

The Sen API

Interface Control Document

An example, generated from the data model by Sen. The reference that follows — every package, every type, every table — is written by the tool and is rewritten whenever the model changes.

This page is not. It is an ordinary Typst file included at a point the generated document leaves open, which is how a project puts its own cover, approvals and appendices around a reference it never has to edit.

Placeholder for your own project or product assets

Contents

About this document	3
Model overview	4
Reading the tables	5
components	6
ether	7
explorer	13
influx	14
jsonrpc	15
logmaster	26
py	27
recorder	28
replayer	30
rest	32
shell	38
term	43
db	45
kernel	46
log	77
Built-in types	80
Index	81

About this document

This is an example published with Sen's documentation. It was produced by running the model through `sen generate typst` and compiling the result, with no editing in between.

Everything from *Model overview* onwards is generated. It follows the model and only the model: if a type is added, renamed or given a new description, the next build says so, and nothing anyone wrote by hand is lost in the process.

This page, and the cover before it, are not generated. They are Typst files of your own that the document includes at points it leaves open for you:

<code>--front-matter</code>	A cover, before the contents.
<code>--before-reference</code>	Anything between the contents and the model — a scope, a set of conventions, an approvals table. This page.
<code>--after-reference</code>	Appendices, after the model.
<code>--style</code>	Every decision about how the document looks.

The generator does not read any of those files. It writes an `#raw("#include")` and Typst resolves it when the document is compiled, so a cover page can be anything Typst can set — and changing it never means regenerating the model.

Model overview

343 types in 14 packages. 14 classes in 12 independent hierarchies.

Packages

Package	Types	Object classes	Fixed records	Enumerations	Variant records	Arrays	Optionals
components	—	—	—	—	—	—	—
ether	31	—	20	1	3	4	3
explorer	1	—	1	—	—	—	—
influx	3	—	1	1	—	1	—
jsonrpc	41	2	17	—	1	8	13
logmaster	3	2	1	—	—	—	—
py	3	1	1	—	—	1	—
recorder	6	1	2	1	—	2	—
replayer	4	2	1	1	—	—	—
rest	37	—	20	5	—	9	3
shell	27	1	20	3	1	2	—
term	8	1	2	2	—	3	—
db	3	—	3	—	—	—	—
kernel	160	4	80	23	8	37	8
log	16	—	11	1	1	3	—

Class hierarchy across all packages

components.jsonrpc.JsonRpcServer	15
components.jsonrpc.StaticFileServer	18
components.logmaster.LogMaster	26
components.logmaster.Logger	26
components.py.PythonInterpreter	27
components.recorder.Recorder	28
components.replayer.Replay	30
components.replayer.Replayer	30
components.shell.Shell	38
components.term.Term	43
kernel.KernelApi	46
kernel.VirtualClock	46
└─ kernel.VirtualKernelClock	46
└─ kernel.VirtualMasterClock	46

Reading the tables

Every property carries three flags, in this order.

Flag	Meaning
R0	Read-only: the value is published by whoever owns it and cannot be set from outside.
RW	Read-write: the value can be set as well as read.
D	Dynamic: the value changes over the life of an instance.
S	Static: the value is fixed once an instance exists.
C	Confirmed: delivery is acknowledged, and the update is retried until it arrives.
BE	Best effort: the update is sent once and may be lost.

components

ether

31 types.

Fixed records

BusConfig	Remember to set these to the same value in all related Sen instances	7
BusJoined		7
BusLeft		8
ByteRange	Range of values of a u8	8
Configuration	Overall component configuration	8
DiscoveryHubAddress	TCP-based discovery hub parameters	8
Endpoint		8
Ephemeral	Let the operating system select an ephemeral port	8
Hello		9
MulticastAddressRange	Inclusive IPv4 multicast address range	9
MulticastDiscovery	Multicast-based discovery parameters	9
PinnedPort	Bind exactly this port, fail closed if taken	9
PortConfig	This configuration is per process and does not need to match across Sen instances. Exclusions apply to a pinned port, not to an ephemeral one: an ephemeral bind asks the OS for any free port and does not learn which it was given. udpUnicast cannot be pinned. Its socket is per peer and UDP has no four-tuple to demultiplex on, so a second peer either fails to bind or silently takes the first peer's traffic.	9
PortRange	Inclusive port range	9
ProbePortRange	Bind within [min,max], random start, walk forward	10
ProtocolVersion		10
QueueConfig		10
Ready		10
SessionPresenceBeam		10
TcpDiscovery	TCP-based discovery parameters	10

BusConfig STRUCTURE

Remember to set these to the same value in all related Sen instances

Name	Type	Description
multicastRange	MulticastRange	Defaults to the whole of 239.0.0.0/8: [239], [0-255], [0-255], [0-255]. RFC 2365 assigns only 239.192.0.0/14, which holds 262144 addresses and is not enough to keep many bus names apart. Set [239], [192-195], [0-255], [0-255] to stay inside it. Remember to keep a range wide enough for avoiding collisions if you have many buses.
multicastExclusions	MulticastAddressExclusions	Address ranges that must not be allocated
multicastPort	u16	Defaults to 50985 Change this in case you have limited control over the infrastructure.
multicastDisabled	bool	If true, no multicast traffic is performed and all the bus settings above become meaningless. Defaults to false.

Named by [Configuration](#).

BusJoined STRUCTURE

Name	Type	Description
participantId	u32	
busId	u32	
busName	string	

Named by [ControlMessage](#).

BusLeft STRUCTURE

Name	Type	Description
participantId	u32	
busId	u32	
busName	string	

Named by [ControlMessage](#).

ByteRange STRUCTURE

Range of values of a u8

Name	Type	Description
min	u8	minimum range
max	u8	needs to be >= min

Named by [MulticastRange](#).

Configuration STRUCTURE

Overall component configuration

Name	Type	Description
discovery	DiscoveryConfig	Discovery-related parameters
busOutQueue	QueueConfig	Queue for broadcasting to buses
processUdpOutQueue	QueueConfig	Queue for communicating with processes over UDP
processTcpOutQueue	QueueConfig	Queue for communicating with processes over TCP
busConfig	BusConfig	Bus transport configuration
portExclusions	PortExclusions	Port ranges that this process must not use
portConfig	PortConfig	If not present, all port bindings are Ephemeral
runDiscoveryHub	u16	If present, run the TCP discovery hub on this port
networkDevice	string	If present, the network device to use

DiscoveryHubAddress STRUCTURE

TCP-based discovery hub parameters

Name	Type	Description
port	u16	Port where the discovery hub is running
host	string	name or IP of the host where the discovery hub is running

Named by [TcpDiscovery](#).

Endpoint STRUCTURE

Name	Type	Description
ip	u32	
port	u16	

Named by [EnpointList](#).

Ephemeral STRUCTURE

Let the operating system select an ephemeral port

Named by [PortBinding](#).

Hello STRUCTURE

Name	Type	Description
info	sen.kernel.ProcessInfo	
udpPort	u16	
version	ProtocolVersion	

Named by [ControlMessage](#).

MulticastAddressRange STRUCTURE

Inclusive IPv4 multicast address range

Name	Type	Description
min	string	
max	string	

Named by [MulticastAddressExclusions](#).

MulticastDiscovery STRUCTURE

Multicast-based discovery parameters

Name	Type	Description
beamPeriod	sen.Duration	How often are we beaming our presence
beamExpirationTime	sen.Duration	Expiry time used by the beam tracker
mcastGroup	string	Defaults to '239.255.0.44'
port	u16	Defaults to 60543
device	string	Device to use for the network communication
allowVirtual Interfaces	bool	If true, virtual (NO_CARRIER) interfaces are also considered

Named by [DiscoveryConfig](#).

PinnedPort STRUCTURE

Bind exactly this port, fail closed if taken

Name	Type	Description
port	u16	

Named by [PortBinding](#).

PortConfig STRUCTURE

This configuration is per process and does not need to match across Sen instances. Exclusions apply to a pinned port, not to an ephemeral one: an ephemeral bind asks the OS for any free port and does not learn which it was given. udpUnicast cannot be pinned. Its socket is per peer and UDP has no four-tuple to demultiplex on, so a second peer either fails to bind or silently takes the first peer's traffic.

Name	Type	Description
tcpAcceptor	PortBinding	inbound TCP listener
udpUnicast	PortBinding	per-peer UDP unicast; must be Ephemeral
tcpSource	PortBinding	outbound TCP source port

Named by [Configuration](#).

PortRange STRUCTURE

Inclusive port range

Name	Type	Description
min	u16	
max	u16	

Named by [PortExclusions](#).

ProbePortRange STRUCTURE

Bind within [min,max], random start, walk forward

Name	Type	Description
min	u16	
max	u16	

Named by [PortBinding](#).

ProtocolVersion STRUCTURE

Name	Type	Description
kernel	u32	
ether	u32	

Named by [Hello](#).

QueueConfig STRUCTURE

Name	Type	Description
evictionPolicy	QueueEvictionPolicy	What to do when a queue is full
maxSize	u64	Maximum element count, 0 means unbounded
warningLevel	u64	Element count where warning will be issued, 0 means none

Named by [Configuration](#).

Ready STRUCTURE

Named by [ControlMessage](#).

SessionPresenceBeam STRUCTURE

Name	Type	Description
protocolVersion	u16	
info	sen.kernel.ProcessInfo	
beamPeriod	sen.Duration	
endpoints	EnpointList	

TcpDiscovery STRUCTURE

TCP-based discovery parameters

Name	Type	Description
beamPeriod	sen.Duration	How often are we beaming our presence
beamExpirationTime	sen.Duration	Expiry time used by the beam tracker
hubAddress	DiscoveryHubAddress	Where to find the discovery hub
allowVirtual Interfaces	bool	If true, virtual (NO_CARRIER) interfaces are also considered

Named by [DiscoveryConfig](#).

Enumerations

QueueEvictionPolicy

What to do when a queue is full

11

QueueEvictionPolicy ENUMERATION

What to do when a queue is full

Held as u32.

dropOldest	0
dropNewest	1

Named by QueueConfig.

Variant records

ControlMessage

11

DiscoveryConfig

Configuration parameters related to the discovery protocol

11

PortBinding

11

ControlMessage VARIANT

Type	Description
Hello	
Ready	
BusJoined	
BusLeft	

DiscoveryConfig VARIANT

Configuration parameters related to the discovery protocol

Type	Description
MulticastDiscovery	
TcpDiscovery	

Named by Configuration.

PortBinding VARIANT

Type	Description
Ephemeral	
PinnedPort	
ProbePortRange	

Named by PortConfig.

Arrays

EnpointList

11

MulticastAddressExclusions

12

MulticastRange

Range of values of a IPv4 multicast address

12

PortExclusions

12

EnpointList SEQUENCE

```
sequence<Endpoint, 27> EnpointList;
```

Named by SessionPresenceBeam.

MulticastAddressExclusions SEQUENCE

```
sequence<MulticastAddressRange> MulticastAddressExclusions;
```

Named by [BusConfig](#).

MulticastRange SEQUENCE

Range of values of a IPv4 multicast address

```
array<ByteRange, 4> MulticastRange;
```

Named by [BusConfig](#).

PortExclusions SEQUENCE

```
sequence<PortRange> PortExclusions;
```

Named by [Configuration](#).

Optionals

MaybeDeviceName	12
MaybeDiscoveryHubPort	12
MaybePortConfig	12

MaybeDeviceName OPTIONAL

```
optional<string> MaybeDeviceName;
```

MaybeDiscoveryHubPort OPTIONAL

```
optional<u16> MaybeDiscoveryHubPort;
```

MaybePortConfig OPTIONAL

```
optional<PortConfig> MaybePortConfig;
```

explorer

1 types.

Fixed records

Configuration	Overall component configuration	13
-------------------------------	---------------------------------	----

Configuration

STRUCTURE

Overall component configuration

Name	Type	Description
layoutFile	string	the imgui layout file. Defaults to ""

influx

3 types.

Fixed records

[Configuration](#) Overall component configuration

14

Configuration STRUCTURE

Overall component configuration

Name	Type	Description
samplingPeriod	sen.Duration	sampling period for the recordings
selections	SelectionList	objects to be monitored (using sql)
telegrafAddress	string	ip or name of the Telegraf server
telegrafPort	u16	port used by the Telegraf server
protocol	Protocol	protocol used
batchSize	u32	points to batch before emission (0 means no batching)

Enumerations

[Protocol](#)

14

Protocol ENUMERATION

Held as u8.

udp	0
tcp	1

Named by [Configuration](#).

Arrays

[SelectionList](#)

14

SelectionList SEQUENCE

```
sequence<string> SelectionList;
```

Named by [Configuration](#).

jsonrpc

41 types.

Object classes

JsonRpcServer	JsonRpcServer -- the JSON-RPC over WebSocket external API, expressed as a Sen class. One instance per connected client lives on <code>`local.jsonrpc`</code> . Methods are the RPC surface; events are the server-pushed notifications. Wire framing (envelope, id, error mapping) is the dispatcher's job, not this class's. Wire-shape convention: JSON-RPC named params (<code>"params": { ... }</code>) use the same field names as the STL method's argument list. The dispatcher treats every method's params object as <code>{ <argName>: <value>, ... }</code> and walks <code>`method->getArgs()`</code> to build the VarList in declaration order. That lets the inbound dispatch path go through Sen's meta machinery with no per-method translation.	15
StaticFileServer	Server-wide registry of in-memory static-file bundles, served alongside the WebSocket+JSON-RPC surface by the same uWS app. Internal-only: lives on the configurable control bus (defaults to <code>`local.jsonrpc_control`</code>), not on the data-plane bus that hosts authenticated per-connection <code>JsonRpcServer</code> instances. The component publishes exactly one instance.	18

JsonRpcServer CLASS

JsonRpcServer -- the JSON-RPC over WebSocket external API, expressed as a Sen class. One instance per connected client lives on ``local.jsonrpc``. Methods are the RPC surface; events are the server-pushed notifications. Wire framing (envelope, id, error mapping) is the dispatcher's job, not this class's. Wire-shape convention: JSON-RPC named params (`"params": { ... }`) use the same field names as the STL method's argument list. The dispatcher treats every method's params object as `{ <argName>: <value>, ... }` and walks ``method->getArgs()`` to build the VarList in declaration order. That lets the inbound dispatch path go through Sen's meta machinery with no per-method translation.

Name	Type	Flags	Description
identity	Identity	RO S BE	Identity of the connected client.

Method	Signature	Description
ping	<code>() → string</code>	Liveness probe.
listTopology	<code>() → SessionInfoList</code>	Snapshot of the currently-detected sessions, each with the buses currently detected inside it. The kernel's discovery surface is asynchronous; this returns whatever was visible at the moment of the call. Use <code>`subscribeTopology`</code> for live updates.
subscribeTopology	<code>()</code>	Sticky subscription to topology changes. Idempotent per connection. Wires a <code>`topologyChanged`</code> push immediately with the current full snapshot, then again on every session or bus add/remove. Cancel with <code>`unsubscribeTopology`</code> ; the subscription is also torn down automatically when the connection closes.
unsubscribeTopology	<code>()</code>	Cancel the topology subscription created by <code>`subscribeTopology`</code> . Idempotent.
getTypes	<code>() → sen.kernel.StringList</code>	Returns the qualified names of every custom type registered with the kernel. Catalog only -- pair with <code>`getType`</code> to fetch a chosen spec. The bundled-with-objects path on <code>`interestUpdate.types`</code> remains the primary way clients learn about types they observe.

Method	Signature	Description
getType	(qualifiedName: string , withSchema: bool) → TypeLookupResult	Look up one type by qualified name. Returns the CustomTypeSpec; when `withSchema` is true, also returns the JSON-Schema fragment for the type (empty string otherwise). Dependency chasing is the client's job: each schema fragment carries `ref` strings keyed by qualified name, resolved against whatever schema cache the client maintains.
createInterest	(interestName: string , query: string , subscribe: SubscribeBlock , withSchemas: bool)	Register a sticky interest defined by an STL query. Optional `subscribe` block pre-subscribes to properties/events on every matched object, including ones that arrive later. When `withSchemas` is true, every subsequent `interestUpdate` for this interest carries the JSON-Schema fragment for each previously-unseen type alongside the existing `CustomTypeSpec`. Re-registering the same name = invalidParams. Wire-ordering contract: the initial `interestUpdate.added` for the current match set is emitted <i>before</i> this RPC's response lands on the wire. Clients must accept pushes for an interest whose response has not yet been observed.
releaseInterest	(interestName: string)	Drop an interest and every subscription living under it.
listObjects	(interestName: string) → ObjectInfos	List the names and classes of objects currently in an interest's match set.
getProperty	(interestName: string , objectName: string , propertyName: string) → string	Read a property's current value, synchronously.
getObjectsBatchState	(interestName: string , objectNames: sen.kernel.StringList , propertyNames: sen.kernel.StringList) → ObjectStateList	Batched per-object snapshot. Walks the interest's match set in one pass instead of forcing the caller to issue N*M `getProperty` round-trips. Omitting `objectNames` reads every object currently matched; omitting `propertyNames` reads every property of each object (the class hierarchy is walked, so inherited members are included). Names from `objectNames` that aren't currently in the match set are silently skipped -- the caller compares its requested list against the returned `objectName`s to detect omissions. Names from `propertyNames` that don't exist on a given object's class are recorded per-object under `errors`, so a heterogeneous batch still returns useful partial results. Bounded: the call rejects with `invalidParams` when the resolved object set exceeds 512 entries (either via an explicit

Method	Signature	Description
setProperty	(interestName: <code>string</code> , objectName: <code>string</code> , propertyName: <code>string</code> , value: <code>string</code>)	<code>`objectNames`</code> list of that length, or via an implicit walk over a match set that has grown that large). Page the request by passing <code>`objectNames`</code>. Write a property's value (sugar over invoking the generated setter). Hand-routed in <code>`Dispatcher::handleSetProperty`</code>: the dispatcher issues <code>`invokeUntyped`</code> against the property's setter and wires the response through the kernel callback in a single tick. The typed <code>`setPropertyImpl`</code> override in the C++ <code>`Server`</code> exists only to satisfy the codegen-generated abstract base and is unreachable at runtime. See <code>`architecture.md`</code> "Method dispatch" for why this method (and <code>`invoke`</code>) bypass the generic meta path.
subscribeProperty	(interestName: <code>string</code> , objectName: <code>string</code> , propertyName: <code>string</code> , maxRateHz: <code>f64</code>)	Subscribe to <code>propertyChanged</code> notifications for one (interest, object, property). Sticky: persists across object churn; auto-rewires when the named object (re)enters the match set. Snapshot of the current value is delivered immediately after wiring. The throttle is per-object, shared across every property/event subscribed for the same object. An explicit <code>`maxRateHz`</code> overrides the current rate for the object; omitting it preserves whatever rate is already in effect (so adding a per-property sub doesn't accidentally retune existing subs back to unlimited).
unsubscribeProperty	(interestName: <code>string</code> , objectName: <code>string</code> , propertyName: <code>string</code>)	Cancel a property subscription previously created with <code>subscribeProperty</code> .
subscribeEvent	(interestName: <code>string</code> , objectName: <code>string</code> , eventName: <code>string</code>)	Subscribe to <code>eventTriggered</code> notifications for one (interest, object, event).
unsubscribeEvent	(interestName: <code>string</code> , objectName: <code>string</code> , eventName: <code>string</code>)	Cancel an event subscription previously created with <code>subscribeEvent</code> .
subscribeAll	(interestName: <code>string</code> , objectName: <code>string</code> , maxRateHz: <code>f64</code>)	Subscribe to every property and every event of one named object. Wildcard expansion against the object's actual class (including parent classes). Object must currently be in the match set; absent objects are <code>invalidParams</code> .
unsubscribeAll	(interestName: <code>string</code> , objectName: <code>string</code>)	Cancel an all-members subscription previously created with <code>subscribeAll</code> .
invoke	(interestName: <code>string</code> , objectName: <code>string</code> , methodName: <code>string</code> , argsJson: <code>string</code>) → <code>string</code>	Call a method on a kernel object and await its return value. Hand-routed in <code>`Dispatcher::handleInvoke`</code> : validates the args, issues <code>`invokeUntyped`</code> , and wires the response through the kernel callback in a single tick. The typed

Method	Signature	Description
		<code>`invokeImpl`</code> override in the C++ <code>`Server`</code> exists only to satisfy the codegen-generated abstract base and is unreachable at runtime. See <code>`architecture.md`</code> "Method dispatch".

Event	Payload	Description
propertyChanged	(interestName: <code>string</code> , objectName: <code>string</code> , values: <code>PropertyValueList</code> , timestamp: <code>sen.TimeStamp</code>)	Bundled per-object snapshot of changed properties. Per-object bundling, not per-property: any property change on one object lands in a single envelope.
eventTriggered	(interestName: <code>string</code> , objectName: <code>string</code> , eventName: <code>string</code> , args: <code>string</code> , timestamp: <code>sen.TimeStamp</code>)	An event fired on one object in one interest. The event signature is decided by the source object's class, not by JSON-RPC.
interestUpdate	(interestName: <code>string</code> , added: <code>AddedObjectList</code> , removed: <code>sen.kernel.StringList</code> , types: <code>sen.kernel.CustomTypeSpecList</code> , typeSchemas: <code>string</code>)	Match-set delta for one interest. The dispatcher tracks knownTypes / knownSchemas per connection and only re-sends specs / schemas new to the client.
topologyChanged	(sessions: <code>SessionInfoList</code>)	Full snapshot of the topology after a session or bus add/remove. Pushed once on <code>`subscribeTopology`</code> (with the current state) and again on every change. Snapshot form rather than deltas: topology changes are rare, and the snapshot keeps client logic trivial and idempotent.
notificationsDropped	(count: <code>u64</code>)	Sent once when outbound backpressure clears on this connection, reporting how many unreliable notifications were dropped while the buffer was above the high watermark. Emitted reliably on purpose: the notice about dropped messages must not be droppable. A client that receives this has an incomplete picture and cannot tell which updates it missed -- only that it missed some. The correct response is to re-read what it displays (<code>getObjectsBatchState</code> over the affected interests) rather than to wait for the next change, which may never come for a property that has stopped moving.

StaticFileServer CLASS

Server-wide registry of in-memory static-file bundles, served alongside the WebSocket+JSON-RPC surface by the same uWS app. Internal-only: lives on the configurable control bus (defaults to ``local.jsonrpc_control``), not on the data-plane bus that hosts authenticated per-connection `JsonRpcServer` instances. The component publishes exactly one instance.

Name	Type	Flags	Description
registeredBundles	<code>RegisteredBundleInfoList</code>	RO D C	Currently-registered bundles. Externally read-only; the component mutates this whenever a bundle is registered or unregistered. Subscribe to receive notifications on every change.

Method	Signature	Description
registerStaticBundle	(bundle: <code>StaticBundle</code>) → <code>u64</code>	Register a bundle. The bundle is copied into the registry; the caller may release the argument

Method	Signature	Description
unregisterStaticBundle	(bundleId: u64)	afterwards. Routes are wired up asynchronously on the uWS thread; the returned bundleId is stable and may be used immediately with <code>unregisterStaticBundle</code> . Stop serving the bundle. The slot in the registry is nulled; the underlying uWS route remains registered (uWS does not expose a clean route-removal API) but becomes inert and returns 404. No-op if <code>bundleId</code> is unknown.

Fixed records

AddedObjectEntry	One object newly entering an interest's match set.	19
Configuration	JSON-RPC over WebSocket component configuration. Only <code>address</code> and <code>port</code> are required; every other field falls back to a documented default.	20
ConnectionLimits	Per-connection buffer / timeout knobs consulted on the uWS thread for the lifetime of the server. Every field is optional; unset fields fall back to the defaults resolved inside <code>ws_server.cpp</code> at the use site.	20
Identity	Identity of the connected client. Set once on connect by the dispatcher from the WebSocket upgrade authenticator. Read-only externally. Other components / admin tools can <code>SELECT * FROM local.jsonrpc</code> to enumerate currently-connected clients.	20
NamedSelection	Named arm of <code>MemberSelector</code> : matches only the listed members. Wire shape: <code>{type: "NamedSelection", value: {memberNames: [...]}}</code> .	20
ObjectInfo	One entry of <code>listObjects</code> result.	20
ObjectState	Per-object snapshot returned by <code>getObjectsBatchState</code> . <code>properties</code> carries successful reads (one entry per property name resolved against the object's class hierarchy); <code>errors</code> is empty when nothing failed for this object.	21
PropertyError	Per-property read error. Used when the caller-supplied <code>propertyNames</code> includes a name that doesn't exist on a given object's class, or when the accessor itself throws. Carried in a separate list from successful reads so a property literally named <code>error</code> can never collide with the failure envelope.	21
PropertyValuePair	Per-property changed value. The value type is decided by the addressed kernel object, not by JSON-RPC, so STL can't statically describe it.	21
RegisteredBundleInfo	Summary record returned by <code>listRegisteredBundles</code> . Excludes the file bytes; just enough to answer "what is currently being served, and where".	21
SessionInfo	One discovered session and the buses currently detected inside it. Names are opaque strings from the kernel's session-discovery surface.	21
StaticBundle	One bundle of in-memory files served on a configurable URL prefix. The same bundle's files are served at <code><urlPrefix>/<file.path></code> . <code>indexFileName</code> is the SPA fallback served whenever a request under <code>urlPrefix</code> does not match any file (lets the client-side router handle paths the server does not know about).	22
StaticFile	One file inside a <code>StaticBundle</code> . Bytes are owned by the bundle; <code>path</code> is relative to the bundle's <code>urlPrefix</code> and uses forward slashes regardless of the producer's OS. <code>contentType</code> is the MIME type the server will return verbatim; the bundle producer decides it (no extension-based inference happens server-side).	22
SubscribeBlock	Pre-subscription block bundled into <code>createInterest</code> . Wires the listed property / event subscriptions on every matched object -- including ones that arrive later. Wire shape mirrors the typed STL representation exactly: each <code>MaybeMemberSelector</code> is either omitted (no auto-subscribe), <code>null</code> , or one of the typed variant arms (<code>{type: "WildcardSelection", value: {}}</code> or <code>{type: "NamedSelection", value: {memberNames: [...]}}</code>); <code>maxRateHz</code> is an optional positive number. Strict alignment with the typed surface lets the dispatcher's uniform meta-dispatch path consume the wire directly via <code>fromJson</code> , without a per-method translation step.	22
TlsConfig	TLS materials. The schema is reserved so configs stay forward-compatible, but the component fails to start if <code>tls</code> is set; full TLS support is not yet wired through the uWebSockets SSL backend.	22
TypeLookupResult	Bundled return of <code>getType</code> : always the <code>CustomTypeSpec</code> , plus the JSON-Schema fragment when the caller opted in. <code>schema</code> is empty when not requested.	22
WildcardSelection	Wildcard arm of <code>MemberSelector</code> : matches every member of the relevant kind. Wire shape: <code>{type: "WildcardSelection", value: {}}</code> .	23

AddedObjectEntry STRUCTURE

One object newly entering an interest's match set.

Name	Type	Description
objectName	string	name of the newly-matched object
qualifiedClassName	string	package-qualified class name of the object
currentValues	PropertyValueList	initial values of pre-subscribed properties (empty if none wired)

Named by [AddedObjectList](#).

Configuration STRUCTURE

JSON-RPC over WebSocket component configuration. Only `address` and `port` are required; every other field falls back to a documented default.

Name	Type	Description
address	string	The IP address or hostname the server will bind to
port	u16	The port number the server will listen on
updateFreqHz	f64	Run-loop frequency in Hz; defaults to 60
connectionLimits	ConnectionLimits	Per-connection uWS buffer / timeout knobs
tls	TlsConfig	schema placeholder; setting this fails component start
controlBusName	string	Control-plane bus on `local`; defaults to "jsonrpc_control"

ConnectionLimits STRUCTURE

Per-connection buffer / timeout knobs consulted on the uWS thread for the lifetime of the server. Every field is optional; unset fields fall back to the defaults resolved inside ws_server.cpp at the use site.

Name	Type	Description
idleTimeoutSeconds	u16	uWS idle close timeout; defaults to 120
maxPayloadBytes	u32	Max inbound frame size; defaults to 16 KiB
maxBackpressureBytes	u32	uWS hard outbound buffer ceiling; defaults to 64 KiB
highBackpressureBytes	u32	Soft trigger for the BackpressureUpdate signal; defaults to 32 KiB

Named by [Configuration](#).

Identity STRUCTURE

Identity of the connected client. Set once on connect by the dispatcher from the WebSocket upgrade authenticator. Read-only externally. Other components / admin tools can `SELECT * FROM local.jsonrpc` to enumerate currently-connected clients.

Name	Type	Description
subject	string	authenticated subject identifier (e.g. user / service principal)

Named by [JsonRpcServer](#).

NamedSelection STRUCTURE

Named arm of MemberSelector: matches only the listed members. Wire shape: `{type: "NamedSelection", value: {memberNames: [...]}}`.

Name	Type	Description
memberNames	sen.kernel.StringList	member names to match

Named by [MemberSelector](#).

ObjectInfo STRUCTURE

One entry of listObjects' result.

Name	Type	Description
objectName	string	object name within the interest's match set
qualifiedClassName	string	package-qualified class name of the object

Named by [ObjectInfos](#).

ObjectState STRUCTURE

Per-object snapshot returned by ``getObjectsBatchState``. ``properties`` carries successful reads (one entry per property name resolved against the object's class hierarchy); ``errors`` is empty when nothing failed for this object.

Name	Type	Description
objectName	string	object name within the interest's match set
qualifiedClassName	string	package-qualified class name at read time
properties	PropertyValueList	resolved property values; empty when caller's filter matched nothing
errors	PropertyErrorList	per-property failures; empty when none occurred

Named by [ObjectStateList](#).

PropertyError STRUCTURE

Per-property read error. Used when the caller-supplied ``propertyNames`` includes a name that doesn't exist on a given object's class, or when the accessor itself throws. Carried in a separate list from successful reads so a property literally named ``error`` can never collide with the failure envelope.

Name	Type	Description
propertyName	string	name from the caller's <code>`propertyNames`</code> list that could not be read
error	string	human-readable reason (e.g. "unknown property")

Named by [PropertyErrorList](#).

PropertyValuePair STRUCTURE

Per-property changed value. The value type is decided by the addressed kernel object, not by JSON-RPC, so STL can't statically describe it.

Name	Type	Description
propertyName	string	property member name
value	string	JSON-encoded SenValue

Named by [PropertyValueList](#).

RegisteredBundleInfo STRUCTURE

Summary record returned by ``listRegisteredBundles``. Excludes the file bytes; just enough to answer "what is currently being served, and where".

Name	Type	Description
bundleId	u64	opaque id assigned by the server, stable for the bundle's lifetime
urlPrefix	string	bundle's URL prefix
fileCount	u64	number of files in the bundle

Named by [RegisteredBundleInfoList](#).

SessionInfo STRUCTURE

One discovered session and the buses currently detected inside it. Names are opaque strings from the kernel's session-discovery surface.

Name	Type	Description
name	string	session name
buses	sen.kernel.StringList	bus names currently detected in this session

Named by [SessionInfoList](#).

StaticBundle STRUCTURE

One bundle of in-memory files served on a configurable URL prefix. The same bundle's files are served at `<urlPrefix>/<file.path>`. `indexFileName` is the SPA fallback served whenever a request under `urlPrefix` does not match any file (lets the client-side router handle paths the server does not know about).

Name	Type	Description
urlPrefix	string	URL prefix, e.g. <code>"/explorer"</code> ; must start with <code>"/</code>
indexFileName	string	SPA fallback, e.g. <code>"index.html"</code> ; must match a <code>path</code> in <code>files</code>
files	StaticFileList	file set

Named by [StaticFileServer](#).

StaticFile STRUCTURE

One file inside a StaticBundle. Bytes are owned by the bundle; `path` is relative to the bundle's `urlPrefix` and uses forward slashes regardless of the producer's OS. `contentType` is the MIME type the server will return verbatim; the bundle producer decides it (no extension-based inference happens server-side).

Name	Type	Description
path	string	path within the bundle, e.g. <code>"assets/main.js"</code>
contentType	string	MIME type, e.g. <code>"text/html"</code>
contents	sen.kernel.Buffer	raw file bytes

Named by [StaticFileList](#).

SubscribeBlock STRUCTURE

Pre-subscription block bundled into `createInterest`. Wires the listed property / event subscriptions on every matched object -- including ones that arrive later. Wire shape mirrors the typed STL representation exactly: each `MaybeMemberSelector` is either omitted (no auto-subscribe), `null`, or one of the typed variant arms (`{type: "WildcardSelection", value: {}}` or `{type: "NamedSelection", value: {memberNames: [...]}}`); `maxRateHz` is an optional positive number. Strict alignment with the typed surface lets the dispatcher's uniform meta-dispatch path consume the wire directly via `fromJson`, without a per-method translation step.

Name	Type	Description
properties	MemberSelector	which properties to auto-subscribe to (or none)
events	MemberSelector	which events to auto-subscribe to (or none)
maxRateHz	f64	optional throttle cap shared across the wired subscriptions

Named by [JsonRpcServer](#).

TlsConfig STRUCTURE

TLS materials. The schema is reserved so configs stay forward-compatible, but the component fails to start if `tls` is set; full TLS support is not yet wired through the `uWebSockets` SSL backend.

Name	Type	Description
certPath	string	PEM-encoded server certificate
keyPath	string	PEM-encoded server private key

Named by [Configuration](#).

TypeLookupResult STRUCTURE

Bundled return of `getType`: always the `CustomTypeSpec`, plus the JSON-Schema fragment when the caller opted in. `schema` is empty when not requested.

Name	Type	Description
spec	sen.kernel.CustomTypeSpec	

Name	Type	Description
schema	string	JSON-encoded schema fragment; "" when withSchema=false

Named by [JsonRpcServer](#).

WildcardSelection STRUCTURE

Wildcard arm of MemberSelector: matches every member of the relevant kind. Wire shape: `{type: "WildcardSelection", value: {}}`.

Named by [MemberSelector](#).

Variant records

[MemberSelector](#) Selector for properties or events in a SubscribeBlock. Absent = no auto-subscribe; WildcardSelection = "*"; NamedSelection = explicit list. 23

MemberSelector VARIANT

Selector for properties or events in a SubscribeBlock. Absent = no auto-subscribe; WildcardSelection = "*"; NamedSelection = explicit list.

Type	Description
WildcardSelection	
NamedSelection	

Named by [SubscribeBlock](#).

Arrays

[AddedObjectList](#) 23
[ObjectInfos](#) 23
[ObjectStateList](#) 23
[PropertyErrorList](#) 23
[PropertyValueList](#) 24
[RegisteredBundleInfoList](#) 24
[SessionInfoList](#) 24
[StaticFileList](#) 24

AddedObjectList SEQUENCE

```
sequence<AddedObjectEntry> AddedObjectList;
```

Named by [JsonRpcServer](#).

ObjectInfos SEQUENCE

```
sequence<ObjectInfo> ObjectInfos;
```

Named by [JsonRpcServer](#).

ObjectStateList SEQUENCE

```
sequence<ObjectState> ObjectStateList;
```

Named by [JsonRpcServer](#).

PropertyErrorList SEQUENCE

```
sequence<PropertyError> PropertyErrorList;
```

Named by [ObjectState](#).

PropertyValueList SEQUENCE

```
sequence<PropertyValuePair> PropertyValueList;
```

Named by [AddedObjectEntry](#), [JsonRpcServer](#), [ObjectState](#).

RegisteredBundleInfoList SEQUENCE

```
sequence<RegisteredBundleInfo> RegisteredBundleInfoList;
```

Named by [StaticFileServer](#).

SessionInfoList SEQUENCE

```
sequence<SessionInfo> SessionInfoList;
```

Named by [JsonRpcServer](#).

StaticFileList SEQUENCE

```
sequence<StaticFile> StaticFileList;
```

Named by [StaticBundle](#).

Optionals

MaybeConnectionLimits	24
MaybeControlBusName	24
MaybeFlag	24
MaybeHighBackpressureBytes	24
MaybeIdleTimeoutSeconds	25
MaybeMaxBackpressureBytes	25
MaybeMaxPayloadBytes	25
MaybeMemberSelector	25
MaybeRateHz	25
MaybeStringList	25
Optional list of object names. Absent on the wire (or `null`) selects every object currently in the addressed interest's match set; present limits the read to the listed names.	
MaybeSubscribeBlock	25
MaybeTlsConfig	25
MaybeUpdateFreqHz	25

MaybeConnectionLimits OPTIONAL

```
optional<ConnectionLimits> MaybeConnectionLimits;
```

MaybeControlBusName OPTIONAL

```
optional<string> MaybeControlBusName;
```

MaybeFlag OPTIONAL

```
optional<bool> MaybeFlag;
```

MaybeHighBackpressureBytes OPTIONAL

```
optional<u32> MaybeHighBackpressureBytes;
```

MaybeIdleTimeoutSeconds OPTIONAL

```
optional<u16> MaybeIdleTimeoutSeconds;
```

MaybeMaxBackpressureBytes OPTIONAL

```
optional<u32> MaybeMaxBackpressureBytes;
```

MaybeMaxPayloadBytes OPTIONAL

```
optional<u32> MaybeMaxPayloadBytes;
```

MaybeMemberSelector OPTIONAL

```
optional<MemberSelector> MaybeMemberSelector;
```

MaybeRateHz OPTIONAL

```
optional<f64> MaybeRateHz;
```

MaybeStringList OPTIONAL

Optional list of object names. Absent on the wire (or `null`) selects every object currently in the addressed interest's match set; present limits the read to the listed names.

```
optional<sen.kernel.StringList> MaybeStringList;
```

MaybeSubscribeBlock OPTIONAL

```
optional<SubscribeBlock> MaybeSubscribeBlock;
```

MaybeTlsConfig OPTIONAL

```
optional<TlsConfig> MaybeTlsConfig;
```

MaybeUpdateFreqHz OPTIONAL

```
optional<f64> MaybeUpdateFreqHz;
```

logmaster

3 types.

Object classes

[LogMaster](#) allows controlling the loggers
[Logger](#) represents a logger

26

26

LogMaster CLASS

allows controlling the loggers

Name	Type	Flags	Description
targetBus	string	RO S BE	

Method	Signature	Description
muteAll	()	mute all the loggers
unmuteAll	()	un-mutes all the muted loggers
toggleMuteAll	()	toggles the muting of all loggers
setLevel	(level: sen.kernel.log.LogLevel)	sets all the loggers to a given level

Logger CLASS

represents a logger

Name	Type	Flags	Description
level	sen.kernel.log.LogLevel	RO D BE	the current log level of this logger
pattern	string	RO D C	the pattern used by this logger
muted	bool	RO D BE	true if muted

Method	Signature	Description
mute	()	mute this logger
unmute	()	unmute this logger
setLevel	(level: sen.kernel.log.LogLevel)	unmute this logger
setPattern	(pattern: string)	sets the pattern for this logger

Fixed records

[Config](#) component configuration

26

Config STRUCTURE

component configuration

Name	Type	Description
targetBus	string	defaults to "local.log"
period	sen.Duration	defaults to 500 milliseconds

py

3 types.

Object classes

[PythonInterpreter](#) represents the embedded Python interpreter 27

PythonInterpreter CLASS

represents the embedded Python interpreter

Method	Signature	Description
eval	(expr: string) → string	evaluates a Python expression
exec	(code: string)	executes a Python code block
execFile	(file: string)	executes a Python file

Fixed records

[Configuration](#) Overall component configuration 27

Configuration STRUCTURE

Overall component configuration

Name	Type	Description
freqHz	f32	update frequency in Hertz
module	string	python module to use (name of a python script, without the .py)
imports	StringList	sen packages to import
bus	string	bus where to publish the interpreter object (none by default)

Arrays

[StringList](#) 27

StringList SEQUENCE

```
sequence<string> StringList;
```

Named by [Configuration](#).

recorder

6 types.

Object classes

[Recorder](#) represents an recording session

28

Recorder CLASS

represents an recording session

Name	Type	Flags	Description
settings	RecordingSettings	RO S BE	
state	RecorderState	RO D C	

Method	Signature	Description
start	()	starts (or resumes) recording
stop	()	stops recording
fetchStats	() → sen.db.OutStats	sample the output statistics

Fixed records

[Configuration](#) Overall component configuration

28

[RecordingSettings](#) Configuration of a predefined recording

28

Configuration STRUCTURE

Overall component configuration

Name	Type	Description
samplingPeriod	sen.Duration	sampling period for the recordings
recordings	RecordingSettingsList	predefined recordings
bus	string	bus to publish the object (local.rec by default)

RecordingSettings STRUCTURE

Configuration of a predefined recording

Name	Type	Description
name	string	name of the recording (cannot be empty)
folder	string	storage folder
selections	SelectionList	which objects shall be recorded
keyframePeriod	sen.Duration	period between keyframes (0 means no keyframes)
indexKeyframes	bool	should we create indexes for keyframes
indexObjects	bool	should we create indexes for objects
autoStart	bool	should we start recording automatically

Named by [Recorder](#), [RecordingSettingsList](#).

Enumerations

[RecorderState](#) the state of a recording session

28

RecorderState ENUMERATION

the state of a recording session

Held as u8.

stopped	0	error	2
recording	1		

Named by Recorder.

Arrays

RecordingSettingsList	List of predefined recordings	29
SelectionList	List of selection queries	29

RecordingSettingsList SEQUENCE

List of predefined recordings

```
sequence<RecordingSettings> RecordingSettingsList;
```

Named by Configuration.

SelectionList SEQUENCE

List of selection queries

```
sequence<string> SelectionList;
```

Named by RecordingSettings.

replayer

4 types.

Object classes

Replay	controls the replay of a given archive	30
Replayer	represents the replayer component and manages Replay objects.	30

Replay CLASS

controls the replay of a given archive

Name	Type	Flags	Description
archiveInfo	sen.db.Summary	RO S BE	information about the archive being replayed
archivePath	string	RO S BE	file path to the archive
status	ReplayStatus	RO D C	playback status
playbackTime	sen.Timestamp	RO D BE	playback time

Method	Signature	Description
play	()	starts the playback from the current position
pause	()	hold the playback position, if playing
stop	()	goes to the start of the playback, erasing all objects
seek	(time: sen.Timestamp)	restores the playback to the keyframe that's closest to a given time
advance	(time: sen.Duration)	if paused or stopped, advances the playback a given time delta

Replayer CLASS

represents the replayer component and manages Replay objects.

Name	Type	Flags	Description
autoPlay	bool	RO S BE	if true, automatically calls play when opening a replay

Method	Signature	Description
open	(name: string , path: string)	opens an archive in a given path, and creates a named Replay object.
close	(name: string)	closes the named Replay object, if any.
closeAll	()	closes all previously opened replays

Fixed records

Configuration	Overall component configuration	30
-------------------------------	---------------------------------	----

Configuration STRUCTURE

Overall component configuration

Name	Type	Description
bus	string	bus to publish the object (local.replay by default)
samplingPeriod	sen.Duration	max sampling period for flushing the replayed data
autoOpen	string	optional archive path to auto open on start-up (empty by default)
autoPlay	bool	if autoOpen is not empty, try to play it on start-up (false by default)

Enumerations

[ReplayStatus](#) the state in which a Replay can be

31

ReplayStatus ENUMERATION

the state in which a Replay can be

Held as u8.

stopped	0
paused	1
playing	2

Named by [Replay](#).

rest

37 types.

Fixed records

Argument	32
Bus	32
BusSummary	32
ClientAuth	32
Configuration	33
Error	33
Interest	33
InterestSummary	33
Link	33
Notification	33
Object	33
ObjectEvent	34
ObjectMethod	34
ObjectSummary	34
Session	34
SessionSummary	34
SubscriptionOptions	34
Subscriptions	34
Success	35
Version	35

Argument STRUCTURE

Name	Type	Description
name	string	
description	string	
type	string	

Named by [Arguments](#).

Bus STRUCTURE

Name	Type	Description
id	string	
status	Status	
objects	ObjectsSummary	

BusSummary STRUCTURE

Name	Type	Description
id	string	
status	Status	

Named by [Buses](#).

ClientAuth STRUCTURE

Name	Type	Description
id	string	

Configuration STRUCTURE

Name	Type	Description
address	string	The IP address or hostname the server will bind to
port	u16	The port number the server will listen on
freqHz	f32	Update frequency in Hertz

Error STRUCTURE

Name	Type	Description
error	string	

Interest STRUCTURE

Name	Type	Description
query	string	
name	string	

InterestSummary STRUCTURE

Name	Type	Description
name	string	
session	string	
bus	string	

Named by [InterestsSummary](#).

Link STRUCTURE

Name	Type	Description
rel	RelType	
href	string	
method	HttpMethod	

Named by [Links](#), [ObjectSummary](#).

Notification STRUCTURE

Name	Type	Description
type	NotificationType	
interest	string	
time	sen.TimeStamp	
data	string	

Object STRUCTURE

Name	Type	Description
objectId	u32	
name	string	
className	string	
localName	string	

Name	Type	Description
description	string	
links	Links	

ObjectEvent STRUCTURE

Name	Type	Description
name	string	
description	string	

ObjectMethod STRUCTURE

Name	Type	Description
name	string	
description	string	
retType	string	
args	Arguments	

ObjectSummary STRUCTURE

Name	Type	Description
objectId	u32	
name	string	
className	string	
localName	string	
link	Link	

Named by [ObjectsSummary](#).

Session STRUCTURE

Name	Type	Description
name	string	

SessionSummary STRUCTURE

Name	Type	Description
name	string	
buses	BusNames	

SubscriptionOptions STRUCTURE

Name	Type	Description
maxUpdateTime	sen.Duration	Max. update time

Subscriptions STRUCTURE

Name	Type	Description
properties	Properties	
events	Events	

Success STRUCTURE

Name	Type	Description
msg	string	

Version STRUCTURE

Name	Type	Description
version	string	

Enumerations

HttpMethod	35
InvokeStatus	35
NotificationType	35
RelType	35
Status	36

HttpMethod ENUMERATION

Held as u8.

httpGet	0
httpPost	1
httpPut	2
httpDelete	3
httpOption	4

Named by [Link](#).

InvokeStatus ENUMERATION

Held as u8.

pending	0
finished	1
failed	2

NotificationType ENUMERATION

Held as u8.

evt	0
invoke	1
property	2
objectAdded	3
objectRemoved	4
count	5

Named by [Notification](#).

RelType ENUMERATION

Held as u8.

method	0
setter	1
getter	2
evt	3
property	4
propertySubscribe	5
propertyUnsubscribe	6
def	7

Named by [Link](#).

Status

ENUMERATION

Held as u8.

open	0
close	1

Named by [Bus](#), [BusSummary](#).

Arrays

Arguments	36
BusNames	36
Buses	36
Events	36
InterestsSummary	36
Links	36
ObjectsSummary	36
Properties	37
Sessions	37

Arguments

SEQUENCE

```
sequence<Argument> Arguments;
```

Named by [ObjectMethod](#).

BusNames

SEQUENCE

```
sequence<string> BusNames;
```

Named by [SessionSummary](#).

Buses

SEQUENCE

```
sequence<BusSummary> Buses;
```

Events

SEQUENCE

```
sequence<string> Events;
```

Named by [Subscriptions](#).

InterestsSummary

SEQUENCE

```
sequence<InterestSummary> InterestsSummary;
```

Links

SEQUENCE

```
sequence<Link> Links;
```

Named by [Object](#).

ObjectsSummary

SEQUENCE

```
sequence<ObjectSummary> ObjectsSummary;
```

Named by [Bus](#).

Properties

SEQUENCE

```
sequence<string> Properties;
```

Named by [Subscriptions](#).

Sessions

SEQUENCE

```
sequence<string> Sessions;
```

Optionals

ThreadPoolSize	37
UpdateFrequency	37
UpdateTime	37

ThreadPoolSize

OPTIONAL

```
optional<u16> ThreadPoolSize;
```

UpdateFrequency

OPTIONAL

```
optional<f32> UpdateFrequency;
```

UpdateTime

OPTIONAL

```
optional<sen.Duration> UpdateTime;
```

shell

27 types.

Object classes

[Shell](#) Provides a text-based interface to interact with the a sen kernel

38

Shell CLASS

Provides a text-based interface to interact with the a sen kernel

Name	Type	Flags	Description
config	Configuration	RO S BE	service configuration

Method	Signature	Description
clear	()	clears the screen
help	()	basic usage help
history	()	lists previous commands
ls	()	lists all the existing objects and sources.
open	(source: string)	opens a source (bus or session) without any condition
query	(name: string , condition: string)	opens a named source using a condition expression
close	(source: string)	closes a previously-opened source (session, bus or query)
info	(term: string)	prints information about a type or instance
shutdown	()	requests the kernel to be shut-down
src	()	lists the current sources

Fixed records

CPrint		38
ClearCurrentLine		39
ClearRemainingCurrentLine		39
ClearScreen		39
Configuration	Configuration params for the shell	39
HideCursor		39
MoveCursorAllLeft		39
MoveCursorDown		39
MoveCursorLeft		39
MoveCursorRight		39
MoveCursorUp		40
NewLine		40
Print		40
Query	Holds information about a named query	40
RestoreCursorPosition		40
SaveCursorPosition		40
SetBgColor		40
SetFgColor		40
SetWindowTitle		40
ShowCursor		41

CPrint STRUCTURE

Name	Type	Description
flags	u32	
color	u32	
text	string	

Named by [TerminalCmd](#).

ClearCurrentLine STRUCTURE

Named by [TerminalCmd](#).

ClearRemainingCurrentLine STRUCTURE

Named by [TerminalCmd](#).

ClearScreen STRUCTURE

Named by [TerminalCmd](#).

Configuration STRUCTURE

Configuration params for the shell

Name	Type	Description
prompt	string	defaults to 'sen> '
timeStyle	TimeStyle	how to display time
bufferStyle	BufferStyle	how to display buffers
serverEnabled	bool	enables remote shell access (false by default)
serverPort	u16	0 means default
noLogo	bool	true if you don't want a welcoming logo
open	SourcesList	list of sources to open automatically
query	QueryList	list of queries to create automatically
logBus	string	where to find the logmaster, if any (default: local.log)

Named by [Shell](#).

HideCursor STRUCTURE

Named by [TerminalCmd](#).

MoveCursorAllLeft STRUCTURE

Named by [TerminalCmd](#).

MoveCursorDown STRUCTURE

Name	Type	Description
cells	u32	

Named by [TerminalCmd](#).

MoveCursorLeft STRUCTURE

Name	Type	Description
cells	u32	

Named by [TerminalCmd](#).

MoveCursorRight STRUCTURE

Name	Type	Description
cells	u32	

Named by [TerminalCmd](#).

MoveCursorUp STRUCTURE

Name	Type	Description
cells	u32	

Named by [TerminalCmd](#).

NewLine STRUCTURE

Named by [TerminalCmd](#).

Print STRUCTURE

Name	Type	Description
text	string	

Named by [TerminalCmd](#).

Query STRUCTURE

Holds information about a named query

Name	Type	Description
name	string	the name of the query (cannot contain spaces or dots)
selection	string	the query expression

Named by [QueryList](#).

RestoreCursorPosition STRUCTURE

Named by [TerminalCmd](#).

SaveCursorPosition STRUCTURE

Named by [TerminalCmd](#).

SetBgColor STRUCTURE

Name	Type	Description
color	u32	

Named by [TerminalCmd](#).

SetFgColor STRUCTURE

Name	Type	Description
color	u32	

Named by [TerminalCmd](#).

SetWindowTitle STRUCTURE

Name	Type	Description
text	string	

Named by [TerminalCmd](#).

ShowCursor STRUCTURE

Named by [TerminalCmd](#).

Enumerations

BufferStyle	How buffers should be printed	41
ListStyle	How lists should be printed	41
TimeStyle	How timestamps should be printed	41

BufferStyle ENUMERATION

How buffers should be printed

Held as u8.

size	0
hexdump	1

Named by [Configuration](#).

ListStyle ENUMERATION

How lists should be printed

Held as u8.

treeStyle	0
plainStyle	1

TimeStyle ENUMERATION

How timestamps should be printed

Held as u8.

timestampUTC	0
timestampLocal	1

Named by [Configuration](#).

Variant records

TerminalCmd	41
-----------------------------	----

TerminalCmd VARIANT

Type	Description
Print	
CPrint	
NewLine	
MoveCursorLeft	
MoveCursorRight	
MoveCursorUp	
MoveCursorDown	
MoveCursorAllLeft	
SetFgColor	
SetBgColor	
ClearCurrentLine	
ClearRemainingCurrentLine	
HideCursor	
ShowCursor	
SaveCursorPosition	

Type	Description
RestoreCursorPosition	
ClearScreen	
SetWindowTitle	

Arrays

QueryList	42
SourcesList	42

QueryList SEQUENCE

```
sequence<Query> QueryList;
```

Named by [Configuration](#).

SourcesList SEQUENCE

```
sequence<string> SourcesList;
```

Named by [Configuration](#).

term

8 types.

Object classes

[Term](#) Modern TUI-based interactive terminal for the sen kernel

43

Term CLASS

Modern TUI-based interactive terminal for the sen kernel

Name	Type	Flags	Description
config	Configuration	RO S BE	service configuration

Fixed records

[Configuration](#) Configuration params for the term component

43

[QueryDef](#) Holds information about a named query

43

Configuration STRUCTURE

Configuration params for the term component

Name	Type	Description
initialScope	string	initial scope, e.g. "local.demo"; its bus must be in open too. Empty means root.
timeStyle	TimeStyle	how to display time (default: utc)
theme	ThemeStyle	color theme (default: catppuccinMocha)
noLogo	bool	true to suppress the welcome banner
open	SourcesList	sources to open automatically at startup
query	QueryList	named queries to create automatically at startup
listen	NameList	event listeners to register at startup (object or object.event)
callTimeout	sen.Duration	async call timeout (default: 30s)

Named by [Term](#).

QueryDef STRUCTURE

Holds information about a named query

Name	Type	Description
name	string	the name of the query (cannot contain spaces or dots)
selection	string	the query expression

Named by [QueryList](#).

Enumerations

[ThemeStyle](#) Color theme for the terminal UI

43

[TimeStyle](#) How timestamps should be displayed

44

ThemeStyle ENUMERATION

Color theme for the terminal UI

Held as u8.

oneDark	0	gruvboxDark	6
oneLight	1	gruvboxLight	7
catppuccinMocha	2	tokyoNight	8
catppuccinLatte	3	solarizedLight	9
dracula	4		
nord	5		

Named by [Configuration](#).

TimeStyle

ENUMERATION

How timestamps should be displayed

Held as u8.

utc	0
local	1

Named by [Configuration](#).

Arrays

NameList	44
QueryList	44
SourcesList	44

NameList

SEQUENCE

```
sequence<string> NameList;
```

Named by [Configuration](#).

QueryList

SEQUENCE

```
sequence<QueryDef> QueryList;
```

Named by [Configuration](#).

SourcesList

SEQUENCE

```
sequence<string> SourcesList;
```

Named by [Configuration](#).

db

3 types.

Fixed records

OutSettings	Settings for outputs	45
OutStats	Statistics of writing	45
Summary	Summary of an archive	45

OutSettings STRUCTURE

Settings for outputs

Name	Type	Description
name	string	name of the archive (cannot be empty)
folder	string	storage folder
indexKeyframes	bool	should we create indexes for keyframes

OutStats STRUCTURE

Statistics of writing

Name	Type	Description
maxQueueSize	u64	in bytes

Named by [sen.components.recorder.Recorder](#).

Summary STRUCTURE

Summary of an archive

Name	Type	Description
firstTime	sen.TimeStamp	the time of the first event
lastTime	sen.TimeStamp	the time of the last event
keyframeCount	u32	number of written keyframes
objectCount	u32	number of recorded objects
typeCount	u32	number of types written
annotationCount	u32	number of annotations
indexedObjectCount	u32	number of indexed objects

Named by [sen.components.replayer.Replay](#).

kernel

160 types.

Object classes

KernelApi	Represents the kernel	46
VirtualClock	Allows inspecting and controlling the advance of virtual time and associated component iteration	46
VirtualKernelClock	A clock for all the components loaded by the kernel	46
VirtualMasterClock	Controls all the VirtualKernelClock that it discovers in the bus where it's published. One step reaches the next component due to run: each clock reports how long until its own next one, and the soonest wins	46

KernelApi CLASS

Represents the kernel

Name	Type	Flags	Description
buildInfo	BuildInfo	RO S BE	kernel build information

Method	Signature	Description
shutdown	()	requests the kernel to shut down
getUnits	() → UnitList	registered units
getTypes	() → StringList	registered types, by name
getConfig	() → KernelParams	configuration of the kernel execution
getRuntimeNetworkFootprint	() → NetworkFootprint	actual network addresses and ports currently used by this process

VirtualClock CLASS

Allows inspecting and controlling the advance of virtual time and associated component iteration

Name	Type	Flags	Description
time	sen.Timestamp	RO D BE	the current virtual time

VirtualKernelClock CLASS

A clock for all the components loaded by the kernel

Method	Signature	Description
processNoFlush	(delta: sen.Duration) → sen.Duration	updates the time, drains the inputs, and cycles components - doesn't flush outputs
flushOutputs	() → sen.Duration	makes component outputs visible to other components and returns the next step duration

VirtualMasterClock CLASS

Controls all the VirtualKernelClock that it discovers in the bus where it's published. One step reaches the next component due to run: each clock reports how long until its own next one, and the soonest wins

Method	Signature	Description
advanceTime	(duration: sen.Duration) → sen.Duration	steps until the virtual time advanced reaches duration, rounding up to a whole step. Returns the wall time the call took
step	() → sen.Duration	single step forward
steps	(count: u64) → sen.Duration	perform 'count' steps forward

Fixed records

AliasTypeSpec	data of an aliased type	48
-------------------------------	-------------------------	----

AliasTypeSpecV4		48
AliasTypeSpecV5		48
ArgSpec	an argument of a callable (methods and events)	48
ArgSpecV4		48
ArgSpecV5		49
BuildInfo	Build-related information	49
BuiltComponentParams	Parameters related to components build by the kernel	49
BusAddress	Address to connect to a bus	49
ClassTypeSpec	spec of a class	49
ClassTypeSpecV4		50
ClassTypeSpecV5		50
ComponentConfig	Basic runtime configuration for a component	50
ComponentInfo	Basic information of a kernel component	50
CustomTypeSpec	spec of a custom type	51
CustomTypeSpecV4		51
CustomTypeSpecV5		51
EnumTypeSpec	spec of an enum type	51
EnumTypeSpecV4		51
EnumTypeSpecV5		52
EnumeratorSpec	an enumerator	52
EnumeratorSpecV4		52
EnumeratorSpecV5		52
EnvVar	Information about an environment variable	52
ErrorData	Information related to the error	52
ErrorReport	Information about a detected error	53
EventSpec	spec for an event	53
EventSpecV4		53
EventSpecV5		53
ExecError	Holds details of an execution error	53
KernelParams	This struct is reachable over the wire through <code>Kernel.getConfig()</code> and is embedded in <code>SenData</code> , and the encoding is positional, so adding a field here shifts what follows it either way.	53
LoadedComponentParams	Parameters related to a component that will be loaded by the kernel	54
MethodSpec	spec for methods	54
MethodSpecV4		54
MethodSpecV5		55
NetworkFootprint	Network endpoints predicted offline or observed at runtime for one process	55
NetworkFootprintAddressRange	Inclusive IPv4 address range excluded from multicast allocation	55
NetworkFootprintBus	Multicast endpoint allocated to a bus	55
NetworkFootprintBusIdentity	Names and IDs that identify a bus	56
NetworkFootprintCollision	Multicast allocation capacity and collision information	56
NetworkFootprintMulticast	Multicast information in the network footprint	56
NetworkFootprintPort	Socket type, port selection mode and port value	56
NetworkFootprintPortExclusions	Port ranges that cannot be used	57
NetworkFootprintPortRange	Inclusive range from which a port may be selected	57
NetworkFootprintSelfCollision	Two buses allocated to the same multicast group	57
OpFinished	An operation is complete	57
OpNotFinished	An operation is not complete	57
OptionalTypeSpec	data of an optional type	57
OptionalTypeSpecV4		58
OptionalTypeSpecV5		58
PrecisionSleep	Be more precise at the expense of some CPU cycles.	58
ProcessData	Information about the process	58
ProcessInfo	Basic information about a process	58
PropertySpec	spec of a property	58
PropertySpecV4		59
PropertySpecV5		59
QuantityTypeSpec	spec of a quantity type	59
QuantityTypeSpecV4		59
QuantityTypeSpecV5		60
QueueConfig		60
SenData	Information about Sen	60
SequenceTypeSpec	spec of a sequence type	60
SequenceTypeSpecV4		61
SequenceTypeSpecV5		61
SignalData	Information about an unhandled signal	61
StructTypeFieldSpec	a field of a struct	61
StructTypeFieldSpecV4		61

StructTypeFieldSpecV5		61
StructTypeSpec	spec of an struct type	62
StructTypeSpecV4		62
StructTypeSpecV5		62
SystemSleep	Use the native system sleep	62
UncaughtException	Information about an unhandled exception	62
UnitInfo	Describes a unit	62
VariantTypeFieldSpec	a field of a variant	63
VariantTypeFieldSpecV4		63
VariantTypeFieldSpecV5		63
VariantTypeSpec	spec of an struct type	63
VariantTypeSpecV4		63
VariantTypeSpecV5		63

AliasTypeSpec STRUCTURE

data of an aliased type

Name	Type	Description
aliasedType	string	qualified type name being aliased

Named by [CustomTypeData](#).

AliasTypeSpecV4 STRUCTURE

Name	Type	Description
aliasedType	string	qualified type name being aliased

Named by [CustomTypeDataV4](#).

AliasTypeSpecV5 STRUCTURE

Name	Type	Description
aliasedType	string	qualified type name being aliased
hash	u32	unique hash

Named by [CustomTypeDataV5](#).

ArgSpec STRUCTURE

an argument of a callable (methods and events)

Name	Type	Description
name	string	argument name
description	string	documentation
type	string	qualified type name of the argument

Named by [ArgSpecList](#).

ArgSpecV4 STRUCTURE

Name	Type	Description
name	string	argument name
description	string	documentation
type	string	qualified type name of the argument

Named by [ArgSpecListV4](#).

ArgSpecV5 STRUCTURE

Name	Type	Description
name	string	argument name
nameHash	u32	hash computed with the argument name
description	string	documentation
type	string	qualified type name of the argument
hash	u32	unique hash

Named by [ArgSpecListV5](#).

BuildInfo STRUCTURE

Build-related information

Name	Type	Description
maintainer	string	principal maintainer of the software
version	string	version string (format-agnostic)
compiler	string	vendor-specific compiler string
debugMode	bool	compiled in debug mode or not
buildTime	string	the commit's timestamp, ISO-8601; not the compiler's clock
wordSize	WordSize	architecture
gitRef	string	git ref spec
gitHash	string	git hash
gitStatus	GitStatus	git status

Named by [ComponentInfo](#), [KernelApi](#).

BuiltComponentParams STRUCTURE

Parameters related to components build by the kernel

Name	Type	Description
name	string	the name of the component
period	sen.Duration	the execution period
config	ComponentConfig	the configuration of the component
imports	StringList	the imported packages

Named by [BuiltComponentList](#).

BusAddress STRUCTURE

Address to connect to a bus

Name	Type	Description
sessionName	string	
busName	string	

ClassTypeSpec STRUCTURE

spec of a class

Name	Type	Description
properties	PropertySpecList	
methods	MethodSpecList	
events	EventSpecList	
constructor	MethodSpec	

Name	Type	Description
parents	StringList	
isInterface	bool	

Named by [CustomTypeData](#).

ClassTypeSpecV4 STRUCTURE

Name	Type	Description
properties	PropertySpecListV4	
methods	MethodSpecListV4	
events	EventSpecListV4	
isInterface	bool	
id	u32	
constructor	MethodSpecV4	
parents	StringList	

Named by [CustomTypeDataV4](#).

ClassTypeSpecV5 STRUCTURE

Name	Type	Description
properties	PropertySpecListV5	
methods	MethodSpecListV5	
events	EventSpecListV5	
isInterface	bool	
id	u32	
constructor	MethodSpecV5	
parents	StringList	
hash	u32	

Named by [CustomTypeDataV5](#).

ComponentConfig STRUCTURE

Basic runtime configuration for a component

Name	Type	Description
priority	Priority	thread priority
stackSize	u32	thread stack size in bytes, 0 means default
group	u32	the group where to run the component
cpuAffinity	u64	CPU affinity mask
inQueue	QueueConfig	queuing of inbound information
outQueue	QueueConfig	queuing of outbound information
sleepPolicy	SleepPolicy	configurable component sleep policy

Named by [BuiltComponentParams](#), [LoadedComponentParams](#).

ComponentInfo STRUCTURE

Basic information of a kernel component

Name	Type	Description
name	string	
description	string	

Name	Type	Description
buildInfo	BuildInfo	

CustomTypeSpec STRUCTURE

spec of a custom type

Name	Type	Description
name	string	name of the type
qualifiedName	string	package-qualified name
description	string	documentation
data	CustomTypeData	spec data

Named by [sen.components.jsonrpc.TypeLookupResult](#), [CustomTypeSpecList](#).

CustomTypeSpecV4 STRUCTURE

Name	Type	Description
name	string	name of the type
qualifiedName	string	package-qualified name
description	string	documentation
data	CustomTypeDataV4	spec data

Named by [CustomTypeSpecListV4](#).

CustomTypeSpecV5 STRUCTURE

Name	Type	Description
name	string	name of the type
qualifiedName	string	package-qualified name
description	string	documentation
data	CustomTypeDataV5	spec data

Named by [CustomTypeSpecListV5](#).

EnumTypeSpec STRUCTURE

spec of an enum type

Name	Type	Description
enums	EnumeratorSpecList	enumerators
storageType	IntegralType	type used to store the value

Named by [CustomTypeData](#).

EnumTypeSpecV4 STRUCTURE

Name	Type	Description
enums	EnumeratorSpecListV4	enumerators
storageType	IntegralType	type used to store the value

Named by [CustomTypeDataV4](#).

EnumTypeSpecV5 STRUCTURE

Name	Type	Description
enums	EnumeratorSpecListV5	enumerators
storageType	IntegralType	type used to store the value
hash	u32	unique hash

Named by [CustomTypeDataV5](#).

EnumeratorSpec STRUCTURE

an enumerator

Name	Type	Description
name	string	enumerator name
key	u32	enumerator value
description	string	enumerator description

Named by [EnumeratorSpecList](#).

EnumeratorSpecV4 STRUCTURE

Name	Type	Description
name	string	enumerator name
key	u64	enumerator value

Named by [EnumeratorSpecListV4](#).

EnumeratorSpecV5 STRUCTURE

Name	Type	Description
name	string	enumerator name
key	u64	enumerator value
hash	u32	unique hash

Named by [EnumeratorSpecListV5](#).

EnvVar STRUCTURE

Information about an environment variable

Name	Type	Description
name	string	name of the variable
value	string	value of the variable

Named by [EnvironmentVarList](#).

ErrorData STRUCTURE

Information related to the error

Name	Type	Description
time	sen.TimeStamp	time of the error
errorMessage	StringList	message presented to the user
exceptionData	UncaughtException	information about the unhandled exception (if any)
signalData	SignalData	information about the received signal (if any)

Named by [ErrorReport](#).

ErrorReport STRUCTURE

Information about a detected error

Name	Type	Description
senData	SenData	information related to Sen
processData	ProcessData	information related to the process
errorData	ErrorData	information related to the error

EventSpec STRUCTURE

spec for an event

Name	Type	Description
name	string	name of the type
description	string	documentation
args	ArgSpecList	arguments
transportMode	TransportModeSpec	transport mode

Named by [EventSpecList](#).

EventSpecV4 STRUCTURE

Name	Type	Description
memberHash	u32	hash of the class member
name	string	name of the type
description	string	documentation
args	ArgSpecListV4	arguments
transportMode	TransportModeSpec	transport mode

Named by [EventSpecListV4](#).

EventSpecV5 STRUCTURE

Name	Type	Description
memberHash	u32	hash of the class member
name	string	name of the type
description	string	documentation
args	ArgSpecListV5	arguments
transportMode	TransportModeSpec	transport mode
hash	u32	unique hash

Named by [EventSpecListV5](#).

ExecError STRUCTURE

Holds details of an execution error

Name	Type	Description
category	ErrorCategory	general problem
explanation	string	human-readable explanation

KernelParams STRUCTURE

This struct is reachable over the wire through `Kernel.getConfig()` and is embedded in `SenData`, and the encoding is positional, so adding a field here shifts what follows it either way.

Name	Type	Description
runMode	RunMode	how to run the kernel
appName	string	optional name of the application
bus	string	where to publish the kernel objects (defaults to local.kernel)
clockBus	string	where to publish the virtual clock (if any), defaults to bus
clockName	string	the name of the virtual clock (if any), defaults to 'clock'
clockMaster	bool	if time is virtualized, publish a master clock to clockBus
logConfig	log.Config	logging configuration
crashReportDir	string	where to store crash reports (defaults to the temp dir)
crashReportDisabled	bool	if true, no reports are generated
lockMemoryPages	bool	keep process pages memory-resident
sleepPolicy	SleepPolicy	configurable sleep policy of the kernel component
compatibilityMode	CompatibilityMode	what to do with a remote type that does not match
limitCpuIdleLatency	bool	keep the CPUs out of the deep idle states they are slow to leave

Named by [KernelApi](#), [SenData](#).

LoadedComponentParams STRUCTURE

Parameters related to a component that will be loaded by the kernel

Name	Type	Description
name	string	the name of the component
config	ComponentConfig	the configuration of the component

Named by [LoadedComponentList](#).

MethodSpec STRUCTURE

spec for methods

Name	Type	Description
name	string	name of the type
description	string	documentation
args	ArgSpecList	arguments
transportMode	TransportModeSpec	transport mode
constness	MethodConstnessSpec	if it changes the state of the object
deferred	bool	if the execution of the method is deferred
returnType	string	qualified return type name
propertyRelation	PropertyRelationSpec	property relation of the method
localOnly	bool	true if the method can only be called locally

Named by [ClassTypeSpec](#), [MethodSpecList](#).

MethodSpecV4 STRUCTURE

Name	Type	Description
memberHash	u32	hash of the class member
name	string	name of the type
description	string	documentation
args	ArgSpecListV4	arguments
transportMode	TransportModeSpec	transport mode

Name	Type	Description
constness	MethodConstnessSpec	if it changes the state of the object
deferred	bool	true if the method is marked as deferred
returnTypeId	string	qualified return type name (empty for none)

Named by [ClassTypeSpecV4](#), [MethodSpecListV4](#).

MethodSpecV5 STRUCTURE

Name	Type	Description
memberHash	u32	hash of the class member
name	string	name of the type
description	string	documentation
args	ArgSpecListV5	arguments
transportMode	TransportModeSpec	transport mode
constness	MethodConstnessSpec	if it changes the state of the object
localOnly	bool	true if the method can only be called locally
returnTypeId	string	qualified return type name (empty for none)
hash	u32	unique hash

Named by [ClassTypeSpecV5](#), [MethodSpecListV5](#).

NetworkFootprint STRUCTURE

Network endpoints predicted offline or observed at runtime for one process

Name	Type	Description
discoveryPort	u16	Discovery port used to calculate multicast addresses
multicast	NetworkFootprintMulticast	Multicast allocation details
ports	NetworkFootprintPortList	Port settings
portExclusions	NetworkFootprintPortExclusions	Port ranges that cannot be used

Named by [KernelApi](#).

NetworkFootprintAddressRange STRUCTURE

Inclusive IPv4 address range excluded from multicast allocation

Name	Type	Description
min	string	
max	string	

Named by [NetworkFootprintAddressRangeList](#).

NetworkFootprintBus STRUCTURE

Multicast endpoint allocated to a bus

Name	Type	Description
sessionName	string	
busName	string	
sessionId	u32	

Name	Type	Description
busId	u32	
groupAddress	string	
source	NetworkFootprintBus Source	

Named by [NetworkFootprintBusList](#).

NetworkFootprintBusIdentity STRUCTURE

Names and IDs that identify a bus

Name	Type	Description
sessionName	string	
busName	string	
sessionId	u32	
busId	u32	

Named by [NetworkFootprintSelfCollision](#).

NetworkFootprintCollision STRUCTURE

Multicast allocation capacity and collision information

Name	Type	Description
usableSpaceSize	u64	Number of addresses available
busCount	u64	Number of reported buses
probability	f64	Estimated conflict from 0 to 1
selfCollisions	NetworkFootprintSelf CollisionList	Collisions in the reported buses

Named by [NetworkFootprintMulticast](#).

NetworkFootprintMulticast STRUCTURE

Multicast information in the network footprint

Name	Type	Description
port	u16	Shared multicast port used by every bus
buses	NetworkFootprintBus List	Multicast address used by each bus
exclusions	NetworkFootprint AddressRangeList	Addresses that cannot be used
collision	NetworkFootprint Collision	Available addresses and conflict information

Named by [NetworkFootprint](#).

NetworkFootprintPort STRUCTURE

Socket type, port selection mode and port value

Name	Type	Description
kind	NetworkFootprintPort Kind	
mode	NetworkFootprintPort Mode	
value	NetworkFootprintPort Value	

Named by [NetworkFootprintPortList](#).

NetworkFootprintPortExclusions STRUCTURE

Port ranges that cannot be used

Name	Type	Description
builtIn	NetworkFootprintPortRangeList	
configured	NetworkFootprintPortRangeList	
operatingSystem	NetworkFootprintPortRangeList	

Named by [NetworkFootprint](#).

NetworkFootprintPortRange STRUCTURE

Inclusive range from which a port may be selected

Name	Type	Description
min	u16	
max	u16	

Named by [NetworkFootprintPortRangeList](#), [NetworkFootprintPortValue](#).

NetworkFootprintSelfCollision STRUCTURE

Two buses allocated to the same multicast group

Name	Type	Description
firstBus	NetworkFootprintBusIdentity	
secondBus	NetworkFootprintBusIdentity	
groupAddress	string	

Named by [NetworkFootprintSelfCollisionList](#).

OpFinished STRUCTURE

An operation is complete

Named by [OpState](#).

OpNotFinished STRUCTURE

An operation is not complete

Name	Type	Description
waitHint	sen.Duration	How long to wait until the next try

Named by [OpState](#).

OptionalTypeSpec STRUCTURE

data of an optional type

Name	Type	Description
type	string	qualified type name

Named by [CustomTypeData](#).

OptionalTypeSpecV4 STRUCTURE

Name	Type	Description
type	string	qualified type name

Named by [CustomTypeDataV4](#).

OptionalTypeSpecV5 STRUCTURE

Name	Type	Description
type	string	qualified type name
hash	u32	unique hash

Named by [CustomTypeDataV5](#).

PrecisionSleep STRUCTURE

Be more precise at the expense of some CPU cycles.

Name	Type	Description
veryCoarseGrainSleepTime	sen.Duration	First set of sleeps. If 0 defaults to 7ms
coarseGrainSleepTime	sen.Duration	Second set of sleeps. If 0 defaults to 1ms

Named by [SleepPolicy](#).

ProcessData STRUCTURE

Information about the process

Name	Type	Description
processInfo	ProcessInfo	general process information
environment	EnvironmentVarList	list of environment variables
stacktrace	StringList	the stack trace (if any)
recentLogs	StringList	the last logs emitted by the process

Named by [ErrorReport](#).

ProcessInfo STRUCTURE

Basic information about a process

Name	Type	Description
hostId	u32	
processId	u32	
sessionId	u32	
sessionName	string	
appName	string	
hostName	string	
osKind	OsKind	
osName	string	
cpuArch	CpuArch	

Named by [sen.components.ether.Hello](#), [sen.components.ether.SessionPresenceBeam](#), [ProcessData](#), [Processes](#).

PropertySpec STRUCTURE

spec of a property

Name	Type	Description
name	string	name of the type
description	string	documentation
category	PropertyCategorySpec	category
type	string	qualified type name of the property
transportMode	TransportModeSpec	how to transport the data
tags	StringList	user-defined tags
checkedSet	bool	if setting the property is checked

Named by [PropertySpecList](#).

PropertySpecV4 STRUCTURE

Name	Type	Description
memberHash	u32	hash of the class member
name	string	name of the type
description	string	documentation
category	PropertyCategorySpec	category
type	string	qualified type name of the property
transportMode	TransportModeSpec	how to transport the data
tags	StringList	user-defined tags

Named by [PropertySpecListV4](#).

PropertySpecV5 STRUCTURE

Name	Type	Description
memberHash	u32	hash of the class member
name	string	name of the type
description	string	documentation
category	PropertyCategorySpec	category
type	string	qualified type name of the property
transportMode	TransportModeSpec	how to transport the data
tags	StringList	user-defined tags
hash	u32	unique hash

Named by [PropertySpecListV5](#).

QuantityTypeSpec STRUCTURE

spec of a quantity type

Name	Type	Description
elementType	NumericType	type used to store the value
unit	UnitInfo	used unit
minValue	f64	discarded if $\geq$ maxValue
maxValue	f64	discarded if $\leq$ minValue

Named by [CustomTypeData](#).

QuantityTypeSpecV4 STRUCTURE

Name	Type	Description
numericType	NumericType	type used to store the value

Name	Type	Description
unit	UnitInfo	used unit
minValue	f64	discarded if >= maxValue
maxValue	f64	discarded if <= minValue

Named by [CustomTypeDataV4](#).

QuantityTypeSpecV5 STRUCTURE

Name	Type	Description
numericType	NumericType	type used to store the value
unit	UnitInfo	used unit
minValue	f64	discarded if >= maxValue
maxValue	f64	discarded if <= minValue
hash	u32	unique hash

Named by [CustomTypeDataV5](#).

QueueConfig STRUCTURE

Name	Type	Description
evictionPolicy	QueueEvictionPolicy	what to do when the queue is full
maxSize	u64	0 means unbounded

Named by [ComponentConfig](#).

SenData STRUCTURE

Information about Sen

Name	Type	Description
version	string	the Sen version
compiler	string	the compiler used to build Sen
debugMode	bool	true if it was build in debug mode
buildTime	string	the commit's timestamp, not the compiler's clock
wordSize	WordSize	architecture
gitRef	string	git ref
gitHash	string	git hash
gitStatus	GitStatus	status of the branch when build
kernelParams	KernelParams	Sen kernel parameters
kernelProtocol	u32	version of the kernel protocol
transportProtocol	u32	version of the transport protocol (if present)
loadedComponents	LoadedComponentList	components to load
builtComponents	BuiltComponentList	components to build

Named by [ErrorReport](#).

SequenceTypeSpec STRUCTURE

spec of a sequence type

Name	Type	Description
elementType	string	qualified type name for stored values
maxSize	u64	discarded when 0
fixedSize	bool	

Named by [CustomTypeData](#).

SequenceTypeSpecV4 STRUCTURE

Name	Type	Description
elementType	string	qualified type name for stored values
maxSize	u64	discarded when 0

Named by [CustomTypeDataV4](#).

SequenceTypeSpecV5 STRUCTURE

Name	Type	Description
elementType	string	qualified type name for stored values
maxSize	u64	discarded when 0
fixedSize	bool	
hash	u32	unique hash

Named by [CustomTypeDataV5](#).

SignalData STRUCTURE

Information about an unhandled signal

Name	Type	Description
signalNumber	i32	signal number
signalName	string	human-readable name

Named by [ErrorData](#).

StructTypeFieldSpec STRUCTURE

a field of a struct

Name	Type	Description
name	string	field name
description	string	documentation
type	string	qualified type name held by this field

Named by [StructTypeFieldSpecList](#).

StructTypeFieldSpecV4 STRUCTURE

Name	Type	Description
name	string	field name
description	string	documentation
type	string	qualified type name held by this field

Named by [StructTypeFieldSpecListV4](#).

StructTypeFieldSpecV5 STRUCTURE

Name	Type	Description
name	string	field name
description	string	documentation
type	string	qualified type name held by this field

Name	Type	Description
hash	u32	unique hash

Named by [StructTypeFieldSpecListV5](#).

StructTypeSpec STRUCTURE

spec of an struct type

Name	Type	Description
fields	StructTypeFieldSpecList	fields
parent	string	qualified type name the parent struct, empty means none

Named by [CustomTypeData](#).

StructTypeSpecV4 STRUCTURE

Name	Type	Description
fields	StructTypeFieldSpecListV4	fields
parent	string	qualified type name the parent struct, empty means none

Named by [CustomTypeDataV4](#).

StructTypeSpecV5 STRUCTURE

Name	Type	Description
fields	StructTypeFieldSpecListV5	fields
parent	string	qualified type name the parent struct, empty means none
hash	u32	unique hash

Named by [CustomTypeDataV5](#).

SystemSleep STRUCTURE

Use the native system sleep

Named by [SleepPolicy](#).

UncaughtException STRUCTURE

Information about an unhandled exception

Name	Type	Description
exceptionKind	ExceptionKind	kind of exception
message	string	message it provides (if any)

Named by [ErrorData](#).

UnitInfo STRUCTURE

Describes a unit

Name	Type	Description
name	string	
abbreviation	string	
category	UnitCat	

Named by [QuantityTypeSpec](#), [QuantityTypeSpecV4](#), [QuantityTypeSpecV5](#), [UnitList](#).

VariantTypeFieldSpec STRUCTURE

a field of a variant

Name	Type	Description
key	u32	key/index of this type within the variant
description	string	documentation
type	string	qualified type name held by this field

Named by [VariantTypeFieldSpecList](#).

VariantTypeFieldSpecV4 STRUCTURE

Name	Type	Description
key	u32	key/index of this type within the variant
description	string	documentation
type	string	qualified type name held by this field

Named by [VariantTypeFieldSpecListV4](#).

VariantTypeFieldSpecV5 STRUCTURE

Name	Type	Description
key	u32	key/index of this type within the variant
description	string	documentation
type	string	qualified type name held by this field
hash	u32	unique hash

Named by [VariantTypeFieldSpecListV5](#).

VariantTypeSpec STRUCTURE

spec of an struct type

Name	Type	Description
fields	VariantTypeFieldSpecList	

Named by [CustomTypeData](#).

VariantTypeSpecV4 STRUCTURE

Name	Type	Description
fields	VariantTypeFieldSpecListV4	

Named by [CustomTypeDataV4](#).

VariantTypeSpecV5 STRUCTURE

Name	Type	Description
fields	VariantTypeFieldSpecListV5	
hash	u32	

Named by [CustomTypeDataV5](#).

Enumerations

BasicType	built-in others	64
CompatibilityMode	What to do when a type published by another kernel does not match the local one	64
ComponentState	State of a kernel component	64
CpuArch	Type of CPU	65
ErrorCategory	Category of a problem during component execution	65
ExceptionKind	Types of exceptions that can occur.	65
GitStatus	Status of git	65
IntegralType	built-in integrals	65
MethodConstnessSpec	method constness	66
NetworkFootprintBusSource	Bus source	66
NetworkFootprintPortKind	Type of socket opened by the process	66
NetworkFootprintPortMode	Port allocation policy	66
OsKind	Type of operating system	66
Priority	Thread execution priorities	67
PropertyCategorySpec	how a property is seen by others	67
PropertyRelationSpec	method property relation	67
QueueEvictionPolicy	Component queue eviction policy	67
RealType	built-in reals	67
RunMode	Run modes for the kernel	67
ThreadCreateErr	What can go wrong when creating a thread	68
TransportModeSpec	how to transport information	68
UnitCat	Type of measurement	68
WordSize	Native processor word size	68

BasicType ENUMERATION

built-in others

Held as u8.

booleanType	0
stringType	1
durationType	2
timestampType	3

Named by [BuiltInType](#).

CompatibilityMode ENUMERATION

What to do when a type published by another kernel does not match the local one

Held as u8.

relaxed	0
strict	1
disabled	2

Named by [KernelParams](#).

ComponentState ENUMERATION

State of a kernel component

Held as u8.

unloaded	0
preloaded	1
loading	2
loaded	3
initializing	4
initialized	5
running	6
stopping	7
stopped	8
unloading	9
error	10

CpuArch ENUMERATION

Type of CPU

Held as u8.

x86	0
x64	1
arm2	2
arm3	3
arm4T	4
arm5	5
arm6T2	6
arm6	7
arm7	8
arm7a	9
arm7r	10
arm7s	11
arm64	12
mips	13
superH	14
ppc	15
ppc64	16
sparc	17
m68k	18
otherArch	19

Named by [ProcessInfo](#).

ErrorCategory ENUMERATION

Category of a problem during component execution

Held as u8.

runtimeError	0
logicError	1
expectationsNotMet	2
ioError	3
other	4

Named by [ExecError](#).

ExceptionKind ENUMERATION

Types of exceptions that can occur.

Held as u8.

runtime	0
logic	1
standard	2
unknown	3

Named by [UncaughtException](#).

GitStatus ENUMERATION

Status of git

Held as u8.

clean	0
modified	1
unknown	2

Named by [BuildInfo](#), [SenData](#).

IntegralType ENUMERATION

built-in integrals

Held as u8.

uint8Type	0
int16Type	1
uint16Type	2
int32Type	3
uint32Type	4
int64Type	5
uint64Type	6

Named by [EnumTypeSpec](#), [EnumTypeSpecV4](#), [EnumTypeSpecV5](#), [NumericType](#).

MethodConstnessSpec ENUMERATION

method constness

Held as u8.

constant	0
nonConstant	1

Named by [MethodSpec](#), [MethodSpecV4](#), [MethodSpecV5](#).

NetworkFootprintBusSource ENUMERATION

Bus source

Held as u8.

configured	0
supplied	1
runtime	2

Named by [NetworkFootprintBus](#).

NetworkFootprintPortKind ENUMERATION

Type of socket opened by the process

Held as u8.

tcpAcceptor	0
udpUnicast	1
tcpSource	2

Named by [NetworkFootprintPort](#).

NetworkFootprintPortMode ENUMERATION

Port allocation policy

Held as u8.

ephemeral	0
probe	1
pinned	2

Named by [NetworkFootprintPort](#).

OsKind ENUMERATION

Type of operating system

Held as u8.

windowsOs	0
linuxOs	1
androidOs	2
appleOs	3
unixOs	4
posixOs	5
otherOs	6

Named by [ProcessInfo](#).

Priority ENUMERATION

Thread execution priorities

Held as u32.

lowest	0
nominalMin	1
nominalMax	2
highest	3

Named by [ComponentConfig](#).

PropertyCategorySpec ENUMERATION

how a property is seen by others

Held as u8.

staticRW	0
staticRO	1
dynamicRW	2
dynamicRO	3

Named by [PropertySpec](#), [PropertySpecV4](#), [PropertySpecV5](#).

PropertyRelationSpec ENUMERATION

method property relation

Held as u8.

nonPropertyRelated	0
propertyGetter	1
propertySetter	2

Named by [MethodSpec](#).

QueueEvictionPolicy ENUMERATION

Component queue eviction policy

Held as u32.

dropOldest	0
dropNewest	1

Named by [QueueConfig](#).

RealType ENUMERATION

built-in reals

Held as u8.

float32Type	0
float64Type	1

Named by [NumericType](#).

RunMode ENUMERATION

Run modes for the kernel

Held as u8.

realTime	0
virtualTime	1
virtualTimeRunning	2
startAndStop	3

Named by [KernelParams](#).

ThreadCreateErr ENUMERATION

What can go wrong when creating a thread

Held as u32.

schedulerAlreadyStarted	0
invalidStackSize	1
invalidThreadFunction	2
invalidThreadFunctionArgument	3
invalidAffinity	4
tooManyThreads	5
internalOsError	6

TransportModeSpec ENUMERATION

how to transport information

Held as u8.

unicast	0
multicast	1
confirmed	2

Named by [EventSpec](#), [EventSpecV4](#), [EventSpecV5](#), [MethodSpec](#), [MethodSpecV4](#), [MethodSpecV5](#), [PropertySpec](#), [PropertySpecV4](#), [PropertySpecV5](#).

UnitCat ENUMERATION

Type of measurement

Held as u8.

length	0
mass	1
time	2
angle	3
temperature	4
frequency	5
velocity	6
angularVelocity	7
acceleration	8
angularAcceleration	9
density	10
pressure	11
area	12
force	13
torque	14

Named by [UnitInfo](#).

WordSize ENUMERATION

Native processor word size

Held as u32.

bits32	0
bits64	1

Named by [BuildInfo](#), [SenData](#).

Variant records

BuiltInType	all built-in types	69
CustomTypeData	all custom types	69
CustomTypeDataV4		69
CustomTypeDataV5		69
NetworkFootprintPortValue	Known port number or range	69
NumericType	all built-in numeric types	70
OpState	The status of an operation	70
SleepPolicy	Component sleep policy	70

BuiltInType VARIANT

all built-in types

Type	Description
NumericType	
BasicType	

CustomTypeData VARIANT

all custom types

Type	Description
EnumTypeSpec	
QuantityTypeSpec	
SequenceTypeSpec	
StructTypeSpec	
VariantTypeSpec	
AliasTypeSpec	
OptionalTypeSpec	
ClassTypeSpec	

Named by [CustomTypeSpec](#).

CustomTypeDataV4 VARIANT

Type	Description
EnumTypeSpecV4	
QuantityTypeSpecV4	
SequenceTypeSpecV4	
StructTypeSpecV4	
VariantTypeSpecV4	
AliasTypeSpecV4	
OptionalTypeSpecV4	
ClassTypeSpecV4	

Named by [CustomTypeSpecV4](#).

CustomTypeDataV5 VARIANT

Type	Description
EnumTypeSpecV5	
QuantityTypeSpecV5	
SequenceTypeSpecV5	
StructTypeSpecV5	
VariantTypeSpecV5	
AliasTypeSpecV5	
OptionalTypeSpecV5	
ClassTypeSpecV5	

Named by [CustomTypeSpecV5](#).

NetworkFootprintPortValue VARIANT

Known port number or range

Type	Description
u16	
NetworkFootprintPortRange	

Named by [NetworkFootprintPort](#).

NumericType VARIANT

all built-in numeric types

Type	Description
IntegralType	
RealType	

Named by [BuiltInType](#), [QuantityTypeSpec](#), [QuantityTypeSpecV4](#), [QuantityTypeSpecV5](#).

OpState VARIANT

The status of an operation

Type	Description
OpFinished	done
OpNotFinished	not done

SleepPolicy VARIANT

Component sleep policy

Type	Description
PrecisionSleep	
SystemSleep	

Named by [ComponentConfig](#), [KernelParams](#).

Arrays

ArgSpecList	list of argument specs	71
ArgSpecListV4		71
ArgSpecListV5		71
Buffer	A general-purpose unbounded buffer	71
BuiltComponentList		71
CustomTypeSpecList	list of custom types	71
CustomTypeSpecListV4		71
CustomTypeSpecListV5		71
EnumeratorSpecList	list of enumerators	72
EnumeratorSpecListV4		72
EnumeratorSpecListV5		72
EnvironmentVarList		72
EventSpecList	list of event specs	72
EventSpecListV4		72
EventSpecListV5		72
LoadedComponentList		72
MethodSpecList	list of method specs	72
MethodSpecListV4		73
MethodSpecListV5		73
NetworkFootprintAddressRangeList		73
NetworkFootprintBusList	Buses included in a network footprint	73
NetworkFootprintPortList	Ports used by the process	73
NetworkFootprintPortRangeList		73
NetworkFootprintSelfCollisionList	Multicast address conflicts found in the reported buses	73
Processes	Unbounded sequence of ProcessInfo	73
PropertySpecList	list of property specs	73
PropertySpecListV4		74

PropertySpecListV5		74
StringList	Unbounded sequence of strings for general purpose	74
StructTypeFieldSpecList	list of struct fields	74
StructTypeFieldSpecListV4		74
StructTypeFieldSpecListV5		74
U32List		74
UnitList	Unbounded sequence of unit information structs	74
VariantTypeFieldSpecList	list of variant fields	74
VariantTypeFieldSpecListV4		75
VariantTypeFieldSpecListV5		75

ArgSpecList SEQUENCE

list of argument specs

```
sequence<ArgSpec> ArgSpecList;
```

Named by [EventSpec](#), [MethodSpec](#).

ArgSpecListV4 SEQUENCE

```
sequence<ArgSpecV4> ArgSpecListV4;
```

Named by [EventSpecV4](#), [MethodSpecV4](#).

ArgSpecListV5 SEQUENCE

```
sequence<ArgSpecV5> ArgSpecListV5;
```

Named by [EventSpecV5](#), [MethodSpecV5](#).

Buffer SEQUENCE

A general-purpose unbounded buffer

```
sequence<u8> Buffer;
```

Named by [sen.components.jsonrpc.StaticFile](#).

BuiltComponentList SEQUENCE

```
sequence<BuiltComponentParams> BuiltComponentList;
```

Named by [SenData](#).

CustomTypeSpecList SEQUENCE

list of custom types

```
sequence<CustomTypeSpec> CustomTypeSpecList;
```

Named by [sen.components.jsonrpc.JsonRpcServer](#).

CustomTypeSpecListV4 SEQUENCE

```
sequence<CustomTypeSpecV4> CustomTypeSpecListV4;
```

CustomTypeSpecListV5 SEQUENCE

```
sequence<CustomTypeSpecV5> CustomTypeSpecListV5;
```

EnumeratorSpecList SEQUENCE

list of enumerators

```
sequence<EnumeratorSpec> EnumeratorSpecList;
```

Named by [EnumTypeSpec](#).

EnumeratorSpecListV4 SEQUENCE

```
sequence<EnumeratorSpecV4> EnumeratorSpecListV4;
```

Named by [EnumTypeSpecV4](#).

EnumeratorSpecListV5 SEQUENCE

```
sequence<EnumeratorSpecV5> EnumeratorSpecListV5;
```

Named by [EnumTypeSpecV5](#).

EnvironmentVarList SEQUENCE

```
sequence<EnvVar> EnvironmentVarList;
```

Named by [ProcessData](#).

EventSpecList SEQUENCE

list of event specs

```
sequence<EventSpec> EventSpecList;
```

Named by [ClassTypeSpec](#).

EventSpecListV4 SEQUENCE

```
sequence<EventSpecV4> EventSpecListV4;
```

Named by [ClassTypeSpecV4](#).

EventSpecListV5 SEQUENCE

```
sequence<EventSpecV5> EventSpecListV5;
```

Named by [ClassTypeSpecV5](#).

LoadedComponentList SEQUENCE

```
sequence<LoadedComponentParams> LoadedComponentList;
```

Named by [SenData](#).

MethodSpecList SEQUENCE

list of method specs

```
sequence<MethodSpec> MethodSpecList;
```

Named by [ClassTypeSpec](#).

MethodSpecListV4 SEQUENCE

```
sequence<MethodSpecV4> MethodSpecListV4;
```

Named by [ClassTypeSpecV4](#).

MethodSpecListV5 SEQUENCE

```
sequence<MethodSpecV5> MethodSpecListV5;
```

Named by [ClassTypeSpecV5](#).

NetworkFootprintAddressRangeList SEQUENCE

```
sequence<NetworkFootprintAddressRange> NetworkFootprintAddressRangeList;
```

Named by [NetworkFootprintMulticast](#).

NetworkFootprintBusList SEQUENCE

Buses included in a network footprint

```
sequence<NetworkFootprintBus> NetworkFootprintBusList;
```

Named by [NetworkFootprintMulticast](#).

NetworkFootprintPortList SEQUENCE

Ports used by the process

```
sequence<NetworkFootprintPort> NetworkFootprintPortList;
```

Named by [NetworkFootprint](#).

NetworkFootprintPortRangeList SEQUENCE

```
sequence<NetworkFootprintPortRange> NetworkFootprintPortRangeList;
```

Named by [NetworkFootprintPortExclusions](#).

NetworkFootprintSelfCollisionList SEQUENCE

Multicast address conflicts found in the reported buses

```
sequence<NetworkFootprintSelfCollision> NetworkFootprintSelfCollisionList;
```

Named by [NetworkFootprintCollision](#).

Processes SEQUENCE

Unbounded sequence of ProcessInfo

```
sequence<ProcessInfo> Processes;
```

PropertySpecList SEQUENCE

list of property specs

```
sequence<PropertySpec> PropertySpecList;
```

Named by [ClassTypeSpec](#).

PropertySpecListV4 SEQUENCE

```
sequence<PropertySpecV4> PropertySpecListV4;
```

Named by [ClassTypeSpecV4](#).

PropertySpecListV5 SEQUENCE

```
sequence<PropertySpecV5> PropertySpecListV5;
```

Named by [ClassTypeSpecV5](#).

StringList SEQUENCE

Unbounded sequence of strings for general purpose

```
sequence<string> StringList;
```

Named by [sen.components.jsonrpc.JsonRpcServer](#), [sen.components.jsonrpc.NamedSelection](#), [sen.components.jsonrpc.SessionInfo](#), [BuiltComponentParams](#), [ClassTypeSpec](#), [ClassTypeSpecV4](#), [ClassTypeSpecV5](#), [ErrorData](#), [KernelApi](#), [ProcessData](#), [PropertySpec](#), [PropertySpecV4](#), [PropertySpecV5](#).

StructTypeFieldSpecList SEQUENCE

list of struct fields

```
sequence<StructTypeFieldSpec> StructTypeFieldSpecList;
```

Named by [StructTypeSpec](#).

StructTypeFieldSpecListV4 SEQUENCE

```
sequence<StructTypeFieldSpecV4> StructTypeFieldSpecListV4;
```

Named by [StructTypeSpecV4](#).

StructTypeFieldSpecListV5 SEQUENCE

```
sequence<StructTypeFieldSpecV5> StructTypeFieldSpecListV5;
```

Named by [StructTypeSpecV5](#).

U32List SEQUENCE

```
sequence<u32> U32List;
```

UnitList SEQUENCE

Unbounded sequence of unit information structs

```
sequence<UnitInfo> UnitList;
```

Named by [KernelApi](#).

VariantTypeFieldSpecList SEQUENCE

list of variant fields

```
sequence<VariantTypeFieldSpec> VariantTypeFieldSpecList;
```

Named by [VariantTypeSpec](#).

VariantTypeFieldSpecListV4

SEQUENCE

```
sequence<VariantTypeFieldSpecV4> VariantTypeFieldSpecListV4;
```

Named by [VariantTypeSpecV4](#).

VariantTypeFieldSpecListV5

SEQUENCE

```
sequence<VariantTypeFieldSpecV5> VariantTypeFieldSpecListV5;
```

Named by [VariantTypeSpecV5](#).

Optionals

MaybeException		75
MaybeF64	optional value for a f64	75
MaybeNetworkFootprintMulticast		75
MaybeNetworkFootprintPortValue	The value is not present when the port number is not known	75
MaybeSignal		75
MaybeStacktrace		75
MaybeTransportProtocol		75
MaybeU64	optional value for a u64	76

MaybeException

OPTIONAL

```
optional<UncaughtException> MaybeException;
```

MaybeF64

OPTIONAL

optional value for a f64

```
optional<f64> MaybeF64;
```

MaybeNetworkFootprintMulticast

OPTIONAL

```
optional<NetworkFootprintMulticast> MaybeNetworkFootprintMulticast;
```

MaybeNetworkFootprintPortValue

OPTIONAL

The value is not present when the port number is not known

```
optional<NetworkFootprintPortValue> MaybeNetworkFootprintPortValue;
```

MaybeSignal

OPTIONAL

```
optional<SignalData> MaybeSignal;
```

MaybeStacktrace

OPTIONAL

```
optional<StringList> MaybeStacktrace;
```

MaybeTransportProtocol

OPTIONAL

```
optional<u32> MaybeTransportProtocol;
```

MaybeU64 OPTIONAL

optional value for a u64

```
optional<u64> MaybeU64;
```

log

16 types.

Fixed records

BasicFile	Basic file sink	77
ColorStderr	Colored stderr sink	77
ColorStdout	Colored stdout sink	77
Config	Overall component configuration	77
LoggerConfig	Configuration used for a given spdlog logger.	77
Null	Single-threaded null sink	78
RotatingFile	Rotating file sink	78
SinkConfig	Configuration used for a given spdlog Sink.	78
Stderr	stderr sink	78
Stdout	stdout sink	78
Syslog	Syslog sink (only in linux)	78

BasicFile STRUCTURE

Basic file sink

Name	Type	Description
fileName	string	file to log to
truncate	bool	defaults to false
createParentDir	bool	true if we should create the parent dir first

Named by [SinkType](#).

ColorStderr STRUCTURE

Colored stderr sink

Named by [SinkType](#).

ColorStdout STRUCTURE

Colored stdout sink

Named by [SinkType](#).

Config STRUCTURE

Overall component configuration

Name	Type	Description
pattern	string	default format (empty means default)
level	LogLevel	default global level for all
sinks	SinkConfigList	sinks to configure
loggers	LoggerConfigList	loggers to configure
backtrace	bool	log backtrace is printed on failure

Named by [sen.kernel.KernelParams](#).

LoggerConfig STRUCTURE

Configuration used for a given spdlog logger.

Name	Type	Description
name	string	name of the logger
sinks	SinkNameList	name of the sinks to connect the logger to
level	LogLevel	default log level for this logger

Name	Type	Description
pattern	string	Formatting for the sinks in this logger. Does nothing if empty. Note that this replaces the format of the associated sinks, and this could affect other loggers using those sinks. The config is applied following the order of the Config.loggers field.

Named by [LoggerConfigList](#).

Null STRUCTURE

Single-threaded null sink

Named by [SinkType](#).

RotatingFile STRUCTURE

Rotating file sink

Name	Type	Description
baseFileName	string	
maxSize	u32	0 means use defaults
maxFiles	u32	0 means use defaults

Named by [SinkType](#).

SinkConfig STRUCTURE

Configuration used for a given spdlog Sink.

Name	Type	Description
name	string	name of the sink
singleThreaded	bool	if true, single-threaded (defaults to false)
pattern	string	the pattern used for the formatting (empty means default).
level	LogLevel	default log level for this sink
config	SinkType	the type-specific configuration parameters

Named by [SinkConfigList](#).

Stderr STRUCTURE

stderr sink

Named by [SinkType](#).

Stdout STRUCTURE

stdout sink

Named by [SinkType](#).

Syslog STRUCTURE

Syslog sink (only in linux)

Name	Type	Description
identity	string	
option	u32	
enableFormatting	bool	

Named by [SinkType](#).

Enumerations

[LogLevel](#)

Severity/importance of a log message.

79

LogLevel ENUMERATION

Severity/importance of a log message.

Held as u8.

warn	0
err	1
critical	2
trace	3
debug	4
info	5
off	6

Named by [sen.components.logmaster.LogMaster](#), [sen.components.logmaster.Logger](#), [Config](#), [LoggerConfig](#), [SinkConfig](#).

Variant records

[SinkType](#)

The type-specific configuration of the used spdlog Sink.

79

SinkType VARIANT

The type-specific configuration of the used spdlog Sink.

Type	Description
ColorStdout	
ColorStderr	
Stdout	
Stderr	
BasicFile	
RotatingFile	
Null	
Syslog	

Named by [SinkConfig](#).

Arrays

[LoggerConfigList](#)

79

[SinkConfigList](#)

79

[SinkNameList](#)

79

LoggerConfigList SEQUENCE

```
sequence<LoggerConfig> LoggerConfigList;
```

Named by [Config](#).

SinkConfigList SEQUENCE

```
sequence<SinkConfig> SinkConfigList;
```

Named by [Config](#).

SinkNameList SEQUENCE

```
sequence<string> SinkNameList;
```

Named by [LoggerConfig](#).

Built-in types

The types the language provides. Every one is written the same way on the wire wherever it appears.

Type	Meaning
<code>bool</code>	classic boolean
<code>f32</code>	32 bit floating point number
<code>f64</code>	64 bit floating point number
<code>i32</code>	32 bit signed integer
<code>sen.Duration</code>	a time duration
<code>sen.TimeStamp</code>	a point in time
<code>string</code>	unbounded string
<code>u16</code>	16 bit unsigned integer
<code>u32</code>	32 bit unsigned integer
<code>u64</code>	64 bit unsigned integer
<code>u8</code>	8 bit unsigned integer

Index

components.ether.BusConfig	7	components.logmaster.Config	26	kernel.AliasTypeSpecV4	48
components.ether.BusJoined	7	components.logmaster.LogMaster	26	kernel.AliasTypeSpecV5	48
components.ether.BusLeft	8	components.logmaster.Logger	26	kernel.ArgSpec	48
components.ether.ByteRange	8	components.py.Configuration	27	kernel.ArgSpecList	71
components.ether.Configuration	8	components.py.PythonInterpreter	27	kernel.ArgSpecListV4	71
components.ether.ControlMessage	11	components.py.StringList	27	kernel.ArgSpecListV5	71
components.ether.DiscoveryConfig	11	components.recorder.Configuration	28	kernel.ArgSpecV4	48
components.ether.DiscoveryHubAddress	8	components.recorder.Recorder	28	kernel.ArgSpecV5	49
components.ether.Endpoint	8	components.recorder.RecorderState	28	kernel.BasicType	64
components.ether.EnpointList	11	components.recorder.RecordingSettings	28	kernel.Buffer	71
components.ether.Ephemeral	8			kernel.BuildInfo	49
components.ether.Hello	9	components.recorder.RecordingSettings	28	kernel.BuiltComponentList	71
components.ether.MaybeDeviceName	12	List	29	kernel.BuiltComponentParams	49
components.ether.MaybeDiscoveryHub	12	components.recorder.SelectionList	29	kernel.BuiltInType	69
Port	12	components.replayer.Configuration	30	kernel.BusAddress	49
components.ether.MaybePortConfig	12	components.replayer.Replay	30	kernel.ClassTypeSpec	49
components.ether.MulticastAddress	12	components.replayer.ReplayStatus	31	kernel.ClassTypeSpecV4	50
Exclusions	12	components.replayer.Replayer	30	kernel.ClassTypeSpecV5	50
components.ether.MulticastAddress	12	components.rest.Argument	32	kernel.CompatibilityMode	64
Range	9	components.rest.Arguments	36	kernel.ComponentConfig	50
components.ether.MulticastDiscovery	9	components.rest.Bus	32	kernel.ComponentInfo	50
components.ether.MulticastRange	12	components.rest.BusNames	36	kernel.ComponentState	64
components.ether.PinnedPort	9	components.rest.BusSummary	32	kernel.CpuArch	65
components.ether.PortBinding	11	components.rest.Buses	36	kernel.CustomTypeData	69
components.ether.PortConfig	9	components.rest.ClientAuth	32	kernel.CustomTypeDataV4	69
components.ether.PortExclusions	12	components.rest.Configuration	33	kernel.CustomTypeDataV5	69
components.ether.PortRange	9	components.rest.Error	33	kernel.CustomTypeSpec	51
components.ether.ProbePortRange	10	components.rest.Events	36	kernel.CustomTypeSpecList	71
components.ether.ProtocolVersion	10	components.rest.HttpMethod	35	kernel.CustomTypeSpecListV4	71
components.ether.QueueConfig	10	components.rest.Interest	33	kernel.CustomTypeSpecListV5	71
components.ether.QueueEvictionPolicy	11	components.rest.InterestSummary	33	kernel.CustomTypeSpecV4	51
components.ether.Ready	10	components.rest.InterestsSummary	36	kernel.CustomTypeSpecV5	51
components.ether.SessionPresenceBeam	10	components.rest.InvokeStatus	35	kernel.EnumTypeSpec	51
components.ether.TcpDiscovery	10	components.rest.Link	33	kernel.EnumTypeSpecV4	51
components.explorer.Configuration	13	components.rest.Links	36	kernel.EnumTypeSpecV5	52
components.influx.Configuration	14	components.rest.Notification	33	kernel.EnumeratorSpec	52
components.influx.Protocol	14	components.rest.NotificationType	35	kernel.EnumeratorSpecList	72
components.influx.SelectionList	14	components.rest.Object	33	kernel.EnumeratorSpecListV4	72
components.jsonrpc.AddedObjectEntry	19	components.rest.ObjectEvent	34	kernel.EnumeratorSpecListV5	72
components.jsonrpc.AddedObjectList	23	components.rest.ObjectMethod	34	kernel.EnumeratorSpecV4	52
components.jsonrpc.Configuration	20	components.rest.ObjectSummary	34	kernel.EnumeratorSpecV5	52
components.jsonrpc.ConnectionLimits	20	components.rest.ObjectsSummary	36	kernel.EnvVar	52
components.jsonrpc.Identity	20	components.rest.Properties	37	kernel.EnvironmentVarList	72
components.jsonrpc.JsonRpcServer	15	components.rest.RelType	35	kernel.ErrorCategory	65
components.jsonrpc.MaybeConnection	15	components.rest.Session	34	kernel.ErrorData	52
Limits	24	components.rest.SessionSummary	34	kernel.ErrorReport	53
components.jsonrpc.MaybeControlBus	24	components.rest.Sessions	37	kernel.EventSpec	53
Name	24	components.rest.Status	36	kernel.EventSpecList	72
components.jsonrpc.MaybeFlag	24	components.rest.SubscriptionOptions	34	kernel.EventSpecListV4	72
components.jsonrpc.MaybeHigh	24	components.rest.Subscriptions	34	kernel.EventSpecListV5	72
BackpressureBytes	24	components.rest.Success	35	kernel.EventSpecV4	53
components.jsonrpc.MaybeIdleTimeout	24	components.rest.ThreadPoolSize	37	kernel.EventSpecV5	53
Seconds	25	components.rest.UpdateFrequency	37	kernel.ExceptionKind	65
components.jsonrpc.MaybeMax	25	components.rest.UpdateTime	37	kernel.ExecError	53
BackpressureBytes	25	components.rest.Version	35	kernel.GitStatus	65
components.jsonrpc.MaybeMaxPayload	25	components.shell.BufferStyle	41	kernel.IntegralType	65
Bytes	25	components.shell.CPrint	38	kernel.KernelApi	46
components.jsonrpc.MaybeMember	25	components.shell.ClearCurrentLine	39	kernel.KernelParams	53
Selector	25	components.shell.ClearRemainingCurrent	39	kernel.LoadedComponentList	72
components.jsonrpc.MaybeRateHz	25	Line	39	kernel.LoadedComponentParams	54
components.jsonrpc.MaybeStringList	25	components.shell.ClearScreen	39	kernel.MaybeException	75
components.jsonrpc.MaybeSubscribe	25	components.shell.Configuration	39	kernel.MaybeF64	75
Block	25	components.shell.HideCursor	39	kernel.MaybeNetworkFootprintMulticast	75
components.jsonrpc.MaybeTlsConfig	25	components.shell.ListStyle	41	kernel.MaybeNetworkFootprintPortValue	75
components.jsonrpc.MaybeUpdateFreqHz	25	components.shell.MoveCursorAllLeft	39		
25		components.shell.MoveCursorDown	39	kernel.MaybeSignal	75
components.jsonrpc.MemberSelector	23	components.shell.MoveCursorLeft	39	kernel.MaybeStacktrace	75
components.jsonrpc.NamedSelection	20	components.shell.MoveCursorRight	39	kernel.MaybeTransportProtocol	75
components.jsonrpc.ObjectInfo	20	components.shell.MoveCursorUp	40	kernel.MaybeU64	76
components.jsonrpc.ObjectInfos	23	components.shell.NewLine	40	kernel.MethodConstnessSpec	66
components.jsonrpc.ObjectState	21	components.shell.Print	40	kernel.MethodSpec	54
components.jsonrpc.ObjectStateList	23	components.shell.Query	40	kernel.MethodSpecList	72
components.jsonrpc.PropertyError	21	components.shell.QueryList	42	kernel.MethodSpecListV4	73
components.jsonrpc.PropertyErrorList	23	components.shell.RestoreCursorPosition	40	kernel.MethodSpecListV5	73
components.jsonrpc.PropertyValueList	24	40		kernel.MethodSpecV4	54
components.jsonrpc.PropertyValuePair	21	components.shell.SaveCursorPosition	40	kernel.MethodSpecV5	55
components.jsonrpc.RegisteredBundle	21	components.shell.SetBgColor	40	kernel.NetworkFootprint	55
Info	21	components.shell.SetFgColor	40	kernel.NetworkFootprintAddressRange	55
components.jsonrpc.RegisteredBundle	21	components.shell.SetWindowTitle	40	List	73
InfoList	24	components.shell.Shell	38	kernel.NetworkFootprintBus	55
components.jsonrpc.SessionInfo	21	components.shell.ShowCursor	41	kernel.NetworkFootprintBusIdentity	56
components.jsonrpc.SessionInfoList	24	components.shell.SourcesList	42	kernel.NetworkFootprintBusList	73
components.jsonrpc.StaticBundle	22	components.shell.TerminalCmd	41	kernel.NetworkFootprintBusSource	66
components.jsonrpc.StaticFile	22	components.shell.TimeStyle	41	kernel.NetworkFootprintCollision	56
components.jsonrpc.StaticFileList	24	components.term.Configuration	43	kernel.NetworkFootprintMulticast	56
components.jsonrpc.StaticFileServer	18	components.term.NameList	44	kernel.NetworkFootprintPort	56
components.jsonrpc.SubscribeBlock	22	components.term.QueryDef	43	kernel.NetworkFootprintPortExclusions	57
components.jsonrpc.TlsConfig	22	components.term.QueryList	44	kernel.NetworkFootprintPortKind	66
components.jsonrpc.TypeLookupResult	22	components.term.SourcesList	44	kernel.NetworkFootprintPortList	73
components.jsonrpc.WildcardSelection	23	components.term.Term	43	kernel.NetworkFootprintPortMode	66
		components.term.ThemeStyle	43	kernel.NetworkFootprintPortRange	57
		components.term.TimeStyle	44	kernel.NetworkFootprintPortRangeList	73
		db.OutSettings	45	kernel.NetworkFootprintPortValue	69
		db.OutStats	45	kernel.NetworkFootprintSelfCollision	57
		db.Summary	45		
		kernel.AliasTypeSpec	48		

[kernel.NetworkFootprintSelfCollisionList](#)

73	
kernel.NumericType	70
kernel.OpFinished	57
kernel.OpNotFinished	57
kernel.OpState	70
kernel.OptionalTypeSpec	57
kernel.OptionalTypeSpecV4	58
kernel.OptionalTypeSpecV5	58
kernel.OsKind	66
kernel.PrecisionSleep	58
kernel.Priority	67
kernel.ProcessData	58
kernel.ProcessInfo	58
kernel.Processes	73
kernel.PropertyCategorySpec	67
kernel.PropertyRelationSpec	67
kernel.PropertySpec	58
kernel.PropertySpecList	73
kernel.PropertySpecListV4	74
kernel.PropertySpecListV5	74
kernel.PropertySpecV4	59
kernel.PropertySpecV5	59
kernel.QuantityTypeSpec	59
kernel.QuantityTypeSpecV4	59
kernel.QuantityTypeSpecV5	60
kernel.QueueConfig	60
kernel.QueueEvictionPolicy	67
kernel.RealType	67
kernel.RunMode	67
kernel.SenData	60
kernel.SequenceTypeSpec	60
kernel.SequenceTypeSpecV4	61
kernel.SequenceTypeSpecV5	61
kernel.SignalData	61
kernel.SleepPolicy	70
kernel.StringList	74
kernel.StructTypeFieldSpec	61
kernel.StructTypeFieldSpecList	74
kernel.StructTypeFieldSpecListV4	74
kernel.StructTypeFieldSpecListV5	74
kernel.StructTypeFieldSpecV4	61
kernel.StructTypeFieldSpecV5	61
kernel.StructTypeSpec	62
kernel.StructTypeSpecV4	62
kernel.StructTypeSpecV5	62
kernel.SystemSleep	62
kernel.ThreadCreateErr	68
kernel.TransportModeSpec	68
kernel.U32List	74
kernel.UncaughtException	62
kernel.UnitCat	68
kernel.UnitInfo	62
kernel.UnitList	74
kernel.VariantTypeFieldSpec	63
kernel.VariantTypeFieldSpecList	74
kernel.VariantTypeFieldSpecListV4	75
kernel.VariantTypeFieldSpecListV5	75
kernel.VariantTypeFieldSpecV4	63
kernel.VariantTypeFieldSpecV5	63
kernel.VariantTypeSpec	63
kernel.VariantTypeSpecV4	63
kernel.VariantTypeSpecV5	63
kernel.VirtualClock	46
kernel.VirtualKernelClock	46
kernel.VirtualMasterClock	46
kernel.WordSize	68
kernel.log.BasicFile	77
kernel.log.ColorStderr	77
kernel.log.ColorStdout	77
kernel.log.Config	77
kernel.log.LogLevel	79
kernel.log.LoggerConfig	77
kernel.log.LoggerConfigList	79
kernel.log.Null	78
kernel.log.RotatingFile	78
kernel.log.SinkConfig	78
kernel.log.SinkConfigList	79
kernel.log.SinkNameList	79
kernel.log.SinkType	79
kernel.log.Stderr	78
kernel.log.Stdout	78
kernel.log.Syslog	78