1 Support for re-INVITE Requests

The application shall handle re-INVITE requests for calls that it is currently handling. Incoming re-INVITE requests may be received in the following situations.

1. An upstream element changes the media destination endpoints when a call is added to a conference
2. An upstream element wishes to change the media state or add media to the call

The application shall be able to send outgoing re-INVITE requests for calls that are in the on-line state in order to add media to the call.

3.1.2 Offer-Less INVITE Requests

The application shall be able to handle incoming INVITE requests for new calls that do not contain an SDP offer in the body of the request. This type of INVITE request is called an offer-less INVITE. In this

case, the application shall send its SDP offer in the OK response that it sends and the caller shall send its SDP answer in the ACK request that it sends.

An offer-less INVITE may contain PIDF-LO location data or additional data either by-value or by reference.

3.1.3 Outbound Call Interface

The application shall allow the user to make outbound calls. Outbound calls are required in the following situations.

1. The user selects an incoming call from the call history list and performs a callback as described in Sections 4.6.1 and 4.20 of NENA-STA-010.3b.
2. The user wishes to call another agency or device.

When the user initiates a callback call or an outgoing call, the application shall provide ring sound.

3.1.3.1 Callback Call Requirements

The user shall be able to select a call from a call history list and perform a callback call. The callback call function shall be available only for incoming calls.

When performing a callback call to a caller from the call history list, the application shall offer the media that was used (or offered) in the original call.

Callback calls shall be made via a configured Outbound Call Interface Function (OCIF server).

When forming the SIP request URI and the To header URI, the application shall use the user part of the From header of the original call and the host information of the configured OCIF.

Note: This is a simplification of the methods suggested in Sections 4.6.1 and 4.20 of NENA-STA-010.3b.

3.1.3.2 Outgoing Call Requirements

When performing an outgoing call, the user shall be able to enter the SIP URI to route the call to.

The default host portion of the destination shall be the host portion of the configured OCIF if one is configured.

The software shall remember the last used SIP URI across application sessions.

When performing a simple outgoing call, the application shall offer the media that it is configured for.

3.1.4 Quality of Service DSCP Requirements for SIP

The application shall apply the Quality of Service (QOS) Differentiated Service Code Point IP packet markings to outgoing SIP packets as specified in Section 2.7 of NENA-STA-010.3b.

This requirement applies to both IPv4 and IPv6 networks.

This requirement applies to the following types of calls.

1. Incoming calls from the ESInet
2. Outgoing callback calls made via the OCIF.
3. SIPREC calls to media recorders.

3.2 LoST Client Interface

See Section 4.6.3 of NENA-STA-010.3b. The application shall support the client LoST interface as described in that section.

The user shall be able to initiate a query to an ECRF to determine the responder to which to transfer a call to. In this case, the user shall be able to select a subtype of the urn:emergency:service:responder URN and send a request to an ECRF with the location of the current call. The ECRF will then provide a SIP URI in the LoST response that the user can use to conference or transfer the call to. The user shall be able to select one of the URN subtypes specified in Sections 10.5, 10.6, 10.7, 10.8 and 10.9 of NENA-STA-010.3b.

The user shall also be able to send an agency locator request to the ECRF in order to determine an agency to conference or transfer the current call to. When the user initiates an agency locator request, the application shall send a LoST request to a configured ECRF with the location of the current call. When the ECRF returns a response with a URI, the application shall retrieve the agency locator record by performing an HTTPS GET request to the URI. The agency locator record will contain the SIP URI of the agency to transfer the current call to.

The application shall allow the user to configure a URI for the ECRF.

3.3 LIS Interfaces

The application shall support geolocation by value or by reference by supporting the SIP Geolocation header as specified in RFC 6442. The application shall support the following types of URIs in the Geolocation header.

1. CID scheme URI (location by value)
2. SIP URI (location by reference using the SIP Presence Event Package (RFC 3856)
3. HTTPS URI (location by reference using HELD, RFC 5985)

The application shall be capable of handling multiple Geolocation headers containing any combination of the above URI schemes.

If the application receives an INVITE request with location by value then it shall use the location provided in the body of the INVITE request as the initial location of the caller for display purposes.

If the INVITE request contains a Geolocation header with a SIP URI, it shall immediately subscribe to the SIP Presence Event package.

If the INVITE request contains a Geolocation header with an HTTP(s) URI, it shall immediately perform a HELD request to the LIS. The application shall set the ResponseTime attribute in the location request to “emergencyDispatch”. Note: It may require the LIS several seconds to respond an emergency dispatch request.

The above location dereferencing requirements apply only to calls that the application does not automatically reject.

3.4 Element State Interface

The application shall implement the notifier side of the ElementState event package. See Section 2.4.1 of NENA-STA-010.3b.

The application shall accept SIP SUBSCRIBE requests for the ElementState event package from multiple subscribers.

The user shall be able to set the current element state to any of the values listed in Section 10.13 of NENA-STA-010.3b.

When the user changes the element state setting, the application shall send NOTIFY SIP requests to all subscribers containing the new element state.

3.5 Service State Interface

The application shall implement the notifier side of the ServiceState event package. See Section 2.4.2 of NENA-STA-010.3b.

The application shall accept SIP SUBSCRIBE requests for the ServiceState event package from multiple subscribers.

The user shall be able to set the current service state to any of the values listed in Section 10.12 of NENA-STA-010.3b.

As part of the service state interface, the user shall be able to set the security posture to any of the values specified in 10.18 of NENA-STA-010.3b.

When the user changes the service state or security posture settings, the application shall send NOTIFY SIP requests to all subscribers containing the new service state.

3.6 De-Queue Registration

The application shall implement the client-side of the De-Queue Registration service of an ESRP. See Section 4.2.1.4 of NENA-STA-010.3b.

The user shall be able to define multiple queues that the application will register on. Each queue shall be identified by a Queue URI which is a SIP URI. The user shall be able to add, edit or delete queues.

The user shall be able to specify the HTTP(s) URI of an ESRP that the application will register with.

The user shall be able to enable or disable the de-queue registration function. If the de-queue registration is enabled then the application will send de-queue registration requests to the ESRP for each configured queue when it starts up.

If the de-queue registration function is enabled then the user must configure at least one queue to register on.

If the application is unable to register with the ESRP then it shall attempt to register every 5 seconds.

The user shall be able to configure the SIP URI of the ESRP to send de-queue registration requests to.

The user shall be able to configure a list of queue URIs that it will register on if the de-queue registration function is enabled. The user shall be able to add, edit or delete queue URIs.

The user shall be able to view the registration status.

3.7 Queue State Interface

The application shall implement the server-side (the notifier) of the QueueState interface. See Section 4.2.1.3 of NENA-STA-010.3b.

The application shall accept SUBSCRIBE requests to the QueueState event package from multiple subscribers.

The user shall be able to set the queue state to any of the values listed in Section 10.17 of NENA-STA-010.3b.

The application shall send a NOTIFY request to all subscribers when the user changes the queue state or the queue length changes.

3.8 Call Related SIP Subscriptions

3.8.1 Presence Event Package

The application shall support the subscriber side of the A Presence Event Package for the Session Initiation Protocol (SIP) (RFC 3856).

If the incoming INVITE request contains a Geolocation header containing a SIP or a SIPS URI, then the application shall automatically subscribe to the presence event for that call.

When the application receives a NOTIFY request containing location data then the application shall update the caller's location information for the call.

The application shall maintain this subscription for the duration of the call.

3.8.2 Conference Event Package

The application shall support the subscriber side of the Session Initiation Protocol (SIP) Event Package for Conference State, IETF, RFC 4575

If the incoming INVITE request contains a Contact header with an "isfocus" parameter, then the application shall automatically subscribe to the conference event for that call.

The application shall maintain the subscription to the conference event package for the duration of the call.

The application shall subscribe to the conference event package when the call is auto-answered or initially answered by the application user.

3.8.3 Refer Event Package

The application shall support the refer event package described in Section 3 of RFC 3515.

When the application initiates a REFER request to a conference bridge to add another participant to a call, the conference bridge automatically creates an implied subscription to the refer event packages. The conference bridge sends NOTIFY request to the sender of the REFER request to notify it of the status of the refer operation. The body of the NOTIFY request contains a fragment of a SIP message (SIPFRAG, Content-Type = message/sipfrag) that describes the status of the refer operation. Section 2.4.5 of RFC 3515. The application shall use this information to inform the user of success or failure of the REFER request.

3.9 NG9-1-1 Logging Service

The application shall support active media recording and NENA NG9-1-1 event logging. The application shall be able to interface to at least two media and two event logging servers.

3.9.1 Media Recoding (SIPREC) Requirements

The application shall be capable of recording all media for answered calls. The application shall support the SIP Recording Client interface as specified in RFC 7866. This interface is commonly known as SIPREC.

The application shall provide the recording metadata with the INVITE request that is sends to SIP Recording Servers (SRS) as specified in RFC 7865. When a party is removed or added to a conference, the application shall provide a full, updated version of the SIPREC metadata using a SIP UPDATE request.

The application shall be capable of recording all media types (voice, video, RTT and MSRP).

The application shall monitor the connection state of the configured SIPREC media recorders. The user shall be able to view the current connection state of each configured SIPREC media recorder.

The user shall be able to enable or disable SIPREC media recording while the application is running. Changing the enabled/disabled state of media recording shall not affect the SIPREC media recording configuration setting.

The user shall be able to enable or disable each configured SIPREC media recorder individually. Changing the enabled/disabled state of an individual SIPREC media recorder shall not affect the configuration setting of that media recorder.

3.9.1.1 SIPREC Media Recording Configuration Settings

The application shall provide the following configuration settings for SIPREC media recording.

1. Enable/Disable Media Recording (master setting)
2. A list of SIPREC media recorders

The user shall be able to add, edit or remove SIPREC media recorders.

The application shall provide the following configuration settings for each SIPREC media recorder.

1. Recorder name
2. Recorder endpoint (IP address and SIP port number)
3. SIP Transport (UDP, TCP, TLS)
4. Local SIP port
5. Enable Media SRTP encryption

The application shall support IPv4 and IPv6 for SIPREC media recorders.

3.9.2 Event Logging Requirements

The application shall support the client side of the NG9-1-1 event logging interface as described in Section 4.12.3 and Appendix E.8 of NENA-STA-010.3b.

The application shall monitor the connection status of each configured event logger. The user shall be able to view the connection status of each event logger.

The user shall be able to enable or disable event logging while the application is running. Changing the enabled/disabled state of event logging shall not affect the global event logging enable/disable setting.

The user shall be able to enable or disable individual event loggers while the application is running. Changing the enabled/disabled state of an individual event logger shall not affect the configuration setting for that event logger.

The application shall log the following events that are defined in Section 4.12.3.7 of NENA-STA-010.3b.

1. CallStartLogEvent
2. RecCallStartLogEvent
3. CallEndLogEvent
4. RecCallEndLogEvent
5. CallTransferLogEvent
6. MediaStartLogEvent
7. MediaEndLogEvent
8. RecMediaStartLogEvent
9. RecMediaEndLogEvent
10. MessageLogEvent
11. AdditionalAgencyLogEvent
12. LostQueryLogEvent
13. LostResponseLogEvent
14. CallSignalingMessageLogEvent
15. MalformedMessageLogEvent
16. EidoLogEvent
17. ElementStateChangeLogEvent
18. ServiceStateChangeLogEvent
19. QueueStateChangeLogEvent
20. AdditionalDataQueryLogEvent
21. AdditionalDataResponseLogEvent
22. LocationQueryLogEvent
23. LocationResponseLogEvent

The application does not need to log the RecordingFailedLogEvent because this event is only logged by an SRS.

The application shall not log CallSignalingMessageLogEvent events for SIP OPTIONS requests that it receives from upstream endpoints or that it sends to downstream SIPREC media recorders.

The application shall log the following events defined in Section 2.9 of NENA-STA-024.1a-2023.

1. EidoLogEvent (same as #16 above with additional fields)
2. EidoTransmissionErrorLogEvent
3. SubscriptionRequestedLogEvent
4. SubscriptionRequestedResponseLogEvent
5. SubscriptionTerminatedLogEvent
6. SubscriptionTerminatedResponseLogEvent
7. WebSocketEstablishedLogEvent
8. WebSocketTerminatedLogEvent

3.9.2.1 Event Logging Configuration Settings

The application shall provide the following configuration settings for event logging.

1. Enable/Disable Event Logging (global setting)
2. List of event logging servers

The user shall be able to add, edit or delete event logging servers.

The application shall provide the following configuration settings for each event logging server.

1. Event logger name
2. Logging service HTTP(s) URI
3. Enable/Disable

3.10Test Call Interface Requirements

The application shall support NG9-1-1 test calls as specified in Section 9 of NENA-STA-010.3b.

The application shall insert its identity into the Contact header of the OK response as specified in Section 9 of NENA-STA-010.3b.

The application shall interpret any request URI that starts with “urn:service:test.sos” as a test call and respond with a 200 OK response.

The application shall automatically handle test calls in the background and shall not require any user interaction.

Section 9 of NENA-STA-010.3b states:

> “To provide authentication, the Identity header field (RFC 8224 [60]) SHOULD be inserted, signed by an entity in the path (such as an ESRP) with a certificate traceable to the PCA.”

For now, the application does not need to support the Identity header.

If the application receives an SDP offer with a loopback role attribute of “loopback-mirror” then it shall reject the INVITE request for the test call with a 406 Not Acceptable response. The reason for this is that the PSAP is intended to be the loopback mirror.

The application shall support both RTP packet loopback (“rtp-pkt-loopback”) and the media loopback (“rtp-media-loopback”) tests specified in RFC 6849.

For RTP packet loopback (rtp-pkt-loopback), the application shall support both direct RTP packet loopback (see Section 7.2 of RFC 6849) and encapsulated RTP packet playback (see Section 7.1 of RFC 6849).

The application shall specify the “loopback-mirror” in the 200 OK response that it sends and it shall act as the media loopback mirror (it shall echo back any packets that it receives).

As specified in Section 9 of NENA-STA-010.3b, the application shall loopback no more than 3 packets of each media type offered in the INVITE request and then it shall send a BYE request to the remote endpoint. Note: The user can configure the application to terminate the call after 3 packets.

If the application does not receive 3 media packets for all offered media types within 500 milliseconds after sending the OK response, then it shall prematurely terminate the test call with a BYE request.

Section 9 of NENA-STA-010.3b, specifies that the application shall refuse repeated requests from the same device (same Contact header URI or same source IP address and port) within 2 minutes. It shall signal a test call refusal with a 486 Busy Here response. Since this application is a test program, there is no need to support this requirement.

The application shall not record loopback media for NG9-1-1 test calls.

Section 9 of NENA-STA-010.3b does not specify that a PSAP needs to log NG9-1-1 events for test calls so the application does not need to support NG9-1-1 event logging for test calls.

3.10.1 Additional Test Call Functional Requirements

The application shall provide the capability of extending the duration of incoming test calls beyond the 3 RTP packets specified in Section 9 of NENA-STA-010.3b via a configuration setting. The user shall be able to specify the maximum length of test calls in minutes. The application shall terminate the test call when it exceeds this limit.

The application shall accept a SIP BYE request from the entity that initiated the test call.

The application shall provide a setting that specifies the maximum number of simultaneous test calls that it can handle. If a new test call arrives when the application is handling the maximum number of test calls, then it shall reject the call with a 486 Busy Here response.

The application shall provide a setting that enables or disables test calls. If test calls are disabled then the application shall reject all test call requests with a 503 Service Not Available response.

When the application is ending, it shall gracefully terminate all active test calls by sending a BYE request for each test call.

The application shall support SDES-SRTP or DTLS-SRTP if the test call offers either form of media encryption.

Neither Section 9 of NENA-STA-010.3b nor RFC 6849 mention MSRP media so the application does not need to support MSRP.

3.11 Advanced Automatic Crash Notification Calls

The application shall support NG-AACN calls as specified in Section 3.1.19 of NENA-STA-010.3b and RFC 8148.

The application shall support multi-media NG-AACN calls.

The application shall support the Vehicular Emergency Data Set (VEDS) provided with the INVITE request for an NG-AACN call either by value or by-reference. The default is expected to be by-value.

The application does not need to support the ability for the call taker to request the vehicle to perform any actions described in RFC 8148 at this time.

3.12 Non-Interactive Calls

The application shall be able to handle non-interactive calls as specified in Section 3.1.11 of NENA-STA-010.3b and in RFC 8876.

Non-interactive calls do not have any media. They contain a Common Alert Protocol (CAP) XML document and a PIDF-LO (XML) document. Non-interactive calls use the SIP MESSAGE method instead of the INVITE method.

The MESSAGE request may also include NG-9-1-1 Additional Data (RFC 7852) so the application needs to support additional data.

The application shall be able to handle the data (CAP, PIDF-LO or additional data) by-value or by-reference.

The application does not need to support the Emergency Data Exchange Language – Distribution Element (EDXL-DE) wrapper format of the CAP emergency data at this time.

It shall not be possible for the user to set up a conference or transfer a non-interactive call.

3.13 Conference Bridge Interface

Section 4.7.1 of NENA-STA-010.3b describes the following two methods of setting of a conference.

1. Ad hoc method
2. Route All Calls Via a Conference Aware UA

The application shall support both of these methods.

If the Contact header of the INVITE request for an incoming call has an “isfocus” header parameter, then the application shall use method 2 to set up a conference. If there is no “isfocus” header parameter in the Contact header then the application shall use method 1.

The application shall prevent the user from attempting to conference a call if the Contact header does not have an “isfocus” header parameter and there is no conference bridge URI configured.

When the user wishes to setup a conference, the application shall allow the user to select the transfer target from a preconfigured list of transfer targets or to enter a full SIP URI of a transfer target.

The application shall support the conference event package described in RFC 4575. When the application answers a call that contains a Contact header with an “isfocus” header parameter, it shall immediately subscribe to the conference event at the SIP URI from the Contact header when it answers the call.

The user shall be able to view a list of the current participants of a conference.

The user shall be able to add new members to the conference.

The user shall be able to remove conference members.

The application shall support the method of passing the current EIDO for the call the transfer target using the method specified in Section 4.7.4 of NENA-STA-010.3b. This requires that the application provide an HTTPS server so that EIDOs can be dereferenced by a transfer target.

Note: Method 2 is the current industry standard method of conference setup so this method shall be implemented first.

3.13.1 Conference Bridge Configuration Settings

In order to support the ad-hoc transfer method, the application shall provide the following configuration settings.

Setting Name	Type	Description
Bridge SIP URI	String	Optional. Specifies the SIP URI of the conference bridge to use when using the ad hoc conference method

If the Bridge SIP URI is not specified then the application shall prevent the user from initiating an ad-hoc NG9-1-1 transfer.

3.13.2 Transfer Target Phone Book

The application shall allow the user to create a list of transfer targets. The list of transfer targets will be stored with the application settings.

The user shall be able to add, edit or delete transfer targets.

Each transfer target shall have the following settings.

Setting Name	Type	Description
Friendly Name	String	Required. Must be unique.
SIP URI	String	Required.

The transfer target list may be empty.

3.14 CAD Interface

The application shall provide an interface to one or more Computer Aided Dispatch (CAD) systems. This interface provides a method of sending an Emergency Incident Data Object (EIDO) to CAD systems that have subscribed with this application.

NENA-STA-021.1a-2022 describes the format of the EIDO document.

NENA_STA-024-1a-2023 describes the conveyance mechanism and the subscribe/notify protocol to be used for sending EIDOs to CAD systems.

The application shall listen on all enabled IP transport protocols (IPv4, IPv6 or both) for EIDO subscription requests.

The application shall use port number 16000 for both IPv4 and IPv6.

The application shall use the secure Web Socket (WSS) protocol for the CAD Interface.

The application shall send an EIDO to each subscribed CAD system when the call state changes. A call state change is one of the following events.

1. A call is answered
2. A call is put on hold
3. A call on hold is picked up
4. A conference is set up or the conferenced call is transferred
5. A conference member is added or removed
6. A call ended
7. Location or additional data for the call is received by the application

When a CAD system subscribes to receive EIDOs, the application shall send it the current EIDO for each call that the application is currently handling.

The user shall have the ability to view the subscription and connection status of each subscribed CAD system.

3.15 EIDO Retrieval Service

The EIDO shall implement both the server side and the client side of the EIDO retrieval service described in Sections 2.6 and 2.7 of NENA-STA-024.1a-2023. The EIDO retrieval service is an HTTPS RESTful interface that allows a PSAP that is the transfer target in a conference/transfer operation to retrieve all of the available call information in the form of an EIDO from the PSAP that initiated the conference/transfer operation.

3.15.1 Server-Side Requirements

When the PsapSimulator initiates a conference/transfer operation it acts as the server for the EIDO retrieval. It sends a URI to the transfer target via the conference bridge and responds to an HTTPS GET request for the EIDO.

The mechanism for this is as follows.

When the application initiates a conference/transfer it sends a REFER request to the conference aware user agent specified in the Contact header of the INVITE request for the incoming call. This REFER request shall contain a SIP Refer-To header that specifies the transfer target that the conference bridge will INVITE to the conference. This Refer-To header also contains an embedded Call-Info header that contains an HTTPS URI for the EIDO for the call.

Example:

Refer-To: <sip:Psap2@192.68.1.64?Call-Info=%3Chttps%3A%2F%2F192.168.1.84%3A11000%2Fincidents%2Feido%2Fb7686f36-e86b-45a8-a809-9dc2d46d40f3%3E%3Bpurpose%3Demergency-eido>

The above Refer-To header will cause the conference bridge to add the following Call-Info header to the INVITE request that it sends to the transfer target.

Call-Info: <https://192.168.1.84:11000/incidents/eido/b7686f36-e86b-45a8-a809-9dc2d46d40f3>;purpose=emergency-eido

When the transfer target PSAP receives an INVITE request with a Call-Info header like that shown above, it shall send an HTTPS GET request to the PsapSimulator application. The application will then provide the EIDO for the call in the body of the HTTPS 200 OK response that it sends in response to the GET request.

The transfer target may at any time perform a subsequent HTTPS GET request to the URI to get an update of the call state.

The URI shall be active for the duration of the call at the PSAP that initiated the conference/transfer operation.

3.15.2 Client-Side Requirements

When the PsapSimulator receives an INVITE request with a Call-Info that has a purpose parameter of "emergency-eido", it shall act as a client to the original PSAP's EIDO retrieval service by sending an HTTPS GET request to the HTTPS URI specified in that Call-Info header.

The application shall send an HTTPS GET request after it sends a 100 Trying response to the incoming INVITE request. The application shall also send an HTTPS GET request when the user sets the call to the on-line state. Section 3.16.1 Processing Received EIDOs specifies how the application shall process EIDOs that it receives.

3.16 EIDO Document Handling

3.16.1 Building EIDOs

The following table specifies which fields of the EIDO that the application shall provide. The "Section" column contains the section number from NENA-STA-021.1a-2022. The "Required" column indicates whether or not that field is required in NENA-STA-021.1a-2022. The "Supported" column indicates whether or not the application shall support the EIDO field.

EIDO Field	Section	Required?	Supported?	Description
eidoVersion	2.4	Yes	Yes	Set to "1.0".
Id		Yes	Yes	Set to the emergency incident ID of the call
issuingElementIdentification	2.4	Yes	Yes	Set to the Element ID configuration setting of the application
mergeComponent	2.7	No	No	Contains merge and split information related to the Incident.

linkComponent	2.8	No	No	Contains link information related to the Incident.
incidentComponent	2.9	No	Yes	Contains general information about the Incident.
callComponent	2.10	No	Yes	Contains information about calls associated with the Incident.
callbackComponent	2.11	No	Yes	Contains information about how to call a person.
dispatchComponent	2.13	No	No	Contains dispatch information related to the Incident.
notesComponent	2.15	No	No	Contains Incident notes and comments associated with the Incident.
emergencyResourceComponent	2.20	No	No	Identifies emergency resources involved with the Incident.
alarmsSensorComponent	2.21	No	No	Identifies Alarms/Sensors associated with the Incident.
agencyComponent	2.6	Yes	Yes	Identifies all agencies involved with the Incident.
agentComponent	2.5	No	Yes	Identifies all agent involved with the Incident. Must be provided if an Agent is involved in the Incident.
additionalDataComponent	2.19	No	Yes	All additional data related to the Incident. The application shall provide additional data by-value and does not need to support additional data by-reference.
locationComponent	2.18	No	Yes	All locations related to the Incident. The application shall support location by-value and does not need to support location by-reference.
personComponent	2.16	No	Yes	Every person related to the Incident.
vehicleComponent	2.17	No	No	Every vehicle related to the Incident.

3.16.2 Processing Received EIDOs

When the application receives an EIDO in response to an HTTPS GET request that it sent see Section 3.15.2), it shall parse the EIDO and populate the following information for the call.

1. Additional Data
2. Location
3. Other (TBD)

4 Call Handling Requirements

The application shall be able to handle multiple incoming NG9-1-1 multimedia calls.

The user shall be able to send and receive media from only one call at a time.

The application shall be capable of handling up to a configured maximum number of calls. If the number of incoming calls and pending call requests exceeds the configured maximum then the application shall respond with a 486 Busy Here response.

The user shall be able to set up a conference for the currently selected call.

If the currently selected call is already in a conference, the user shall be able to add new conference participants or remove current conference participants.

The application shall provide an auto answer function that can be configured.

If the auto answer function is enabled, the application shall automatically answer each incoming call and then place it on “hold”.

If auto answer is not enabled, the application shall accept each incoming and respond with a 180 Ringing SIP response. Each incoming call shall be placed in the call queue.

If the application does not currently have a call on-line, then the application shall play ring sound to the user’s headset if present or the computer’s speakers when a new call arrives.

The application shall stop playing ring sound to the user when a call is answered or picked up.

The user shall be able to answer the longest ringing call in the queue with a single button click.

The user shall be able to display the current call queue and to select any call in the queue that is either in the ringing state or the on-hold state.

If there is a currently selected call then the application shall put the currently selected call on-hold and switch the media and call data displays to the newly selected call.

The user shall be able to select any calls in the call queue and end that call without changing the currently selected call.

There shall be an option for the user to end all calls in the call queue.

The user shall be able to add media to the currently selected call.

If the current call has MSRP media and is conferenced, the user shall be able to send private messages to any of the conference members except the original caller.

4.1 Placing Calls on Hold

Unlike a standard telephone hold operation, both send and receive media shall remain active when an incoming call is placed on hold. When a call is on hold, transmit media is taken from the following sources.

Media Type	On-Hold Media Source
Audio	Pre-recorded file
Video	A static image from a JPEG file
MSRP	A configured MSRP message
RTT	A configured RTT message

The configured MSRP and configured RTT messages will be only sent once when the call state transitions from on-line to on-hold.

The application shall continue to send media to all configured SIPREC media recorders when a call is on hold.

If a call is currently conferenced then the application shall not play MOH when the call is placed on hold, it shall send silence. It shall also not send the configured MSRP or RTT messages.

4.2 Miscellaneous Special SIP Protocol Requirements

The application shall be able to handle offer-less INVITE requests. In an offer-less INVITE request, the caller does not provide an SDP body in the INVITE request. The application provides its SDP offer in the 200 OK response that it sends to the caller, then the caller provides its SDP in the ACK request that it sends in response to the OK response.

An offer-less INVITE may contain PIDF-LO location data and additional data in the body of the request. The application shall be able to handle this condition.

If the INVITE request for an incoming call contains an offer of both MSRP and RTT media then the application shall accept only the MSRP media and reject the RTT offer.

Section 3.1.1.1 of NENA-STA-010.3b recommends that provisional responses (non-Trying) should be sent every 3 seconds. The application shall send 180 Ringing every 3 seconds.

The application shall send the 180 Ringing response. It shall not use 183 Session Progress.

For outgoing calls, the application shall treat a 183 Session Progress as a 180 Ringing response and ignore any media that may be sent. Section 3.1.1.1 of NENA-STA-010.3b states:

> "An i3 PSAP SHOULD normally only return a 180 Ringing provisional response when a 9-1-1 call is queued for answer. 183 Session Progress may be used in some specific circumstances."

The application shall support the RTCP attribute in the SDP as specified in RFC 3605. This is required in Section 3.1.9 of NENA-STA-010.3b. The application shall always send RTCP packets on the next port number of the media (i.e. the odd port) and shall set the "rtcp" attribute to that port. The application shall check for the presence of the "rtcp" attribute in the SDP it receives and shall expect to receive RTCP packets on that port. If the SDP that the application receives does not contain an "rtcp" attribute then it shall assume that the remote endpoint is using the media port number plus 1.

The application shall support the SDP Label attribute specified in RFC 4574. **Note:** the label attribute is used only for SIPREC calls.

The application shall implement support for the SIP Replaces header as defined in RFC 3891. This header is required for the ad-hoc transfer method.

4.3 Call Identifier and Incident Tracking Identifier Requirements

Calls that are delivered to this application are expected to be NG9-1-1 calls. Incoming NG9-1-1 are expected to have the NG9-1-1 Call-Info headers for a Call Identifier (see Section 2.1.6 of NENA-STA-010.3b) and an Incident Tracking Identifier (see Section 2.1.7 of NENA-STA-010.3b).

The application shall automatically create a Call Identifier if one is not present in the incoming INVITE request. The application shall automatically create an Incident Tracking Identifier if one is not present in the incoming INVITE request. The reason for this requirement is that these identifiers are required for NG9-1-1 event logging and SIPREC media recording.

4.4 DTMF Digits Transmission Requirements

The need to send DTMF digits on an audio media channel may arise in situations where the application performs a callback call or an outgoing call. This can happen if the outgoing call hits a system that is using an Interactive Voice Response (IVR) system to answer calls.

The application shall support DTMF telephone events in the RTP payload as specified in RFC 4733.

The application shall always offer DTMF telephone event media in the SDP that it sends.

The application only needs to support sending the following DTMF events: digits 0-9, # and *.

The application shall provide a DTMF keypad pop-up dialog box that the user can use to end DTMF digits to send.

The user shall be able to use the keyboard keys to enter DTMF digits.

The application shall send fixed length DTMF telephone events on the audio RTP channel. The digit duration is TBD.

The application shall queue outgoing DTMF digits in case the user enters the digits faster than they can be transmitted of the audio RTP channel.

The application does not need to decode incoming DTMF events.

4.5 Selected Call Display Requirements

The selected call display of the application shall display information about the call that the user is currently communicating with. The user may communicate with only one call at a time.

4.5.1 Basic Call Information

1. The user portion of the From URI
2. Call State (Ringing, On-Line, On-Hold …)
3. Available media
4. Call Participant Information

4.5.1.1 Conference Participant Information

If the PsapSimulator application receives a NOTIFY request for the conference event SIP package from a conference-aware user agent that delivered the call to it, it shall display the following information from the conference information received in the body of the NOTIFY request for each call participant. See RFC 4575. This information may be sent to the PsapSimulator when it subscribes to the conference event package even though the call has not be added to a conference yet.

Column	Description
Participant	This shall be the user part of the SIP URI from the "entity" attribute for the "usertype" element for the call participant (user) if the "entity" attribute is a valid SIP URI. See Section 5.6 of RFC 4575. This shall be set to "Unknown" if the "entity" attribute is not a valid SIP URI. If the SIP URI does not contain a user part, then this field shall be set to the string version of the SIP URI.
Media	This field shall display a list of the types of media that the call participant is using for the call. This information shall be taken from the list of media for the first endpoint[1] for the call participant (user). See Section 5.7 of RFC 4575.
Status	This field shall display the status of from the first endpoint of the participant. Typical values are "Connected", "Disconnected", etc. This field shall be set to "Unknown" if the status is not specified. See Section 5.7 of RFC 4575.
Roles	This field shall display a list of roles for the user. See Sections 5.6 and 5.6.3 of RFC 4575. Typical values are "Caller" or "Call Taker". This field shall be set to "Unknown" if no role information is provided.

4.5.2 Video Display Requirements

The application shall provide a video preview display and a display for the caller's video if video media is enabled and available for the currently selected call.

There shall be a configuration option to enable or disable the transmit video from the computer's camera.

If transmit video is disabled, the application shall display a static image in the video preview display and transmit that static image to the caller.

The user shall have the option of overriding the transmit video setting while handling calls. If the user enables transmit video, the application shall replace the static image with frames captured from the computer's camera. If the user disables transmit video again then the application shall replace the camera's video frames with the static image file. Changing the setting during a call shall not change the configuration setting.

The application shall support incoming calls with video media even if the computer does not have a camera. If the computer does not have a camera, then the application shall always send a static image to the caller.

The static image file shall be configurable.

[1] The XML schema defined in RFC 4575 allows for each user to have multiple endpoints. In practical NG9-1-1, there is only one endpoint for each user.

4.5.3 Text Display

1. Text Type: MSRP or RTT
2. Message List
3. Text Box for typing a new message

The Message List shall show the following information for each text message that was sent or received.

1. From
2. Text of the message
3. Time that the message was sent or received

If the text type for the call is MSRP, then the call form shall have the following controls.

1. Send button
2. Send on Enter check box
3. Use CPIM check box
4. Private Message check box

4.5.4 Location

The application shall display the following location information provided in the most recently received PIDF-LO XML document for the currently selected call.

1. Latitude, Longitude
2. Radius in meters (if shape is a circle)
3. Elevation in meters if available
4. Confidence (0 – 100) if available
5. Location Method (GPS, A-GPS, Cell…) if available
6. Civic Address (formatted street address, city, state, county) if available.
7. Data Provider String (See Section 4.1.1 of RFC 7852) if the provider information is available in by-value in the provided-by element of the geopriv element.

The application shall automatically update the location information if it receives new location data.

The user shall have the ability to manually request updated location if the incoming INVITE has a Geolocation header containing an HTTP URI.

The ability to display geodetic information in a map display is not required at this time. A map display feature may be added in the future.

4.5.5 Subscriber Information

Subscriber Information shall be taken from the SubscriberInfo additional data block defined in Section 4.4 of RFC 7852. The application must be able to display the following subscriber information if it is available for the call.

1. Name (first, last, middle) – from the first xCard in the SubscriberInfo data block.
2. Telephone Number

3. E-Mail
4. Address (formatted street address, city, state, country)
5. Languages
6. Data Provider String (See Section 4.1.1 of RFC 7852)

The above data is only a small subset of the data available in the SubscriberInfo additional data block.

4.5.6 Service Information

Service information shall be taken from the ServiceInfo additional data block defined in Section 4.2 of RFC 7852. The application shall display the following service information if it is available for the call.

1. Service Environment (Section 4.2.1 of RFC 7852)
2. Service Type (Section 4.2.2 of RFC 7852)
3. Service Mobility (Section 4.2.3 of RFC 7852)
4. Data Provider String (See Section 4.1.1 of RFC 7852)

4.5.7 Device Information

Information about the calling device shall be taken from the DeviceInfo additional data block defined in Section 4.3 of RFC 7852. The application shall display the following device information if it is available for the call.

1. Device Classification (See Section 4.3.1 of RFC 7852)
2. Device Manufacturer (See Section 4.3.2 of RFC 7852)
3. Device Model Number (See Section 4.3.3 of RFC 7852)
4. Unique Device Identifier (See Section 4.3.4 of RFC 7852)
5. Data Provider String (See Section 4.1.1 of RFC 7852)

4.5.8 Provider Information

Provider information identifies the provider of one or more additional data information blocks. This information is in the ProviderInfo additional data block defined in Section 4.1 of RFC 7852.

The application shall display the following information for each ProviderInfo data block that it receives.

1. Data Provider String (See Section 4.1.1 of RFC 7852)
2. Type of Data Provider (See Section 4.1.4 of RFC 7852)
3. Data Provider Contact URI (See Section 4.1.5 of RFC 7852)

4.5.9 Comments

Textual comments are available in the Comment additional data block defined in Section 4.5 of RFC 7852.

There may be more than one comments data block and each comments data block may contain multiple comments. The application shall be capable of displaying all of the comments blocks that it receives.

4.5.10 AACN Information

The application shall be able to display at least the following information if provided in the VEDS document.

1. Vehicle VIN, year, make, model

2. Impact velocity
3. Vehicle location
4. Air bag deployment (i.e., indicating which airbags deployed)
5. Vehicle final resting orientation (e.g., on driver's side, on roof)
6. Number of occupants
7. Seat belt status
8. Hazardous cargo indicator(s)
9. Timestamp
10. Recent previous location
11. Call back number (e.g., to driver cellphone or vehicle cell number)

4.5.11 Selected Call Display Actions

1. Pickup or Answer
2. Hold
3. End Call
4. Add Media
5. Refresh Location
6. DTMF Keypad button
7. Conference Controls
8. Close the Form

Conference controls

1. Refer
2. Drop
3. Drop Last

4.6 Call Queue Display Requirements

The call queue display shall display the following information about each call.

1. The user portion of the From URI.
2. Time that the INVITE was received (HH:MM:SS)
3. Call State (Ringing, Hold, Auto-Answered or On-Line)
4. Queue URI (INVITE request URI or the Route header URI if present)
5. Conference status (conferenced or not)
6. Media available (audio, video, MSRP, RTT)

The user shall be able to perform the following actions on the calls in the call queue display.

1. Answer a selected call
2. Answer the longest ringing call in the call queue
3. Pick up a call that is on hold or is auto-answered

4. End a selected call
5. End all calls

The software shall automatically remove calls from the call queue display when the call is terminated by the user of the calling party.

The call queue display shall provide the following summary information.

1. Total number of calls
2. Number of ringing calls
3. Number of calls on-hold
4. Number of auto-answered calls

4.7 Call History Storage and Display Requirements

TBD

4.8 Non-Interactive Call Display Requirements

TBD

5 Application Configuration Settings

5.1 Network Settings

Setting	Values	Default	Description
Network Type	IPv4, IPv6 or IPv4 and IPv6.	IPv4	Specifies the networks that the application will listen on.
SIP Transport Protocol	UDP, TCP, TLS	TCP	Allows the user to select which SIP transports to use. At least one must be selected.
Use Mutual Authentication	Boolean	True	Applies to SIP over TLS. If true then clients requesting a TLS connection must provide an X.509 certificate and the application will provide its X.509 certificate when connecting as a client. If false then clients do not need to provide an X.509 certificate and the application will not provide its X.509 certificate when connecting as a client.
SIP Port Number	Numeric	5060	SIP port for UDP and TCP.
SIPS Port Number	Numeric	5061	SIP port for Transport Layer Security (TLS).
IPv4 Address	String	NA	The default will be the last selected IP address for the IPv4 network if it's available, or the first IP address in the list of available IP addresses for the IPv4 network.
IPv6 Address	String	NA	The default will be the last selected IP address for the IPv6 network if it's available, or the first IP address in the list of available IP addresses for the IPv6 network.
Media Port Settings	NA	NA	See Media Port Ranges below.

5.1.1 Media Port Ranges

The application shall provide the following media port settings for each media type.

1. Starting Port
2. Number of Ports

The following table specifies the default media port settings.

Media Type	Starting Port	Number of Ports
Audio	6000	1000
Video	7000	1000
RTT	8000	1000

MSRP	9000	1000

5.2 Certificate Settings

The application requires an X.509 certificate so that it can act as a server for SIP TLS and HTTPS requests.

The application shall provide a default X.509 PFX file that will be installed by the installation program.

The user shall be allowed to provide a custom X.509 PFX file by specifying the following.

1. X.509 Certificate Path
2. X.509 Certificate Password

The user shall be able to restore the certificate to the default X.509 certificate file.

5.3 Call Handling Settings

5.3.1 Maximum Calls

This setting shall specify the maximum number of simultaneous calls that the application can handle. If a new call arrives when the application is already handling the maximum number of calls then the application shall reject the new call with a SIP 486 Busy Here response.

The minimum setting shall be 1. The default setting shall be 10. There is no maximum setting at this time.

5.3.2 Non-Interactive Maximum Calls

This setting shall specify the maximum number of simultaneous non-interactive calls that the application can handle. The application shall respond with a SIP 486 Busy Here response if a non-interactive call arrives while the application is handing this number of calls.

The minimum setting shall be 1. The default setting shall be 10. There is no maximum setting at this time.

5.3.3 Auto Answer

This is an on/off setting. If Auto Answer is on, then the application shall automatically answer all incoming calls. If Auto Answer is off then the application shall set the call state to ringing.

5.3.4 Media Source Settings

5.3.4.1 Auto Answer Media Source Settings

The user shall be able to configure the following auto answer parameters.

1. Audio auto answer recording file
2. Video auto answer static pattern file
3. Auto answer text message message

The application shall provide a default audio answer recording file, a default video static pattern file and a default text message.

The user shall have the ability to restore the above settings to their default values.

5.3.4.2 Call Hold Media Source Settings

The user shall be able to configure the following call hold parameters.

1. Audio call hold recording file, call hold tone sound or silence
2. Video call hold static pattern file
3. Text call hold message

The application shall provide defaults for the above parameters.

The user shall have the ability to restore the above settings to their default values.

5.3.5 Transmit Video Disabled Image File

This setting specifies the static image file that the application will transmit for calls with video media when the computer's camera is absent or transmit video is disabled.

The user shall be able to change the file location of this static image file.

The application shall provide a default static image file.

The user shall have the ability to restore this setting to its default value.

5.3.6 Enabled Media

This setting shall determine the types of media that the application shall accept for incoming calls and the types of media that the application shall offer for outgoing calls.

The media choices shall be:

1. Audio
2. Video
3. RTT
4. MSRP

At least 1 media type must be enabled.

The default setting shall be enabled for all media types.

This enabled media settings shall not prevent the user from adding a media type to a call that does not currently have that media type.

The application shall also provide a setting that enables or disables transmit video. If transmit video is disabled, the application shall transmit a static image file instead of video captured from a camera. The default shall be to enable transmit video.

5.3.7 Outgoing Call Media Encryption Settings

These settings specify the type of media encryption to offer for outgoing calls.

The following options shall be available for RTP type media (audio, video and RTT).

1. None
2. SDES-SRTP
3. DTLS-SRTP

The default setting shall be “None”.

The following options shall be available for MSRP media.

1. None
2. MSRPS (MSRP over TLS)

The default setting shall be “None”.

5.4 Audio Device Settings

The application shall provide a setting that specifies the audio device to use. The application shall provide a list of audio devices that are available on the computer. The user shall be able to select an audio device from this list.

The default audio device shall be the first audio device in the list of available audio devices.

5.5 Video Device Settings

The application shall provide the following video device settings.

1. Video device name
2. Image Format (NV12, YUY2, RGB etc.)
3. Image Resolution (640x480, 1280x720 etc.)
4. Frame Rate (10, 20, 30 fps)

The default video format shall be NV12, 1280x720 at 30 frames per second.

5.6 Identity Settings

The application shall provide settings for Agency ID, Agent ID and Element ID. These identification settings will be used for NG9-1-1 event logging.

The user shall be able to configure each of these NG9-1-1 identifiers.

The application shall provide a default value for each type of NG9-1-1 identifier.

The user shall be able to restore the default settings.

5.6.1 Agency ID

The Agency ID identifies the agency to which a PSAP belongs to. Section 2.1.1 of NENA-STA-010.3b describes the Agency ID.

The default setting shall be: ng911test.net

5.6.2 Agent ID

The Agent ID identifies an agent (call taker) within an agency. Section 2.1.2 of NENA-STA-010.3b describes the Agent ID.

The default setting shall be: psapsimulator1@ng911test.net.

5.6.3 Element ID

The Element ID is a logical name used to represent a physical implementation of a functional element. Section 2.1.3 describes the Element ID.

The default setting shall be: psapsimulator1.ng911test.net.

5.7 ESRP Settings

5.7.1 Enable De-Queue Registration

This setting enables or disables de-queue registration to the ESRP.

The default setting shall be false.

5.7.2 ESRP HTTP or HTTPS URI

This setting specifies the URI to send HTTP(s) de-queue registration request to.

5.7.3 List of Queues

The application shall allow the user to create of list of call queues to register on. The user shall be able to add, delete or edit call queues from this list.

5.7.3.1 Queue Settings

Each call queue shall have the following settings.

1. Queue SIP URI
2. Expiration Time in seconds (5 – 86400 seconds, default = 3600)
3. De-queue Preference (1 – 5)
4. Enabled (true or false, default = true)

5.8 ECRF Settings

The application shall provide the following ECRF interface settings.

Setting	Type	Default	Description
LoST URI	String	Null	Specifies the HTTP(s) URI that the application will use to perform Location to Service Translation (LoST) requests. See Section 3.2. Optional. If null then LoST queries cannot be performed.
Agency Locator URI	String	Null	Specifies the HTTP(s) URI that the application will use to perform agency locator requests. See Section 3.2. Optional. If null then agency locator requests cannot be performed.

5.9 Outbound Call Interface Function (OCIF) Settings

These settings specify how the application can place outgoing callback calls or simple outgoing calls. See Section 3.1.3.

5.9.1 IP Endpoint

This setting specifies the IP endpoint at which the OCIF may be contacted at. The IP address may be an IPv4 or IPv6 address.

This setting is optional. If not specified then outgoing calls cannot be made.

5.9.2 SIP Transport Setting

This setting specifies the SIP transport (UDP, TCP or TLS) to use for contacting the OCIF. The default setting shall be TCP.

5.9.3 Media Encryption Settings

See Section 5.3.7.

5.10 NG9-1-1 Logging Service Settings

5.10.1 SIPREC Media Recording Settings

The application shall provide the following settings for SIPREC media recording.

Setting	Type	Default	Description
Enable SIPREC	Boolean	False	Specifies whether or not media from calls is recorded.
SIPREC Recorder List	List<>	Empty	Contains a list of SIPREC media recorder settings. The user can add, edit or delete recorders.

5.10.1.1 SIPREC Recorder Settings

Setting	Type	Default	Description
Name	String	NA	Specifies the friendly name of the SIPREC media recorder. Must be SIP URI compatible. Must be unique. Required.
Enabled	Boolean	True	If True, then the application will use this recorder to record media.
SIP Transport	String	TCP	Must be one of UDP, TCP or TLS.
IP Endpoint	String	NA	Specifies the IP endpoint of the SIPREC media recorder. Required.
Media Encryption Settings	See Section 5.9.3	See Section 5.9.3	Specifies the media encryption that the application will offer to the SIPREC media recorder.

5.10.2 NG9-1-1 Event Logging Settings

Setting	Type	Default	Description

Enable Event Logging	Boolean	False	Specifies whether or not to log NG9-1-1 events.
Event Logger List	List<>	Empty	Contains a list of NG9-1-1 event loggers. The user shall be able to add, edit or delete event loggers from this list.

5.10.2.1 Event Logger Settings

Setting	Type	Default	Description
Name	String	NA	Specifies the friendly name of the event logger. Must be unique. Required.
Enabled	Boolean	True	If true then the application will log NG9-1-1 events with this logger.
Logger URI	String	NA	Specifies the HTTP(s) URI of the logger. Required. Must be unique.

5.11 Test Call Settings

Section 3.10 specifies the requirements for handling test calls. The application shall provide the following configuration settings for test calls.

Setting	Type	Default	Description
Enable	Boolean	True	If true then the application shall respond to test call requests. If false, then the application shall reject test calls with a 503 Service Not Available response.
MaxTestCalls	Integer	1	Specifies the maximum number concurrent test calls. The minimum number is 1 and there is no upper limit.
DurationUnits	Integer	0	Shall be 0 or 1. 0 = Duration in packets. 1 = Duration in minutes.
DurationPackets	Integer	3	Specifies the number of RTP packets to receive before terminating the call. Used only if DurationUnits = 0 (packets). The minimum value shall be 3. There is no upper limit.
DurationMinutes	Integer	1	Duration in minutes. If the test call duration exceeds this limit then the application shall terminate the test call.

6 Application Logging Requirements

The application software shall write significant events to an application log.

Each log event shall contain the following information.

1. Date/Time (format shall be: 2023-08-26T08:27:53.18042)
2. Application software version
3. Logging level (DEBUG, INFO, WARNING, ERROR, CRITICAL)
4. Class name
5. Class method
6. Message
7. Exception information (may be null)

The application shall use a rolling file appendor. The maximum file size for each application log file shall be 1Mbyte. The maximum number of application log files shall be 5.

The location of the application log files shall be: the Windows special folder called LocalApplicationData. For instance, if the current user is “John”, the application log files will be located in: C:\Users\John\AppData\Local\PsapSimulator\Logs.

7 Issues with NENA-STA-010.3b and Future Development

7.1 RFC 4235 an INVITE-Initiated Dialog Event Package for SIP

Section 3.1.1.1 INVITE of NENA-STA-010.3b states that all SIP entities must support RFC 4235. This RFC specifies a SIP Subscribe/Notify event package. The abstract of this RFC states:

> "The dialog package allows users to subscribe to another user and to receive notification of the changes in state of INVITE-initiated dialog usages in which the subscribed-to user is involved."

The application shall not implement this RFC because NENA-STA-010.3b does not specify how this Subscribe/Notify event package is to be used in the context of NG9-1-1 calls and I am not aware of any vendor having ever implemented this event package.

7.2 RFC 4508 Conveying Feature Tags with the SIP REFER Method

Section 3.1.1.2 of NENA-STA-010.3b states:

> "SIP entities implementing REFER MUST implement RFC 4508 …"

RFC 4598 extends the SIP REFER method to allow the use of feature tags defined in RFC 3840. RFC 3840 defines a collection of feature tags that indicate user capabilities.

The application shall not implement this RFC because NENA-STA-010.3b does not specify how the user capabilities relate to NG9-1-1 calls.

7.3 RFC 3857 A Watcher Event Template Package for SIP

Section 3.1.3.2 of NENA-STA-010.3b states:

> "Entities implementing a notifier MUST implement RFC 3857"

RFC 3857 defines the watcher information template-package for SIP. The abstract for this RFC states:

> "Watcher information refers to the set of users subscribed to a particular resource within a particular event package. A user can subscribe to this information, and therefore learn about changes to it."

The application shall not implement the functionality defined in RFC 3857 because NENA-STA-010.3b provides no justification as to why a subscription watcher is required in the context of NG9-1-1.

7.4 RFC 5888 The Session Description Protocol Grouping Framework

Section 3.1.9 Media of NENA-STA-010.3b states:

> "All elements in the ESInet/NGCS MUST support RFC 5888 …"

The abstract of RFC 5888 states:

> "In this specification, we define a framework to group "m" lines in the Session Description Protocol (SDP) for different purposes. This framework uses the "group" and "mid" SDP attributes, both of which are defined in this specification. Additionally, we specify how to use the framework for two different purposes: for lip synchronization and for receiving a media flow consisting of several media streams on different transport addresses."

The application does not need to implement RFC 5888 at this time because there does not appear to be a requirement for it and no other vendors appear to support this requirement.

7.5 Network Address Translation

Section 3.1.18 of NENA-STA-010.3b states:

> "All elements in an ESInet that implement SIP interfaces MUST comply with RFC 5626 (Outbound) to maintain connections from User Agents."

RFC 5626 is entitled "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)".

This RFC requires the use of the SIP REGISTER method for user agents. However Section xxx.x of NENA-STA-010.3b states that the REGISTER method shall not be used in an NG9-1-1 ESInet. However, Section 3.1.3.1 of NENA-STA-010.3b states:

> "Use of REGISTER is not defined in this document, so REGISTER SHALL NOT be used."

Therefore, this application does not need to support RFC 5626.

Section 3.1.18 of NENA-STA-010.3b also states:

> "PSAPs, IMRs, bridges and other elements that terminate calls from entities outside an ESInet that may be behind NATs MUST implement "Interactive Connectivity Establishment (ICE)", RFC 8445 which includes support for "Session Traversal Utilities for NAT (STUN), RFC 5389. ESInets/NGCS SHOULD maintain a "Traversal Using Relays around NAT (TURN)" (RFC 5766) server for use by entities inside the ESInet placing outbound calls."

It is not expected that this application will be used behind a NAT so the application does not need to implement the above requirements at this time.

7.6 NENA-STA-010.3b Section 9 Test Call Issues

On page 402, the standard says that the PSAP should accept the "rtp-start-loopback" option. There is no mention of "rtp-start-loopback" in RFC 6849. What does "rtp-start-loopback" mean?

On page 402, the standard says: "The PSAP user agent would specify a loopback attribute of "loopback-source", the PSAP being the mirror". If the PSAP is the mirror, it should answer with "loopback-mirror" and not "loopback-source".

On page 402, the standard says: "If the location was provided by value, the response would be a natural text version of the received location." What does "natural text version" mean? If there is no specification of what this means then it does not seem that the location information is useful.

On page 402, the standard says: "If the location was provided by reference, the PSAP SHOULD dereference the location, using credentials acceptable to the LIS issued specifically for test purposes.

Credentials issued by a PCA-rooted CA MUST have the token “test” as the agent name or the first token in the FQDN.” This means that a separate certificate is required. Is this really necessary?

8 Revision History

8.1 Revision 1.0.0 – 25 Sep 2025

Initial version for software version 1.0.0.