BRAINLAB

IGTLink INTERFACE SPECIFICATION

Version 1.0

Software User Guide
Revision 1.2

TABLE OF CONTENTS

1 GENERAL INFORMATION

1.1 Contact Data and Legal Information

1.1.1 Contact Data

Support

If you cannot find information you need in this guide, or if you have questions or problems, contact Brainlab support:

Region	Telephone and Fax	Email
United States, Canada, Central and South America	Tel: (800) 597-5911 Fax: (708) 409-1619	us.support@brainlab.com
Brazil	Tel: (0800) 892-1217	
UK	Tel: +44 1223 755 333	support@brainlab.com
Spain	Tel: +34 (900) 649 115	
France and French-speaking regions	Tel: +33 800 676 030	
Africa, Asia, Australia, Europe	Tel: +49 89 991568-44 Fax: +49 89 991568-811	
Japan	Tel: +81 3 3769 6900 Fax: +81 3 3769 6901	

Feedback

Despite careful review, this manual may contain errors.

Please contact us at igs.manuals@brainlab.com if you have suggestions as to how we can improve this manual.

Manufacturer

Brainlab AG
Kapellenstr. 12
85622 Feldkirchen
Germany

1.1.2 Legal Information

Copyright

Brainlab Trademarks

The following are trademarks of Brainlab AG® in Germany and/or the US:

- **BrainSUITE**®
- **Curve**™
- **IGSonic**®
- **Kick**®
- **Kolibri**™
- **Softouch**®
- **StarLink**®
- **VarioGuide**™
- **VectorVision**®
- **Z-touch**®

Non-Brainlab Trademarks

- Mobius Imaging®, the Mobius Imaging logo, and MobiCT® are registered trademarks of Mobius Imaging, LLC.
- Doro® is a registered trademark of pro med instruments GmbH.
- Mayfield® is a registered trademark of OMI, Inc.
- Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and other countries.
- MultiVision™ is a trademark of Carl Zeiss Meditec AG.
- OPMI®,Pentero®, and PENTERO® are registered trademarks of Carl Zeiss Meditec AG.

Integrated 3rd-Party Software

This software is based in part on the work of the Independent JPEG Group.

CE Label

CE 0123

The CE mark shows that the Brainlab products comply with the essential requirements of the Medical Device Directive.

According to the MDD (Council Directive 93/42/EEC):

- **Cranial/ENT** is a Class IIb product.
- **IGSonic 10V5** and **IGSonic 3000** are Class IIa products.

Disposal Instructions

Only dispose of electrical and electronic equipment in accordance with statutory regulations. For information regarding the WEEE (Waste Electrical and Electronic Equipment) directive, visit:

http://www.brainlab.com/weee

Sales in the US

US federal law restricts this device to sale by or on the order of a physician.

1.2 Symbols

1.2.1 Symbols Used in This Guide

Warnings

Warnings are indicated by triangular warning symbols. They contain safety-critical information regarding possible injury, death or other serious consequences associated with equipment misuse.

Cautions

Cautions are indicated by circular caution symbols. They contain safety-critical information regarding possible problems with the device. Such problems include device malfunctions, device failure, damage to device or damage to property.

Notes

NOTE: Notes are formatted in italic type and indicate additional useful hints.

1.3 Compatibility with Medical Devices

1.3.1 Brainlab Medical Software

Compatible Brainlab Medical Software

IGTLink is compatible with:

- **Cranial/ENT** Ver. 2.1
- **Cranial/ENT** Ver. 3.0 and 3.1

Please refer to the relevant Software User Guide for more information.

Other Brainlab Software

If you are running software versions other than those specified above, contact Brainlab support for clarification regarding compatibility with Brainlab devices.

Only Brainlab medical software specified by Brainlab may be installed and used with the system.

1.4 Documentation

1.4.1 Documentation

Intended Audience

This manual is intended for software developers able to implement the OpenIGTLink interface. For anything else refer to the relevant Cranial/ENT Software User Guide.

Reading User Guides

The user guides describe complex medical devices and surgical navigation software that must be used with care.

It is important that all users of system, instruments and software:

- Read the user guides carefully before handling the equipment
- Have access to the user guides at all times

Available User Guides

User Guide	Contents
Software User Guides	• Overview of treatment planning and image-guided navigation • Description of OR system setup • Detailed software instructions
Instrument User Guides	Detailed instructions on instrument handling
Cleaning, Disinfection and Sterilization Guide	Details on cleaning, disinfecting and sterilizing instruments
System User Guides	Comprehensive information on system setup
Technical User Guide	Detailed technical information on the system, including specifications and compliances

2 IGTLink Interfaces

2.1 About IGTLink

2.1.1 Overview

General Information

IGTLink is a network based software interface to Brainlab's navigation system. It enables researchers and skilled practitioners to download medical images, segmentation results, surgical planning and real-time tool tracking data from the navigation system.

Using **IGTLink**, you can generate customized displays and to implement novel applications while benefiting from Brainlab's well-proven tools for pre-operative image data import, image registration and segmentation as well as intra-operative patient registration and instrument tracking.

Client API

A client API is available for building IGTLink clients. It implements the data exchange protocol between client and server based on OpenIGTLink developed by the National Alliance for Medical Image Computing (NA-MIC), see http://openigtlink.org/.

This guide contains the interface specification. Please consult it when implementing IGTLink client applications.

Safety Notes

Brainlab offers the IGTLink interface. Brainlab assumes no responsibility for data processing outside the navigation system. Brainlab shall not be liable for damage of whatever nature resulting from or in connection with data processing outside the navigation system.

The IGTLink interface shall be used in accordance with its specifications.

Brainlab shall not be liable for any damage arising of any use of IGTLink that is not in accordance with its specifications.

If you want to perform clinical studies which involve the use of IGTLink, all legal requirements of local legislation must be fulfilled.

The user is advised to ensure the safety of the network connected to the navigation system by taking appropriate measures.

Transferred data is not encrypted. The user is advised to ensure the confidentiality of the patient data transferred to and from the navigation system by taking appropriate measures.

Depending on the network and computer performance as well as the data size of messages, transmitted data might be out of date. No assumptions about latency of transferred data shall be made.

Be aware that connections can be closed by the server without informing the client directly.

Transmitted image data may be re-sampled giving degraded image quality e.g., when using non-orthogonal image sets or voxel objects. The pixel size may also differ.

2.1.2 Setting Up a Client

Connection to the IGTLink server

The connection to **IGTLink** server is established using a TCP/IP socket. The IP address or hostname and a port are necessary connection information.

Performing Queries and Data Transfer

The server system contains static and dynamic information.

- Static information is data that is constant over time or changes very rarely. Examples of static information are slice sets or labeled points.
- Dynamic information is data that vary over time such as tracking data.

IGTLink supports both types.

- Static data can be queried. This means that a query is sent to the navigation system and the response is sent back to the **IGTLink** client.
- Dynamic data must be updated whenever it changes. Using a special query, an automatic update can be setup, which means the server sends messages to the client every time new data is available.

An example IGTLink using the provided Client API:

```
// CONNECT
igtlink::Link link;
if (!link.connect('hostname', 22222))
return;

// STATIC DATA (first slice set)
if (!link.send(&SendGetImgMeta())) // get meta data of all slice sets
return;
std::auto_ptr<RecvCommand> cmd =
std::auto_ptr<RecvCommand>(link.recv());
igtlink::RecvImgMeta* meta =
dynamic_cast<igtlink::RecvImgMeta*>(cmd.get());
if (!meta || meta->getDataCount() < 1 || !link.send(&SendGetImage(meta-
>getId(0)))) // get data of first slice set if available
return;
cmd = std::auto_ptr<RecvCommand>(link.recv());
igtlink::RecvImage* image =
dynamic_cast<igtlink::RecvImage*>(cmd.get()); // get data of first slice
set
// do something with the slice set

// DYNAMIC DATA
if (!link.send(&igtlink::SendSttTData())) // start tracking update
return;
cmd = std::auto_ptr<igtlink::RecvCommand>(link.recv());
igtlink::RecvRtsTData* rtsTData =
dynamic_cast<igtlink::RecvRtsTData*>(cmd.get());
if (!rtsTData || rtsTData->getDataCount() != 1 || !rtsTData-
>getResult())
return;
// else: starting the update was successful

while(true) {
 cmd = std::auto_ptr<igtlink::RecvCommand>(link.recv());
```

```
igtlink::RecvTData* tdata =
dynamic_cast<igtlink::RecvTData*>(cmd.get());
if (tdata) {
// do something with the tracking data
}
}
```

Coordinates and Coordinate Systems

The server system uses coordinates in different coordinate systems in order to specify where points are in space. Not all coordinate systems are connected, e.g. if slice sets are not fused or the patient is not registered.

The coordinate system of the first slice set is used as the **patient coordinate system**. Every object, which is connected to this coordinate system, is available via **IGTLink**. These objects are always specified in this coordinate system, so no coordinate translation is necessary.

Exception: tracking data can be optionally queried in the **camera coordinate system** to allow the development of tracking applications without having to load or register a patient.

The unit is “mm”.

2.2 Command Specification

2.2.1 Overview

General Information

The following specification is a subset of OpenIGTLink version 2. To be compatible with Brainlab's **IGTLink**, it is not necessary to implement the whole standard protocol; it is enough to support the commands as shown below.

- Every command consists of a common 58 byte header and a body.
- The endian type is always big endian.
- A string shorter than the appropriate field length is padded with 0.
- All GET_<…> messages as well as STT_TDATA and STP_TDATA are used to be sent from client to server, the other messages are used to be sent from server to client.

Header

Data	Type	Description
Version	16 bit unsigned	The version number must be always 1.
Type name	char[12]	The type name specifies the type of the message, e.g. GET_IMAGE; if a type name is not known to the system, it is legal to read the following Body size bytes from the input buffer and to ignore the command.
Device name	char[20]	The device name is used if several devices of a type exist, e.g. every slice sets hat its own ID.
Time stamp	64 bit unsigned	The time stamp is not used.
Body size	64 bit unsigned	Size of body in bytes, can be 0.
CRC	64 bit unsigned	64 bit CRC for body data (not the header).

2.2.2 STATUS

Description

After establishing the connection through TCP/IP, a dialog pops up on the navigation system to explicitly accept or deny the connection. The result of this decision is sent to the client via a STATUS message.

If the connection has been accepted, the client is allowed to send further commands to query data. Otherwise the connection will be terminated automatically by the server after sending the STATUS answer.

Device Name

Empty

Body

Data	Type	Description
Code	16 bit unsigned	1: Connection accepted. 2: Internal error. 5: Connected denied.
Sub Code	64 bit unsigned	Not used.
Name	char[20]	Not used.
Message	char[Body size - 30]	Not used.

2.2.3 GET_CAPABIL

Description

Get a list of commands, which are accepted by the navigation system.

Device Name

Empty

Body

Empty

2.2.4 CAPABILITY

Device Name

Empty

Body

The body is repeated for every accepted command:

- GET_CAPABIL
- GET_POINT
- GET_TRAJ
- GET_IMGMETA
- GET_LBMETA
- GET_IMAGE
- GET_COLORT
- STT_TDATA
- STP_TDATA

Data	Type	Description
Type name	char[12]	Known type name.

2.2.5 GET_IMGMETA

Description

.Query for Meta data of slice sets.

Device Name

A slice set ID known from a previous query to get the meta data for a specific slice set. If the device name is empty, the meta data of every available slice set is returned.

Body

Empty

2.2.6 IMGMETA

Device Name

Same as used in GET_IMGMETA.

Body

If no slice set is available, the body is empty. The body will be repeated multiple times if device name is empty and more than one slice set is available.

Data	Type	Description
Name	char[64]	Name of the slice set.
ID	char[20]	ID to query the IMAGE and COLORT.
Modality	char[32]	String which specifies the modality.
Patient name	char[64]	Name of the corresponding patient.
Patient ID	char[64]	ID of the corresponding patient.
Timestamp integer	32 bit unsigned	Scan time in seconds relative to 00:00:00 January 1, 1970, UTC.
Timestamp fraction	32 bit unsigned	Not used.
RI, RJ, RK	16 bit unsigned	Number of pixels in each direction (same as in IMAGE).
S	8 bit unsigned	Scalar type (same as in IMAGE): 3: 8 bit unsigned int 5: 16 bit unsigned int
RESERVED	8 bit unsigned	Not used.

Safety Notes

IDs can change when reloading the patient or restarting the navigation software. In these cases the meta data has to be queried again.

The original slice set name, modality, patient name and ID from server have 16 bit encoding. They are converted into 8-bit for sending. Information loss may occur while converting.

2.2.7 GET_LBMETA

Description

Query for meta data of voxel objects.

Device Name

A voxel object ID known from a previous query to get the meta data for a specific voxel object. If the device name is empty, the meta data of every available voxel objects is returned.

Body

Empty

2.2.8 LBMETA

Device Name

Same as used in GET_LBMETA.

Body

If no voxel object is available, the body is empty. The body will be repeated multiple times if device name is empty and more than voxel object set is available.

Data	Type	Description
Name	char[64]	Name of the voxel object.
ID	char[20]	ID to query the IMAGE.
Label	8 bit unsigned	Always 0.
Reserved	8 bit unsigned	Not used.
R, G, B, A	8 bit unsigned	Color in RGBA.
RI, RJ, RK	16 bit unsigned	Number of pixels in each direction (same as in IMAGE), bounding box of the structure(s).
Owner image	char[20]	ID of the owner slice set.

Safety Notes

IDs can change when reloading the patient or restarting the navigation software. In these cases the meta data has to be queried again.

The original voxel object name from server has 16 bit encoding and is converted into 8-bit for sending. Information loss may occur while converting into strings.

2.2.9 GET_IMAGE

Description

Query for a slice set or a voxel object, depending on the device name field.

Device Name

The ID of a voxel object or slice set known from IMGMETA or LBMETA.

Body

Empty

2.2.10 IMAGE

Device Name

Same as used in GET_IMAGE.

Body

If the slice set or the voxel object is not available, the body is empty.

Data	Type	Description
Version	16 bit unsigned	The version number must be always 1.
Image type	8 bit unsigned	Always 1.
Scalar type	8 bit unsigned	3: 8 bit unsigned int. 5: 16 bit unsigned int.
Endian type	8 bit unsigned	Always 1 (means big endian).
Image coord	8 bit unsigned	Always 1 (means RAS coordinates).
RI, RJ, RK	16 bit unsigned	Number of pixels in each direction.
TX, TY, TZ	32 bit float	Vector 'i' / length pixel size in 'i' direction.
SX, SY, SZ	32 bit float	Vector 'j' / length pixel size in 'j' direction.
NX, NY, NZ	32 bit float	Vector 'k' / length pixel size in 'k' direction.
PX, PY, PZ	32 bit float	Center position of the image.
DI, DJ, DK	16 bit unsigned	Always 0,0,0 (subvolume not used).
DRI, DRJ, DRK	16 bit unsigned	Always RI, RJ, RK (subvolume not used).
Data	Binary data	Voxels of the slice set; size of field: RI*RJ*RK of type S.

Safety Notes

Non-orthogonal slice sets or voxel objects will be reconstructed. This can degrade the image quality and change properties of the original slice set such as the number of slices or slice thickness.

The transfer of the image data from a slice set might block the server system for a short period of time (e.g. during re-sampling). Avoid repeating slice set transfers, and especially avoid making transfers in a program loop!

The voxel object pixel size can differ from the slice set which owns the voxel object.

2.2.11 GET_COLORT

Description

Query for a color table of a slice set.

Device Name

The ID of a slice set known from IMGMETA.

Body

Empty

2.2.12 COLORT

Device Name

Same as used in GET_COLORT.

Body

Data	Type	Description
Index type	8 bit unsigned	3: 8 bit unsigned int. 5: 16 bit unsigned int.
Map type	8 bit unsigned	Always 19 (means RGB color).
Table	Binary data	Table size: 2^8 or 2^16 * 3 Byte.

If body size is 0, no color table is defined for the slice set. A linear gray windowing can be assumed in this case.

2.2.13 GET_POINT

Description

.Query for points.

Device Name

Empty

Body

Empty

2.2.14 POINT

Device Name

Empty

Body

The body will be repeated multiple times if more than one point is available.

Data	Type	Description
Name	char[64]	Name or description of the point.
Group name	char[32]	Can be “LabeledPoint”, “Marker”, “IntraOpLandmark”, ...
R, G, B, A	8 bit unsigned	Color.
X, Y, Z	32 bit float	Coordinate of the point.
Diameter	32 bit float	Diameter of the point; can be 0.
Owner image	char[20]	ID of the owner slice set.

Safety Notes

The original point name from server has 16-bit encoding and is converted into 8-bit for sending. Information loss may occur while converting into a string.

2.2.15 GET_TRAJ

Description

Query for trajectories.

Device Name

Empty

Body

Empty

2.2.16 TRAJ

Device Name

Empty

Body

The body will be repeated multiple times if more than one trajectory is available.

Data	Type	Description
Name	char[64]	Name or description of the trajectory.
Group name	char[32]	Can be “Trajectory”, “TargetTrajectory”, ...
Type	8 bit unsigned	1: only entry point. 2: only target point. 3: entry and target point.
Reserved	8 bit unsigned	Not used.
R, G, B, A	8 bit unsigned	Color.
X1, Y1, Z1	32 bit float	Entry point of the trajectory.
X2, Y2, Z2	32 bit float	Target point of a trajectory.
Diameter	32 bit float	Diameter of the trajectory; can be 0.
Owner image	char[20]	ID of the owner slice set.

Safety Notes

The original trajectory name from server has 16-bit encoding and is converted into 8-bit for sending. Information loss may occur while converting into a string.

2.2.17 STT_TDATA

Description

The server starts sending tracking data via TDATA messages.

This can be stopped with a STP_TDATA message. It is valid to send another STT_TDATA message without sending STP_TDATA, the previous notification will be stopped automatically.

A STT_TDATA message is answered by a RTS_TDATA message.

Device Name

Empty

Body

Data	Type	Description
Resolution	32 bit unsigned	Minimum time between two frames in ms. • 0 ms: the server sends a TDATA message every time when new data is available. • 50 ms: the maximum update rate will be 20 Hz. If the resolution is set to a value greater than 0, the tick count time on the server system might influence the frame rate. Example: Resolution = 50 ms, tick count time = 16.67 ms à worst case = 15 Hz, best case = 20 Hz.
Coordinate system name	char[32]	• “CAMERA”: camera coordinate system • Anything other then “CAMERA”: patient coordinate system. See page 14 for more information.

2.2.18 STP_TDATA

Description

The server stops sending tracking data. A STP_TDATA message is answered by a RTS_TDATA message. No more TDATA messages will be sent.

Device Name

Empty

Body

Empty

2.2.19 RTS_TDATA

Description

This message is sent as answer for STT_TDATA and STP_TDATA.

Device Name

Empty

Body

Data	Type	Description
Status	8 bit unsigned	• 0: Success • 1: Error

2.2.20 TDATA

Description

The TDATA message type is used to transfer 3D positions of surgical tools, markers etc. Those positions are continuously transferred as series of messages.

As it is important for software that receives TDATA to control the data flow, STT_TDATA query data type uses the interval field to control the frame rate of consecutive messages.

Device Name

Empty

Body

The body will be repeated multiple times if more than one tracker or instrument is available.

Available means the tracker or instrument is:

- Connected to the chosen coordinate system
- Visible
- Intended for the use of **IGTLink**.

If a tracker or instrument is not included, it is assumed to be not visible. If no tracker or instrument is visible, the body is empty.

Data	Type	Description
Name	char[20]	Name (=ID) of the instrument or tracker.
Type	8 bit unsigned	• 1: tracker. • 2: 6D instrument (regular instrument). • 3: 3D instrument (only tip of the instrument defined). • 4: 5D instrument (tip and handle are defined, but not the normal vector).
Reserved	8 bit unsigned	Not used.
Matrix	32 bit float	12 values as in a TRANSFORM message, means the following order (1 to 12): $\begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \\ ″0″ & ″0″ & ″0″ & ″1″ \end{bmatrix}$ The last row is always 0, 0, 0, 1.

Safety Notes

Tracking data is sent from server to client without waiting for acknowledgment. Before processing tracking data, ensure that the TCP/IP receive buffer does not contain any newer tracking data. To avoid being overwhelmed by tracking updates, choose an appropriate resolution value when starting the notification.

INDEX

BRAINLAB

brainlab.com

Art-No. 60914-90EN