Brainsight Network Server Protocol

v1.0.1

© Rogue Research Inc.

2025-02-21

Contents

Overview

This network protocol allows you to instruct Brainsight to perform various tasks and allows Brainsight to inform you when various events have occurred.

You can instruct Brainsight to:

- list all open documents
- list all open sessions
- list all targets in the active session
- create a target in the active session
- create a sample in the active session
- change the target selection in the active session

And Brainsight can inform you when:

- the active session's crosshairs coordinate changes
- the active session's selected target changes
- a sample is created in the active session
- EMG data is received for a newly created sample
- the Polaris camera updates with new tool positions
- an IOBox TTL trigger occurs

The former group are explicit commands that you give to Brainsight, while the latter group Brainsight sends to you as soon as they occur (you must opt-in to receive these).

Example code

We provide a Python library to handle the low-level networking details, and a sample Python application illustrating its use. Alternatively, you can use the information in this document to implement the network protocol yourself in a different programming language.

Network Connection Overview

- All connections are with TCP over either IPv4 or IPv6 on port 60000.

- Brainsight acts as the server, your code is the client.
- Multiple clients can be connected simultaneously.
- Each client connects with a single socket.
- Brainsight advertises its availability using Bonjour (also known as mDNS).
- Clients can be running any operating system and programmed in any language.
- The network protocol is based on JSON, encoded as UTF-8.
- Each JSON packet is ended with the RS (record separator) character (ASCII 30 decimal, 0x1E hexadecimal).

Packet types

1. `request`: Corresponds to a packet sent by the client to the server.
2. `response`: Corresponds to a packet sent by the server to the client pertaining to a previous `request` packet.
3. `stream`: Corresponds to live feedback sent from the server (not relating to any particular request) as a result of the user's interaction with Brainsight, example: when the crosshairs are moved the updated position is streamed to the client as feedback `stream:session-crosshairs-moved`.
4. `error`: Corresponds to a packet sent by the server to the client in response to a packet that was missing vital information or completely malformed.

Common for client to server packets

The following are some common keys that may/must exist in packets sent from a client to the server:

- `packet-name` (required, string) : corresponds to the request to be executed. This defines what other keys to expect in the dictionary. Must have the prefix `request:`.
- `packet-uuid` (required, string) : An identifier that will be returned back in response packets in the `response-to-uuid` field. This is highly suggested to be unique, like a UUID, so that it can be used for bookkeeping.

Common for server to client packets

The following are some common keys that may/must exist in packets sent from the server to a client:

- `packet-name` (required, string) : This defines what other keys to expect in the dictionary. Prefixed with `response:`, `response-delayed:`, or `stream:` for further elaboration of the nature of the packet.
- `packet-uuid` (required, string) : A UUID in string representation.
- `error-code` (required, integer) : Zero if the command being replied about was successful, non-zero if some error occurred. The non-zero value indi-

cates what kind of error occurred. This field is suitable for parsing in code. See Appendix I for possible error codes.

- `error-message` (optional, string) : Not present if the command being replied about was successful, otherwise a human-readable English error message describing the reason for failure. This field is not suitable for parsing in code, it is there as an aid to a human reader.
- `response-to-uuid` (optional, string) : When a packet sent to Brainsight contained a `packet-uuid`, the response to that packet will have the same string passed back in this field.
- `timestamp` (required, string): A date and timestamp, intended for bookkeeping purposes. See Appendix II for format details.

Protocol commands

Fetch Protocol Version

This command retrieves the current protocol version number. When backwards-incompatible changes are inevitably made, the major version number will be incremented. The minor and patch version numbers will be incremented when smaller backwards-compatible changes are made. See Appendix VI for the history of changes to this network protocol. This document describes version 1.0.1.

Request format:

- `packet-name` (required, string) : The string `request:get-protocol-version`.

Response format:

- `packet-name` (required, string) : The string `response:get-protocol-version`.
- `response-data` (required, dictionary) : Dictionary with the following keys:
  - `major-version` (required, integer) : The current protocol major version.
  - `minor-version` (required, integer) : The current protocol minor version.
  - `patch-version` (required, integer) : The current protocol patch version.

Example request:

```
{
    "packet-name": "request:get-protocol-version",

    "packet-uuid": "2F2DA0F9-785B-41EA-B197-B3AF8AECDA55"
}
```

Example response:

```
{
    "packet-name": "response:get-protocol-version",
```

```
    "response-data": {
        "major-version": 1,
        "minor-version": 0,
        "patch-version": 1
    },

    "error-code": 0,
    "response-to-uuid": "2F2DA0F9-785B-41EA-B197-B3AF8AECDA55",
    "packet-uuid": "04C91186-0B70-4289-8768-789F62BFA55F",
    "timestamp": "2024-07-26T15:50:59.123Z"
}
```

Listing of project documents

This command lists all currently open project documents.

Request format:

- `packet-name` (required, string) : The string `request:list-documents`.

Response format:

- `packet-name` (required, string) : The string `response:list-documents`.
- `response-data` (required, array of dictionaries) : Array of dictionaries with the following keys:
  - `file-name` (required, string) : The name of the document, including the `.bsproj` file extension.
  - `file-path` (optional, string) : The full path to the document on disk. If the document has never been saved, this will not be present.

Example request:

```
{
    "packet-name": "request:list-documents",

    "packet-uuid": "FCB16A8E-F170-4383-A74B-75A9245F3689"
}
```

Example response:

```
{
    "packet-name": "response:list-documents",

    "response-data": [
        {
            "file-name": "Experimental.bsproj",
            "file-path": "/Users/brainsight/Documents/Experimental.bsproj"
        }
    ],
```

```
    "error-code": 0,
    "response-to-uuid": "FCB16A8E-F170-4383-A74B-75A9245F3689",
    "packet-uuid": "F3F3D1FF-71D7-4640-AF23-BC385E134333",
    "timestamp": "2024-07-26T15:55:09.456Z"
}
```

Listing of sessions

This command lists all sessions present in the opened document.

Request format:

- `packet-name` (required, string) : The string `request:list-sessions`.

Response format:

- `packet-name` (required, string) : The string `response:list-sessions`.
- `response-data` (required, array of dictionaries) : Array of dictionaries with the following keys:
  - `uuid` (required, string) : The session's uuid, which uniquely identifies it amongst all Brainsight objects.
  - `name` (required, string) : The session's name.

The action may fail if:

- No document is opened. `kBSErrorCode_NoDocuments`.
- Multiple documents are opened. `kBSErrorCode_MoreThanOneDocument`.

The action will succeed only if exactly one document is open. If that one document has no sessions, the result will be an empty array and an `error-code` of 0.

Example request:

```
{
    "packet-name": "request:list-sessions",

    "packet-uuid": "2038BB78-A79A-4929-B262-C67877100753"
}
```

Example response:

```
{
    "packet-name": "response:list-sessions",

    "response-data": [
        {
            "name": "Session 1",
            "uuid": "A2ECC95A-66E7-44F7-B8A6-A770396903AC"
```

```
        }
    ],

    "error-code": 0,
    "response-to-uuid": "2038BB78-A79A-4929-B262-C67877100753",
    "packet-uuid": "67E37AB3-3FD2-42C2-AD05-5EA055B987A1",
    "timestamp": "2024-07-26T16:18:03.789Z"
}
```

Listing of targets in the active session

This command lists all targets present in the active session. The list is sorted in the same order as in the session.

Request format:

- `packet-name` (required, string) : The string `request:list-session-targets`.
- `session-name` (optional, string) : The name of the session of interest, targets pertaining to it will be listed if the session is active. Generally not necessary because the currently active session is presumed. Only in the case of two simultaneous sessions (for tracking two TMS coils) must the session name be provided.

Response format:

- `packet-name` (required, string) : The string `response:list-session-targets`.
- `response-data` (required, array of dictionaries) : Array of dictionaries with the following keys:
  - `name` (required, string) : The name of the target.
  - `position` (optional, array of 16 floats) : The position of the target as a 4x4 matrix. See Appendix III for format details. For folders, this will not be present.
  - `coordinate-system` (optional, string) : The coordinate system of the position given. The coordinate system of the session's Perform step will be used. 'See Appendix III for details. For folders, this will not be present.
  - `index-path` (required, 0-based array of integers) : An array of 0-based integers that represent the path to the target in the list of targets of a session. For example, the array [0, 2] indicates that the target is the 3rd item `[2]` of the 1st folder `[0]`.
  - `uuid` (required, string) : The target's uuid, which uniquely identifies it amongst all Brainsight objects.

The action may fail if:

- No document is opened. `kBSErrorCode_NoDocuments`.
- No session with the given `session-name` is active. `kBSErrorCode_NoActiveSessionWithName`.

- The session window is open, but the perform step has not been activated. `kBSErrorCode_PerformStepNotLoaded`.
- No session is active. `kBSErrorCode_NoActiveSession`.

Example request:


```
{
    "packet-name": "request:list-session-targets",
    "session-name": "Session 1",
    "packet-uuid": "02EDD26B-95D1-4319-BC1E-2A978F2215E4"
}
```


Example response:


```
{
    "packet-name": "response:list-session-targets",

    "response-data": [
        {
            "index-path": [ 0, 2 ],
            "name": "Marker 1",
            "coordinate-system": "World",
            "position": [
                -0.6, -0.5, -0.4,  7.8,
                 0.6, -0.7, -0.1, 18.5,
                -0.2, -0.4,  0.8, 10.8,
                   0,    0,    0,    1
            ],
            "uuid": "7856B7EA-36F2-4CC9-9C41-68E92F077BD6"
        },
        {
            "index-path": [ 1 ],
            "name": "Marker 2",
            "coordinate-system": "World",
            "position": [
                -0.7, -0.6, -0.9,  19.3,
                 0.5, -0.6, -0.5,  82.9,
                 0.2, -0.4,  0.8, 141.0,
                   0,    0,    0,     1
            ],
            "uuid": "801CC810-C459-42F6-9D40-3F9DBB9E7D29"
        }
    ],

    "error-code": 0,
    "response-to-uuid": "02EDD26B-95D1-4319-BC1E-2A978F2215E4",
    "packet-uuid": "A7F110F2-6399-4443-AE50-4716200CCF52",
    "timestamp": "2024-07-26T16:32:41.348Z"
```

```
}
```

Creating a target in the active session

This command creates a target and adds it to the active session's list of targets (at the end of the list). The target position can optionally be specified, otherwise the current crosshairs position in the session's Perform window is used.

Request format:

- `packet-name` (required, string) : The string `request:create-target-at-location`.
- `name` (optional, string) : The name for the newly created target. Defaults to using Brainsight's naming convention if not provided (ex: `Marker 1` or `Trajectory 1`, etc.)
- `position` (optional, array of 16 floats) : The position and orientation of the target as a 4x4 matrix. See Appendix III for format details. If specified, `coordinate-system` must also be provided. If unspecified, defaults to placing the target at the crosshairs position in the Session Perform window.
- `coordinate-system` (optional, string) : The coordinate system of the position given. See Appendix III for details. Required if `position` is also specified.
- `session-name` (optional, string) : The name of the session to add the target to. Generally not necessary because the currently active session is presumed. Only in the case of two simultaneous sessions (for tracking two TMS coils) must the session name be provided.

Response format:

- `packet-name` (required, string) : The string `response:create-target-at-location`.
- `response-data` (required, dictionary) : Dictionary of the created target's attributes as its keys. Note, the target will be added to the session.
  - `index-path` (required, 0-based array of integers) : An array of 0-based integers that represent the path to the target in the list of targets of a session. For example, the array [0, 2] indicates that the target is the 3rd item `[2]` of the 1st folder `[0]`.
  - `uuid` (required, string) : The target's uuid, which uniquely identifies it amongst all Brainsight objects.
  - `name` (required, string) : The name of the target.
  - `position` (required, array of 16 floats) : The position and orientation of the target as a 4x4 matrix. See Appendix III for format details.
  - `coordinate-system` (required, string) : The coordinate system of the `position` returned. The coordinate system of the session's Perform step will be used. See Appendix III for details.

Prerequisites:

- Exactly one session must be opened (or in the case of two simultaneous sessions, for tracking two TMS coils, the session name must be provided).

The action may fail if:

- No document is opened. `kBSErrorCode_NoDocuments`.
- No session with the given `session-name` is active. `kBSErrorCode_NoActiveSessionWithName`.
- No session is active. `kBSErrorCode_NoActiveSession`.
- The session window is open, but the perform step has not been activated. `kBSErrorCode_PerformStepNotLoaded`.
- The `position` has less or more than exactly 16 floats. `kBSErrorCode_MatrixSizeNot4x4`
- The `position` has non-finite or crazy values. `kBSErrorCode_CrazyFloatingPoint`
- The `position` is not an invertible matrix. `kBSErrorCode_NonInvertibleMatrix`
- The `position` is not a sufficiently rigid matrix (too much scaling/shearing). `kBSErrorCode_NonRigidMatrix`
- The `coordinate-system` string does not match any coordinate system in the document. `kBSErrorCode_CoordinateSystemUnknown`

Example request:

```
{
    "packet-name": "request:create-target-at-location",

    "session-name": "Session 1",
    "name": "My new target",
    "position": [
         -0.1, -0.6, -0.8, 12.8,
          0.6, -0.7,  0.0, 54.2,
        -0.12,  0.2,  0.9, 62.9,
            0,    0,    0,    1
    ],
    "coordinate-system": "World",

    "packet-uuid": "E17E37B2-499D-4C76-AFFF-83E7FB6023EC"
}
```

Example response:

```
{
    "packet-name": "response:create-target-at-location",

    "response-data": {
        "name": "My new target",
        "index-path": [ 3 ],
        "position": [
             -0.1, -0.6, -0.8, 12.8,
              0.6, -0.7,  0.0, 54.2,
            -0.12,  0.2,  0.9, 62.9,
                0,    0,    0,    1
        ],
        "coordinate-system": "World",
```

```
        "uuid": "7856B7EA-36F2-4CC9-9C41-68E92F077BD6"
    },

    "error-code": 0,
    "response-to-uuid": "E17E37B2-499D-4C76-AFFF-83E7FB6023EC",
    "packet-uuid": "7C4CF4A8-E6C0-4A1B-B540-AAD3A6D1DF96"
}
```

Creating a sample in the active session

This command creates a new sample in the currently active session at the crosshairs location in the Perform step. The session window must already be opened.

Request format:

- `packet-name` (required, string) : The string `request:create-sample`.
- `session-name` (optional, string) : The name of the session to create the sample in. Generally not necessary because the currently active session is presumed. Only in the case of two simultaneous sessions (for tracking two TMS coils) must the session name be provided.
- `name` (optional, string) : The name for the newly created Sample. Defaults to using Brainsight's naming convention if not provided (ex: `Sample 1`, `Sample 2`, etc.)

Response format:

- `packet-name` (required, string) : The string `response:create-sample`.
- `response-data` (required, dictionary) : Dictionary of the created sample's attributes as its keys. Note, the sample will be added to the session.
  - `name` (required, string) : The name of the newly created sample. This will either be the name provided given back, or the automatically assigned name if none was given.
  - `uuid` (required, string) : The uuid of the newly created sample, which uniquely identifies it amongst all Brainsight objects.
  - `position` (required, array of 16 floats) : The position and orientation of the newly created Sample expressed as a 4x4 matrix.
  - `coordinate-system` (required, string) : The coordinate system of the `position` given. The coordinate system of the session's Perform step will be used. See Appendix III for details.

The action may fail if:

- No document is opened. `kBSErrorCode_NoDocuments`.
- No session with the given `session-name` is active. `kBSErrorCode_NoActiveSessionWithName`.
- No session is active. `kBSErrorCode_NoActiveSession`.
- The session window is open, but the perform step has not been activated. `kBSErrorCode_PerformStepNotLoaded`.

- The sample could not be created for the same kinds of reasons that clicking "Sample Now" in Brainsight might fail (mainly because the tracked tool is not visible). `kBSErrorCode_GeneralSampleCreationFailure`.

Example request:

```
{
    "packet-name": "request:create-sample",

    "session-name": "Session 2",
    "name": "My new sample",

    "packet-uuid": "16AA49DB-13DC-4EDF-9BA2-47A83D845705"
}
```

Example response:

```
{
    "packet-name": "response:create-sample",

    "response-data": {
        "name": "My new sample",
        "position": [
            -0.7, -0.6, -0.08, 12.8,
             0.6, -0.7,   2.9, 15.2,
            -0.1,  0.2,   0.9, 62.7,
               0,    0,     0,    1
        ],
        "coordinate-system": "World",
        "uuid": "0056C7EA-25C2-4XY9-9C41-67E92F034BE9"
    },

    "error-code": 0,
    "response-to-uuid": "16AA49DB-13DC-4EDF-9BA2-47A83D845705",
    "packet-uuid": "C08BCCDC-626A-4F50-BB94-9EF4D6382271"
}
```

Changing the target selection in the active session

This command changes the selected target to the specified one (in the active session).

Request format:

- `packet-name` (required, string) : The string `request:select-target-in-session`.
- `session-name` (optional, string) : The name of the session in which to change the selected target. Generally not necessary because the currently

active session is presumed. Only in the case of two simultaneous sessions (for tracking two TMS coils) must the session name be provided.
- `index-path` (optional, 0-based array of integers) : A 0-based array of integers that represent the path to the target in the list of targets of a session. For example, the array [0, 2, 1] indicates that the target is the 2nd item `[1]` of the 3rd sub-folder `[2]` of 1st folder `[0]`. Note, either `index-path` or `name` must be specified, but not both. Preferred because it's unique and avoids cases where targets may have the same names.
- `name` (optional, string) : The name of a target in the list of targets in a session. One of `index-path` or `name` must be specified, but not both. If the target appears in the list multiple times, the first instance is selected.

Response format:

- `packet-name` (required, string) : The string `response:select-target-in-session`.
- `response-data` (required, dictionary) : Dictionary of the selected target & its attributes where the attributes represent the keys to the dictionary pertaining to the session name provided and feedback if action was successful or not.
  - `index-path` (required, 0-based array of integers) : The index-path of the newly selected target in the session's list of targets.
  - `name` (required, string) : The name of the newly selected target.
  - `uuid` (required, string) : The uuid of the newly selected target, which uniquely identifies it amongst all Brainsight objects.
  - `position` (optional, array of 16 floats) : The position and orientation of the newly selected target expressed as a 4x4 matrix. Almost always present, except in the rare case of the session's target not matching the name of an actual target (i.e. the target is missing).
  - `coordinate-system` (optional, string) : The coordinate system of the position given. The coordinate system of the session's Perform step will be used. See Appendix III for details.

The action may fail if:

- No document is opened. `kBSErrorCode_NoDocuments`.
- No session with the given `session-name` is active. `kBSErrorCode_NoActiveSessionWithName`.
- No session is active. `kBSErrorCode_NoActiveSession`.
- The session window is open, but the perform step has not been activated. `kBSErrorCode_PerformStepNotLoaded`.
- There's no target with the given name. `kBSErrorCode_NoTargetWithName`.
- There's no target with the given index path. `kBSErrorCode_NoTargetWithIndexPath`.

Example request:

```
{
    "packet-name": "request:select-target-in-session",

    "session-name": "Session 1",
    "index-path": [ 0, 3 ],
```

```
    "packet-uuid": "FD1249F8-C1BE-40DC-A421-61D03C403761"
}
```

Example response:

```
{
    "packet-name": "response:select-target-in-session",

    "response-data": {
        "name": "Target 2",
        "position": [
            1, 0, 0, 0,
            0, 1, 0, 0,
            0, 0, 1, 0,
            0, 0, 0, 1
        ],
        "coordinate-system": "World",
        "index-path": [ 0, 3 ],
        "uuid": "307FE810-C459-42F6-9D40-3F9DBC9E7D28"
    },

    "error-code": 0,
    "response-to-uuid": "FD1249F8-C1BE-40DC-A421-61D03C403761",
    "packet-uuid": "4C21D07F-12B8-4F88-A746-14C9DDECA70F"
}
```

Setting stream options

This command allows turning on or off the various streaming options. By default, no streams are sent, you must turn on any you are interested in.

Request format:

- `packet-name` (required, string) : The string `request:set-stream-option`.
- `stream-name` (required, string) : The name of the stream option to change. Must be one of:
  - `stream:session-crosshairs-moved`
  - `stream:target-selected`
  - `stream:sample-creation`
  - `stream:sample-emg`
  - `stream:session-polaris-update`
  - `stream:session-ttl-triggers`
- `stream-value` (required, boolean) : Whether the named stream option should be enabled or not.

Response format:

- `packet-name` (required, string) : The string `response:set-stream-option`.

Example request:

```
{
    "packet-name": "request:set-stream-option",
    "packet-uuid": "72EAD6AB-4E5F-4DE5-9348-7B7C3F1EDF2E",

    "stream-name": "stream:session-polaris-update",
    "stream-value": true,
}
```

Example response:

```
{
    "packet-name": "response:set-stream-option",

    "error-code": 0,
    "response-to-uuid": "72EAD6AB-4E5F-4DE5-9348-7B7C3F1EDF2E",
    "packet-uuid": "31EDCFBB-21AD-4A53-8E8F-85FBF044D2EF"
}
```

Protocol Streams

Active session crosshairs coordinate changed

This packet is streamed over to clients whenever the crosshairs changes its location (in the active Session's Perform step). Note, this takes places only during an active session.

Stream format:

- `packet-name` (required, string) : The string `stream:session-crosshairs-moved`.
- `timestamp` (required, string) : The date/time that the crosshairs moved. See Appendix II for format details.
- `crosshairs-mode` (required, string) : The mode of the crosshairs, example: `Mouse`, `Pointer`, or a name of a coil calibration. See Appendix IV for details.
- `position` (required, array of 16 floats) : The position and orientation of the crosshairs. See Appendix III for details.
- `coordinate-system` (required, string) : the coordinate system of the `position`. The coordinate system of the session's Perform step will be used. See Appendix III for details.

Example stream:

```
{
    "packet-name": "stream:session-crosshairs-moved",
    "packet-uuid": "801CC810-FD7F-42F6-9E04-3F9DBBD63E69",

    "timestamp": "2024-07-26T16:32:41.348Z",
    "coordinate-system": "World",
    "crosshairs-mode": "Mouse",
    "position": [
        4, 0,  1,  0.0,
        0, 2,  0, -4.2,
        0, 0,  3,  7.5,
        0, 0, -8,  0.1
    ]
```

```
}
```

Active session selected target changed

This packet is streamed over to clients whenever the selected target in the active Session's Perform step changes.

Stream format:

- `packet-name` (required, string) : The string `stream:target-selected`.
- `timestamp` (required, string) : The date/time that the target selection changed. See Appendix II for format details.
- `name` (required, string) : If one target is now selected, the target's name. If no targets are now selected, the special string `<No Selection>`.
- `index-path` (optional, 0-based array of integers) : The index-path of the newly selected target in the session's list of targets. For the case of no selection or multiple selection, this key will not be present.
- `uuid` (optional, string) : If one target is now selected, the target's uuid. Otherwise, not present.
- `position` (optional, array of 16 floats) : If exactly one target is now selected, the target's location and orientation (as a 4x4 matrix). Usually present, but not present when: zero targets are now selected, a folder is selected, the target is missing (not matching the name of an actual target).
- `coordinate-system` (optional, string) : The coordinate system of the `position`. The coordinate system of the session's Perform step will be used. Present when `position` is. See Appendix III for details.

Example stream:


```
{
    "packet-name": "stream:target-selected",
    "packet-uuid": "A9B03439-FD7F-4FCE-9E04-8AE7CAD63E69",

    "timestamp": "2024-07-26T15:50:59.123Z",
    "name": "Marker 2",
    "index-path": [ 0, 3 ],
    "position": [
         0.78,  0.67, 0.09, 8.069,
        -0.55,  0.62, 0.55,  59.3,
         0.27, -0.48, 0.82, 61.88,
            0,     0,    0,     1
    ],
    "coordinate-system": "NIfTI:S:Scanner",
    "uuid": "801CC810-C459-42F6-9D40-3F9DBB9E7D29"
}
```

Sample created in the active session

This packet is streamed over to clients whenever a sample is created during a session.

Stream format:

- `packet-name` (required, string) : The string `stream:sample-creation`.
- The created sample's attributes:
  - `name` (required, string) : The sample's name.
  - `uuid` (required, string) : The sample's uuid.
  - `index` (required, integer) : The 0-based index of the newly created sample in the session's list of samples.
  - `position` (required, array of 16 floats) : The sample's location and orientation (as a 4x4 matrix).
  - `target-position` (optional, array of 16 floats) : The location and orientation (as a 4x4 matrix) of the target that was being targeted when the sample was created. If no target was selected, this is not present.
  - `target-name` (optional, string) : The name of the target that was being targeted when the sample was created. If no target was selected, this is not present.
  - `coordinate-system` (required, string) : The coordinate system of both the `targetPosition` and the `position`. The coordinate system of the session's Perform step will be used. See Appendix III for details.
  - `creation-cause` (required, integer) : An integer indicating the cause of the sample's creation. See Appendix V for details.
  - `creation-date` (required, string) : The timestamp at the time of the sample's creation. See Appendix II for format details.
  - `crosshairs-mode` (required, string) : The mode of the crosshairs when the sample was created, ex: `Mouse`, `Pointer`, or a coil calibration name. See Appendix IV for details.
  - `crosshairs-offset` (required, float) : The offset of the crosshairs (in mm) when the sample was created.
  - `crosshairs-twist` (required, float) : The crosshairs's twist (in radians) when the sample was created. (The user interface for this is only available in the Vet version of Brainsight, and is labeled "pointer twist".)

Example stream:


```
{
    "packet-name": "stream:sample-creation",
    "packet-uuid": "D5AA04CE-861C-4463-866C-7D9EFB0EB375",

    "name": "Sample 1",
    "index": 1,
    "position": [
```

```
        -1,  0, 0,     0,
         0, -1, 0, 18.25,
         0,  0, 1, 16.86,
         0,  0, 0,     1
    ],
    "creation-cause": 6,
    "creation-date": "2024-02-05T18:32:37.846Z",
    "crosshairs-mode": "Pointer",
    "crosshairs-offset": 0,
    "crosshairs-twist": 0,
    "coordinate-system": "NIfTI:S:Scanner",
    "target-name": "Target 2",
    "target-position": [
        -0.69, -0.53, -0.47,   97.8,
         0.67, -0.71, -0.18, 118.55,
        -0.24, -0.45,  0.85, 150.84,
            0,     0,     0,      1
    ]
    "uuid": "597EAD78-1E26-4EA9-949C-3708484F8427",
}
```


EMG data received for a newly created sample

This packet is sent whenever EMG data is added to a sample, either from a Brainsight EMG Pod, or a NeuroConn NEURO PRAX device. EMG data can arrive substantially after the creation of the sample by a TMS pulse (up to 1.2 seconds with a Brainsight EMG Pod), and so this is a separate packet from `stream:sample-creation`. In order to know about the Sample's initial creation immediately, `stream:sample-creation` is sent first and `stream:sample-emg` comes a bit later. The information in the former is repeated in the latter, so you might choose to ignore the former depending on your needs, otherwise the sample's `uuid` should be used for bookkeeping to coordinate between the two packets.

Stream format:

- `packet-name` (required, string) : The string `stream:sample-emg`.
- The sample's attributes:
  - `name` (required, string) : The sample's name.
  - `index` (required, integer) : The 0-based index of the sample in the session's list of samples.
  - `uuid` (required, string) : The sample's uuid. This matches a `uuid` from a previously received `stream:sample-creation` packet and should be used for any bookkeeping.
  - `min-emg-time-range-ms` (required, float) : The starting time, relative to the time the TMS pulse occurred, in ms, used for peak-to-peak

and latency calculations.

- `max-emg-time-range-ms` (required, float) : The ending time, relative to the time the TMS pulse occurred, in ms, used for peak-to-peak and latency calculations.
- `waveform-info` (required, array of dictionaries) : A sorted array of dictionaries describing the EMG waveform data for each EMG channel. For the Brainsight EMG Pod, there are at most 2 channels. For the NEURO PRAX, there can be many more. The array is sorted by channel index. Each dictionary has the following keys:
  - `device-type` (required, integer) : Indicates the device that provided the EMG data: 1 for Brainsight EMG Pod, 2 for NeuroConn NEURO PRAX.
  - `channel-name` (optional, string) : Indicates the channel name, for example `EMG_16`. Will always be present for data from NEURO PRAX, and never be present for data from Brainsight EMG Pod.
  - `channel-index` (required, integer) : Indicates the zero-based channel index. For Brainsight EMG Pod, 0 for channel 1, and 1 for channel 2. For NEURO PRAX, this matches the montage order.
  - `peak-to-peak-uV` (required, float) : the EMG peak-to-peak value for the channel, in µV. The peak to peak is calculated for the time range between `min-emg-time-range-ms` and `max-emg-time-range-ms`.
  - `latency-ms` (optional, float) : the EMG latency value for the channel, in ms. The latency is calculated for the time range between `min-emg-time-range-ms` and `max-emg-time-range-ms`. This will usually be present, but it is possible for the latency algorithm to fail, for example if the time range is very small.
  - `acquisition-uV` (optional, array of floats) : The EMG waveform data for the channel. Each number represents a voltage, in µV, taken at the device's sampling frequency. For example, the Brainsight EMG Pod device samples at 3 kHz, so there will be 3000 numbers for each second of EMG data.
  - `sampling-interval-ms` (required, float) : The waveform's sampling interval (reciprocal of sampling frequency). For example, the Brainsight EMG Pod device samples at 3 kHz, so this will be 0.3333333333 ms there.
  - `origin-time-ms` (required, float) : The waveform's origin time, in ms. This represents the location of the first sample in the acquisition relative to where time=0 is defined. For example, a waveform created in response to a TMS stimulation records data before the stimulation. The stimulation is defined to be at time=0 and this offset tells how much data was recorded before that.
- also, to simplify bookkeeping, the following attributes from `stream:sample-creation` are repeated, see there for their meaning:

        * creation-cause
        * creation-date
        * crosshairs-mode
        * crosshairs-offset
        * crosshairs-twist
        * target-name
        * target-position
        * coordinate-system
        * position

Example stream:


```
{
    "packet-name": "stream:sample-emg",
    "packet-uuid": "03AAD82D-31C2-4806-9276-068EB88963B1",

    "name": "Sample 1",
    "index": 1,
    "position": [
        -1,  0, 0,     0,
         0, -1, 0, 18.25,
         0,  0, 1, 16.86,
         0,  0, 0,     1
    ],
    "creation-cause": 6,
    "creation-date": "2024-02-05T18:32:37.846Z",
    "crosshairs-mode": "Pointer",
    "crosshairs-offset": 0,
    "crosshairs-twist": 0,
    "coordinate-system": "NIfTI:S:Scanner",
    "target-name": "Target 2",
    "target-position": [
        -0.69, -0.53, -0.47,   97.8,
         0.67, -0.71, -0.18, 118.55,
        -0.24, -0.45,  0.85, 150.84,
            0,     0,     0,      1
    ]
    "uuid": "597EAD78-1E26-4EA9-949C-3708484F8427",

    "min-emg-time-range-ms": 10.5,
    "max-emg-time-range-ms": 76.1,

    "waveform-info": [
      {
        "device-type" : 1,
        "channel-index": 0,
        "peak-to-peak-uV": 550.5,
```

```
        "latency-ms": 25.3,
        "acquisition-uV": [0.0, 0.9, 3.7, 34.5],
        "sampling-interval-ms": 0.3333333333,
        "origin-time-ms": -0.5
      }
    ]
}
```

Polaris camera updated with new tool positions

This packet is sent whenever a Polaris tool location changes. One such packet is sent for every tool being tracked. The Polaris updates at 20 or 60 Hz, depending on model, so this packet is frequent.

Stream format:

- `packet-name` (required, string) : The string `stream:session-polaris-update`.
- `frame-number` (optional, integer) : A monotonically increasing integer representing the Polaris' internal clock. This is always present if the tool is visible.
- `serial-number` (required, string) : The serial number of the tool, like ST-123 or P-456. Each tool has a unique serial number.
- `tool-in-sensor` (optional, array of 16 floats) : The location and orientation (as a 4x4 matrix) of the tool in the Polaris' coordinate system. This is provided when the tool is visible to the camera, and absent when thee tool is not visible. See Polaris documentation for details of this coordinate system. In brief, the origin is at the midpoint between the two camera lenses.
- `tool-in-desired` (optional, array of 16 floats) : The location and orientation (as a 4x4 matrix) of the tool in the `coordinate-system` coordinate system. The coordinate system of the session's Perform step will be used. This is provided only when subject registration has been performed and when the toolsa are visible to the camera; it is absent otherwise.
- `coordinate-system` (required, string) : The coordinate system of `tool-in-desired`. See Appendix III for details.
- `timestamp` (required, string) : The date/time that the data was received from the Polaris. See Appendix II for format details. For more exact time keeping, use the `frame-number`.

Example stream:

```
{
    "packet-name": "stream:session-polaris-update",
    "packet-uuid": "47A2E0F8-3F98-447F-A282-ACC8C1A002F0",

    "frame-number": 34255,
    "serial-number": "ST-123",
```

```
    "tool-in-sensor": [
        0.7065256352896276, 0.7076874498523965, 0, -100,
        -0.7076874498523965, 0.7065256352896276, 0, 0,
        0, 0, 1, -1750,
        0, 0, 0, 1
    ]
    "tool-in-desired": [
        -0.69, -0.53, -0.47,   97.8,
         0.67, -0.71, -0.18, 118.55,
        -0.24, -0.45,  0.85, 150.84,
            0,     0,     0,      1
    ]
    "coordinate-system": "World",
    "timestamp": "2024-07-26T15:50:59.123Z"
}
```

IOBox TTL trigger occurred

This packet is sent whenever one of the TTL triggers on the IOBox is triggered.

Stream format:

- `packet-name` (required, string) : The string `stream:session-ttl-triggers`.
- `timestamp` (required, string): The date and timestamp of when the trigger occurred. See Appendix II for format details.
- `ttl1` (required, boolean) : True if TTL1 was triggered, false if it not.
- `ttl2` (required, boolean) : True if TTL2 was triggered, false if it not.
- `ttl-switch` (required, boolean) : True if TTLSwitch was triggered, false if it not.

Example stream:

```
{
    "packet-name": "stream:session-ttl-triggers",
    "packet-uuid": "FC77310D-DD10-4C01-92D6-BBF848BFFBE0",

    "timestamp": "2024-06-19T16:57:29.834Z",
    "ttl1": true,
    "ttl2": false,
    "ttl-switch": false
}
```

Appendix I: Error Codes

The following numerical values may be returned as the `error-code` is a response packet. Their name should give an idea

- `kBSErrorCode_NoError = 0`
- `kBSErrorCode_PacketInvalidJSON = 100`
- `kBSErrorCode_PacketNameInvalid = 101`
- `kBSErrorCode_PacketUUIDInvalid = 102`
- `kBSErrorCode_RequiredFieldMissing = 103`
- `kBSErrorCode_WrongType = 104`
- `kBSErrorCode_TooLong = 105`
- `kBSErrorCode_TooShort = 106`
- `kBSErrorCode_InvalidCombination = 107`
- `kBSErrorCode_NoDocuments = 201`
- `kBSErrorCode_MoreThanOneDocument = 202`
- `kBSErrorCode_NoActiveSession = 301`
- `kBSErrorCode_NoActiveSessionWithName = 302`
- `kBSErrorCode_PerformStepNotLoaded = 303`
- `kBSErrorCode_MatrixSizeNot4x4 = 401`
- `kBSErrorCode_CrazyFloatingPoint = 402`
- `kBSErrorCode_NonInvertibleMatrix = 403`
- `kBSErrorCode_NonRigidMatrix = 404`
- `kBSErrorCode_CoordinateSystemUnknown = 501`
- `kBSErrorCode_GeneralSampleCreationFailure = 601`
- `kBSErrorCode_InvalidStreamName = 801`
- `kBSErrorCode_NoTargetWithName = 901`
- `kBSErrorCode_NoTargetWithIndexPath = 902`

Appendix II: Date and time format

All timestamps used in this protocol are strings in ISO 8601 format. They are always in the UTC time zone. They include 3 digits of milliseconds. The exact Unicode date format pattern used is `yyyy-MM-dd'T'HH:mm:ss.SSS'Z'`.

Appendix III: Positions, orientations, coordinate systems

- Positions/orientations are represented as 4x4 matrices, as an array of 16 floating point numbers. They are in row-major order. The matrix must be invertible. The position components are in mm.
- Coordinate systems are identified by one of the following strings:
  - `Brainsight`
  - `World`
  - `NIfTI:S:Scanner`
  - `NIfTI:Q:Scanner`
  - `NIfTI:S:Aligned`
  - `NIfTI:Q:Aligned`
  - `NIfTI:S:MNI-152`
  - `NIfTI:Q:MNI-152`
  - `NIfTI:S:Talairach`
  - `NIfTI:Q:Talairach`
  - `NIfTI:S:Other-Template`
  - `NIfTI:Q:Other-Template`
  - `MNI`
  - `MNI Rhesus`
  - `MNI Cynomolgus`
  - `MNI Macaque`
  - `Talairach`
  - `Paxinos`
  - `Fraunhofer Sheep`
  - `Saleem D99`
  - `Marmoset`
  - `Pig`
  - `Cornell Canine`

These correspond to same names used in the Brainsight user interface.

Appendix IV: Crosshairs Modes

The 'crosshairs mode' corresponds to the selection in the "Driver" popup menu in the Brainsight session window.

- `None` - corresponds to "None" in the "Driver" popup menu.
- `Mouse` - corresponds to "Mouse" in the "Driver" popup menu.
- `Pointer` - corresponds to "Pointer" in the "Driver" popup menu.
- `Placed Chamber` - corresponds to selecting a placed chamber in the the "Driver" popup menu.
- `Axilum Arm` - corresponds to "Axilum Arm" in the "Driver" popup menu.
- `Vet Robot` - corresponds to "Vet Robot" in the "Driver" popup menu.
- or the name of a Tool Calibration
- `Unknown` - a fallback in the unlikely case that none of the above apply.

Appendix V: Sample Creation Causes

The following are possible integer values for a sample's 'creation cause', they indicate what mechanism was used to create the sample.

- 0: created by pressing the "Sample Now" button in Brainsight's Perform window.
- 1: created by speech recognition, for example, speaking the phrase "Sample".
- 2: created by a press of an Apple Remote play button .
- 6: created by the TTL1 trigger on the IOBox.
- 7: created by the TTL2 trigger on the IOBox.
- 8: created by the Switch trigger on the IOBox.
- 9: created by a Neuro PRAX device communicating to Brainsight.
- 10: created by a client using the network protocol described in this document.
- 5: a fallback in the unlikely case that none of the above apply.

Appendix VI: Version history

- 1.0:
  - Initial version. Shipped in Brainsight 2.5.6.
- 1.0.1:
  - In the `request:create-target-at-location` packet, the `position` and `coordinate-system` fields were made optional. Shipped in Brainsight 2.5.7.