IBEX Bone Health v1.5.0
Instructions For Use
Integration Manual

Document Revision 1

March 3rd 2025

Document approvals:

Signature: David Joyce (Mar 3, 2025 15:05 GMT) Product Manager	Print: David Joyce	Date: Mar 3, 2025
Signature: Paul Scott CEO	Print: Paul Scott	Date: Mar 3, 2025

Contents

Overview

Instructions for use of the IBEX Bone Health software are separated by topics relevant to the user.

The end users manual (IBX_PRD_SOP_002) provides an overview of the intended use as well as information on the indications for use, contraindications, suggested use cases, and limited troubleshooting advice. All of the information needed by the end user for safe and effective use of the product, including background on the technical basis and clinical validation of the product is provided in this manual.

The service manual (IBX_PRD_SOP_003) provides guidance to the engineer on how to configure the product for use. The instructions include details on licensing, system calibration, settings optimisation, and troubleshooting. Engineers should familiarise themselves with the contents of the end users manual so they have a good understanding of how the product will be used.

The integration manual (IBX_PRD_SOP_004) is intended for developers integrating IBEX BH into OEM systems. Guidance on hardware requirements, API use, and system optimisation is provided. Integrators should familiarise themselves with both the service manual and the end users manual for a complete understanding of the product and its use.

Revision History

Starting from IBEX BH v1.5.0 the instructions for use will be split into three manuals; one each for end users, service engineers, and integrators. To ensure consistency across all versions of the product, vendor-specific IFUs have been merged into a single set of documents describing the configuration and use of IBEX BH. For traceability purposes, the revision histories for previous versions of the manuals are shown below.

Revision	Date	Comment	Initials
1	2024/01/26	First release IFU for IBEX Bone Health v1.1	BL
2	2024/03/08	IFU for IBEX Bone Health v1.2 including customer feedback and error corrections from the first version; clarification of user responsibilities and activities; additional information on errors and warnings; additional information on system configuration including output configuration and licensing; removing the kV offset setting; adding information on deep learning components.	DJ
3	2024/05/22	IFU for IBEX Bone Health v1.3 including calibration workflow simplification; updated default aBMD normative data; new APIs for flushing pipeline and deleting records; new settings for licensing configuration, report configuration, and HTTP request size limit; minor additions to the licensing status API; updated screenshots to the latest interface.	DJ, CS
v1.3rev2	2024/05/29	Corrected information in [Production and Service Engineers] Section 2.2.6.4 on the management of the "calibration QA mode" setting.	CS
v1.3rev3	2024/06/13	Added instructions for native integrations. Added documentation for new settings on report configuration (rotation and clip). Added recommendations on password storage and sharing. Clarify that passwords may only contain ASCII characters.	CS
v1.4rev1	2024/12/11	Added documentation for new settings on report configuration (scaling and version label). Added description of new error code E210. Added details on the training and testing of deep learning elements of the product, and other minor edits.	CS, DJ

Table 1: Instructions for Use IBEX Bone Health, AGF_01_PRD_SOP_004 (DISCONTINUED)

Revision	Date	Comment	Initials
0.1	2024/02/14	First release IFU for IBEX Bone Health v1.1	BL
v1.3rev1	2024/06/13	IFU for IBEX Bone Health v1.3 including customer feedback and error corrections from the first version; clarification of user responsibilities and activities; additional information on errors and warnings; additional information on system configuration including output configuration and licensing; removing the kV offset setting; adding information on deep learning components. Also including calibration workflow simplification; updated default aBMD normative data; new APIs for flushing pipeline and deleting records; new settings for licensing configuration, report configuration, and HTTP request size limit; minor additions to the licensing status API; updated screenshots to the latest interface. Corrected information in [Production and Service Engineers] Section 2.2.6.4 on the management of the "calibration QA mode" setting. Added instructions for native integrations. Added documentation for new settings on report configuration (rotation and clip). Added recommendations on password storage and sharing. Clarify that passwords may only contain ASCII characters.	DJ, CS
v1.3rev2	2024/05/29	Corrected information in [Production and Service Engineers] Section 2.2.6.4 on the management of the "calibration QA mode" setting.	CS
v1.3rev3	2024/06/13	Added instructions for native integrations. Added documentation for new settings on report configuration (rotation and clip). Added recommendations on password storage and sharing. Clarify that passwords may only contain ASCII characters.	CS
v1.4rev1	2024/11/15	Added documentation for new settings on report configuration (scaling and version label). Added details on use of IBEX BH in a mammography setting, details on the training and testing of deep learning elements of the product, and other minor edits.	CS, DJ
v1.4rev2	2024/12/11	Added documentation for new settings on report configuration (bone edges scaling). Added description of new error code E210.	CS

Table 2: Instructions for Use IBEX Bone Health, MET_01_PRD_SOP_001 (DISCONTINUED)

Revision	Date	Comment	Initials
v1.5rev1	2025/03/03	IFU for IBEX Bone Health split into three documents (end users manual, service manual, and integration manual). Document links updated. Add references to the technical specification. Specify which warnings are raised in Section 2 and add brief summary of the QA procedure. Add footnote in Section 3.4.1 for processing with raw image only (new in v1.5.0). Fixed references in Sections 3.5 and 3.5.1 only valid for sandboxed installations, and add information about user configuration and permissions (new in v1.5.0). Add settings (new in v1.5.0) to Section 3.5.1: `receiver.port`, `processing.qa_expiry_days`, `report.series_description`, `report.series_number_policy`, `report.series_number`, `report.require_presentation_dataset`, `report.script`, `report.watermark_enabled.206`, `report.watermark_enabled.211`, `report.watermark_label.206`, `report.watermark_label.211`, `report.watermark_position`, `report.watermark_xray_only`, `report.watermark_brightness`, `report.watermark_scale`, `system.data_encryption_scheme`, `system.retain_datasets`. Added missing documentation for `first_use` field in Section 3.7.4. Add files API to Section 3.9.1 (new in v1.5.0).	CS, DJ, BL

Table 3: Instructions for Use IBEX Bone Health, `IBX_PRD_SOP_004`

Terms and Definitions

Term	Definition
Central Osteoporosis	An individual with a measured T-score ≤ -2.5 at the femoral neck or lumbar spine is defined as having central osteoporosis.
Customer/End User	Clinical end user or medical physicist — responsible person.
Fingerprint	The set of images used to produce calibration files by IBEX.
Fingerprinting	The process of collecting the set of images used to produce calibration files by IBEX.
IBEX	Engineer or other appropriate personnel appointed by IBEX.
Integrator	Production, service, or other appropriate personnel appointed by the system manufacturer.
T-score	Score created by comparing the patient's bone density to the average bone density of a healthy population of the same sex between 20 and 30 years of age. A score of 0 indicates normal bone health, and a score of -1 indicates bone health lower than the norm by one standard deviation. Osteoporosis is typically indicated by a score lower than -2.5 at the neck of femur.
Working hours	Core working hours are Monday through Friday from 10am - 5pm CET. Response times outside of these hours cannot be guaranteed but special arrangements for out of hours support can be made by prior arrangement if necessary.

Abbreviations

Abbreviation	Meaning
AET	Application Entity Title
AP	Anterior Posterior
API	Application Programming Interface
aBMD	areal Bone Mineral Density
CPU	Central Processing Unit
DICOM	Digital Imaging and Communications in Medicine
DP	Dorsi-Palmar
DR	Digital Radiograph
DXA	Dual-energy X-ray Absorptiometry
FRAX	Fracture Risk Assessment Tool
GPU	Graphics Processing Unit
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IBEX BH	IBEX Bone Health
IP	Internet Protocol
JSON	JavaScript Object Notation
NHS	National Health Service
NICE	National Institute for Healthcare and Excellence
NoF	Neck of Femur
OEM	Original Equipment Manufacturer
PA	Posterior Anterior
PACS	Picture Archiving and Communication System
PMMA	Polymethyl Methacrylate
QA	Quality Assurance
RAM	Random Access Memory
ROI	Region Of Interest
SCP	Service Class Provider
SCU	Service Class User
SDK	Software Development Kit
SID	Source-to-Image Receptor Distance
SOP	Service Object Pair
TD	Distal-third
UD	Ultra-distal
UDI	Unique Device Identifier
UK	United Kingdom
WHO	World Health Organisation
WSL2	Windows Subsystem for Linux version 2

1 Description of the Device

The integration manual describes the installation and initial configuration of IBEX BH.

For the purpose of these instructions, the device consists exclusively of the Trueview Server and the underlying Trueview Software Development Kit (SDK). Instructions on how to set up the Trueview Server can be found in Section 3.2 for sandboxed installations, and Section 3.3 for native installations.

1.1 Software

The software is an automated processing server. It is configured to accept input images for processing from an X-ray modality, automatically process the image data using the Trueview SDK, and send an output to a configured destination (X-ray modality or PACS).

The Trueview SDK offers functionalities to improve, enhance, and exploit data from X-ray imaging. These functionalities are offered in multiple "specialisms". Currently, only the following specialism is available for use through the Trueview Server:

- Bone Health for Wrist.

1.2 Sandboxed Installation (Docker)

For sandboxed installations, the Trueview Server is provided as a Docker image. This image can be used to instantiate isolated containers, which offer several interfaces listed in Section 1.4. See Section 3.2.4 for more information on how to run the image.

For this type of installation, the Trueview Server contains the following items:

- The Trueview Server Production Image.
- The Trueview Server Mockup Image (for testing and development only).

1.3 Native Installation

For native installations (not sandboxed), the Trueview Server is provided as a standalone application, which is meant to be run as a background service. When started, this application offers several interfaces listed in Section 1.4. See Section 3.3.3 for more information on how to run the application.

For this type of installation, the Trueview Server contains the following items:

- The Trueview Server Production application.
- The Trueview Server Mockup application (for testing and development only).

1.4 Interfaces

1.4.1 DICOM API

The Trueview Server DICOM API provides programmatic access to the Trueview SDK functionalities. The DICOM API implements the C-STORE SCP and C-STORE SCU, so the server is able to receive input data and send output data. See Section 3.4 for more information on how to access and interact with the DICOM API. The full DICOM API is also documented in the DICOM conformance statement (`IBX_SWD_PRS_001`).

1.4.2 Configuration Interface

The Trueview Server configuration interface provides access to the Trueview Server configuration.

It is composed of a programmatic HTTP API, which implements the GET, PUT, and PATCH commands for querying and editing the current configuration. See Section 3.5.1 for more information on how to access

and interact with the HTTP configuration API. The full HTTP API is also documented in the technical specification (IBX_SWD_PRS_006).

An HTML configuration page is also served, providing a lightweight user interface for the HTTP configuration API. See Section 3.5.2 for more information on how to access and interact with the HTML configuration page. Higher-level instructions for field engineers are also provided in the service manual (IBX_PRD_SOP_003).

The Mockup Image also includes a separate mockup configuration interface, described in Section 3.6. The mockup interface does not have a dedicated web page.

2 Accuracy & Calibration

In the Bone Health specialism, the software runs an outlier detection algorithm that outputs an "outlier" flag. This flag is raised whenever a given data set (inputs and outputs) deviates significantly from expectations. In such cases and for elective scans, the operator should be advised to double-check their inputs, as well as review intermediate outputs of the software (such as the automatically detected ROIs). This flag is raised as Warning 202 "Similarity check failed" (for more information on warnings and errors, see the service manual; IBX_PRD_SOP_003).

Otherwise, no measurement of the accuracy of the outputs is made by the software. The end user operator is responsible for verifying the quality of the outputs, and for determining if an output is of diagnostic quality.

Calibration must be performed prior to deploying the Trueview Server. The calibration process is described in the service manual (IBX_PRD_SOP_003) and produces calibration databases for the respective imaging protocols. These databases enable the Trueview scatter-correction and bone health functionalities for radiographs obtained with these protocols.

Each database forms a "calibration dataset". It is specific to a particular configuration of the X-ray system and a particular imaging protocol. Calibration data are produced by IBEX, and are delivered as a ZIP archive containing the calibration for a selected number of system configurations and protocols. This ZIP archive may include only one configuration, or a series of configurations suitable for a given family of systems; this can be decided on a case-by-case basis when requesting the package.

In all cases, the Trueview Server will automatically select the appropriate calibration for each input image (or reject the input if no suitable calibration is available). This is achieved by reading values of standard DICOM attributes in the input DICOM files sent for processing. Since standard DICOM attributes are not always sufficient to unambiguously identify the system configuration, extra inputs may need to be provided to the Trueview Server to select the appropriate calibration. These inputs are called "system properties". They are set globally for the system and are assumed to apply equally to all input images sent to the Trueview Server. Please refer to Section 3.8 for managing calibrations and system properties.

Prior to a calibration being used for processing clinical data, it must pass an automated Quality Assurance (QA) test. This test requires an X-ray exposure of a standard phantom, and guarantees that the calibration is suitable for the X-ray system. More information on this procedure is available in the service manual (IBX_PRD_SOP_003).

Periodic re-calibration or correction of the system may be required to compensate for changes in the X-ray system. To flag protocols that may require a re-calibration, the software will start raising Warning 205 "Protocol calibration is out of date" after a specific number of days since the last run of the QA procedure. The corrective action frequency is dependent on the stability of the X-ray system, so service intervals cannot be specified here, and this time interval is configurable (defaults to one year). If a calibration has expired (or is about to expire), a successful run of QA testing will dismiss the warning and restart the expiration timer.

3 How to Use the Device

The instructions in this section apply equally to the Production and Mockup Images, unless otherwise specified.

3.1 Hardware Requirements

The Trueview Server is designed to run on hardware meeting or exceeding the following specifications. Operating the Trueview Server on hardware that does not meet these specifications is not supported.

3.1.1 For Sandboxed Installation (Docker)

Production Image:

- Operating system: Ubuntu 18.04 with Jetson Linux R32.7.3 (from Nvidia Jetpack 4.6.3).
- Container environment: Docker 23.0.3.
- Central/Graphics Processing Unit (CPU/GPU): Nvidia Jetson Xavier NX 8GB.
- Storage: 2GB free space (before installation).

Mockup Image:

- Operating system: Windows 10 64bit Professional version 22H2 with Windows Subsystem for Linux version 2 (WSL2) running Ubuntu 22.04.
- Container environment: Docker 23.0.3 (using WSL2 back-end).
- CPU: Intel i5 multi-core, or equivalent from other manufacturers.
- Random Access Memory (RAM): 1GB.
- Storage: 1GB free space (before installation).
- GPU: No GPU required.

The Docker Engine must be installed on any computer running the Trueview Server image. See the Docker documentation: `https://docs.docker.com/engine/install/`.

3.1.2 For Native Installation

Production and Mockup applications:

- Operating system: Windows 10 64bit Professional version 22H2.
- CPU: Intel i5 multi-core, or equivalent from other manufacturers.
- Random Access Memory (RAM): 32GB.
- Storage: 4GB free space (before installation).
- GPU: Nvidia GeForce GTX 1070 8GB (not required for Mockup application).

3.2 Sandboxed Installation of the Trueview Server

Instructions in this section are applicable only to sandboxed installations, using Docker.

3.2.1 Loading a Docker Image

The Trueview Server is provided as an image in .tar format. This image can be loaded into a local Docker image repository with the following command (replace `<image.tar>` by the actual path to the delivered image .tar file, e.g. `trueview-server-production-arm64-x.y.z.w.tar`):

```
docker load -i <image.tar>
```

This command only needs to be executed once on each computer that needs to run the server. Once completed, the image is installed on the system. In what follows, the image is identified by its tag. Trueview Server images have the following tags (where `x.y.z.w` is replaced by the actual release version number):

- Production Image: `trueview-server-production-arm64:x.y.z.w`
- Mockup Image: `trueview-server-mockup-x64:x.y.z.w`

To check that the image has been installed successfully, run the command `docker images`. The displayed list of images should contain one of the tags listed above, depending on which image was installed.

Please consult the Docker documentation for more in-depth information on managing Docker images.

3.2.2 Creating a Persistent Storage Volume

The Trueview Server requires dedicated persistent storage for its operation. This storage is used to store the server configuration and the calibration data. It can be created with the following command:

```
docker volume create trueview
```

This command only needs to be executed once on each computer that needs to run the server. To check that the volume has been created successfully, run the command `docker volume ls`. The displayed list of volumes should contain `trueview`.

Please consult the Docker documentation for more in-depth information on managing Docker volumes.

3.2.3 Initialising the Persistent Storage Volume

The volume created in Section 3.2.2 is empty. Before the Trueview Server can begin processing, the volume must be initialised. This includes creating the default configuration, as well as setting up the initial admin password required to access and edit the configuration (see Sections 3.5 and 3.5.2). This is done by running the Trueview Server image as root, with the following command (replacing `<image>` by the tag of the image to run, and `<passwd>` by a secure password):

```
docker run -u=0 --env CONF_PASSWORD=<passwd> -v trueview:/mnt/host <image>
```

This command will only initialise the volume, then return.

The password must contain between 1 and 43 ASCII characters. No other constraint is imposed on the password, which may contain any combination of ASCII characters within these limits. It is nonetheless recommended to follow state-of-the-art good practice on creating strong passwords, e.g., using a combination of at least 8 letters, digits, and symbols. In addition, it is recommended to avoid storing this password in an easily-accessible location (such as in a plain text file on the device), and to only share the password with the personnel that requires it.

3.2.4 Running the Trueview Server

Once the image has been installed as described in previous sections, the Trueview Server can be run with the following command (replacing `<image>` by the tag of the image to run):

Production Image:

```
docker run -p 2534:2534 -p 8080:8080 -v trueview:/mnt/host --runtime nvidia --gpus all <image>
```

Mockup Image:

```
docker run -p 2534:2534 -p 8080:8080 -v trueview:/mnt/host <image>
```

This command starts a new container based on the chosen image, and immediately launches the Trueview Server.

The option `-p 2534:2534` defines the networking port to open for the DICOM API (see Section 3.4). The option `-p 8080:8080` defines the networking port to open for the HTTP API (see Section 3.5.1). In both cases, the option is of the form `-p <outer_port>:<inner_port>`; the `<inner_port>` is the port that is configured internally in the Trueview Server image and cannot be changed, while `<outer_port>` can be changed at will (e.g., to avoid conflicts with another application running on the host system). These options must be specified even if the outer and inner ports are identical.

The option `-v trueview:/mnt/host` enables the use of the required persistent storage. The options `--runtime nvidia --gpus all` enable GPU support (only necessary for the Production Image).

To shut down the server, simply instruct Docker to stop the container; there is no specific shutdown sequence. Please consult the Docker documentation for more in-depth information on managing Docker containers.

3.3 Native Installation of the Trueview Server

Instructions in this section are applicable only to native installations, without a sandbox (e.g., no Docker).

3.3.1 Installing the Software

The Trueview Server is provided as compressed archive in .zip format. This archive can be unpacked (unzipped) into a chosen installation directory using the Windows Explorer. The recommended installation directory is:

```
C:\Program Files\IBEX\Trueview Server
```

If unpacking is successful, the installation directory contains the following folders:

- `bin`
- `share`

It is recommended to create a dedicated Windows user with lowest possible privileges, which will be used to run the software. This is outside the scope of these instructions for use; please refer to the Windows documentation for more information.

It is also recommended to protect the installed files against modification (accidental or malicious). This can be achieved by marking the whole installation directory as read-only, and by changing its owner to a privileged user (e.g., administrator).

By default, the software will use the following directories for storing configuration and data (where `%LOCALAPPDATA%` is the local application data for the user running the software, i.e., `%USERPROFILE%\AppData\Local` for regular users):

- `%LOCALAPPDATA%\trueview-server` for storing configuration, logs, and uploaded calibrations,
- `%LOCALAPPDATA%\Temp\trueview-server` for storing temporary files.

If these directories are not suitable, alternative paths can be chosen by creating a `paths.json` configuration file, using the JSON format, indicating the desired paths. The format of the file is the following (the paths indicated here are for illustration only):

```
{
  "persistent_storage_dir": "C:\\IBEX\\Storage",
  "temporary_dir": "C:\\IBEX\\Temp"
}
```

This configuration file must then be provided to the command-line interface of the application, using the additional argument

```
--paths <paths.json>
```

(replacing `<paths.json>` with the actual path to the file).

Finally, the software requires the Microsoft Visual C++ Redistributable runtime to be installed. If not already installed on the system, this can be obtained from Microsoft at the following address: `https://aka.ms/vs/17/release/vc_redist.x64.exe`. Please follow instructions provided with the Microsoft installer to install this prior to using the software.

3.3.2 Initialising the Configuration

Before the Trueview Server can begin processing, the local data must be initialised. This includes creating the default configuration, as well as setting up the initial admin password required to access and edit the configuration (see Sections 3.5 and 3.5.2). This is done by running the Trueview Server application with the following command (replacing `<install-dir>` by the selected installation directory):

```
<install-dir>\bin\trueview_server.exe --init-only --reset-password --reset-local-data
```

This command will only initialise the local data, prompt for a new password (press Enter after the password is entered to submit the new password), then return.

The password must contain between 1 and 43 ASCII characters. No other constraint is imposed on the password, which may contain any combination of ASCII characters within these limits. It is nonetheless recommended to follow state-of-the-art good practice on creating strong passwords, e.g., using a combination of at least 8 letters, digits, and symbols. In addition, it is recommended to avoid storing this password in an easily-accessible location (such as in a plain text file on the device), and to only share the password with the personnel that requires it.

3.3.3 Running the Trueview Server

Once the application has been installed as described in previous sections, the Trueview Server can be run

with the following command (replacing `<install-dir>` by the selected installation directory):

```
<install-dir>\bin\trueview_server.exe
```

To shut down the server, send a SIGINT, SIGTERM, or SIGBREAK signal to the application.

3.4 Using the DICOM API

The DICOM API is used to transfer the inputs and outputs of the Trueview Server. It is composed of a C-STORE SCP for receiving inputs, and a C-STORE SCU for sending outputs. This API is intended for use by the integrator to embed the Trueview Server in their workflow. It is not intended for use directly by end users. The connections do not support secure DICOM transfers. Therefore, the connections to the DICOM interface are not secure, and must not be established over an untrusted network.

Both SCP and SCU components are configured with default values specified in the following sections. Some elements of this configuration can be changed using the configuration interface, see Section 3.5.

3.4.1 Inputs

The C-STORE SCP receiving the inputs has the following default configuration:

- Receiver AET: `TrueviewReceiver`
- Receiver port: `2534`
- Calling AET filter: `<empty>`
- Input queue size: `20`

This SCP will only accept DICOM objects of the following Service Object Pair (SOP) classes, all with transfer syntax “Little Endian Explicit” or “Little Endian Implicit”:

- Verification (for testing and C-Echo commands)
- Digital X-Ray Image Storage - For Processing (for raw images)
- Digital X-Ray Image Storage - For Presentation (for OEM-processed images)

For processing to start, the server requires a pair of images: one OEM-processed image, and the corresponding raw unprocessed image. The two images must be sent within the same DICOM association. If one of the two images is missing, the input is discarded[1]. The input images must be acquired with one of the supported and calibrated protocols (see Section 2).

More detail about the expected DICOM tags and pixel data format can be found in the DICOM conformance statement (`IBX_SWD_PRS_001`) and the technical specification (`IBX_SWD_PRS_006`).

3.4.2 Outputs

The C-STORE SCU sending the outputs has the following default configuration:

- Sender AET: `TrueviewSender`
- Destination AET: `HostReceiver`
- Destination host name or IP: `127.0.0.1`
- Destination port: `2535`

[1] If desired, the software can be configured to accept inputs only containing the raw unprocessed image, but this is not enabled by default. This mode is meant mostly for experimentation and early integrations.

Note: In sandboxed integrations, since Docker containers cannot access their host through `127.0.0.1`, the destination IP must be changed to a valid IP address other than `127.0.0.1`, even if the receiver is running on the same host as the Trueview Server.

This SCU will only send DICOM objects of the following SOP class, with transfer syntax "Little Endian Explicit":

- Secondary Capture Image Storage

The machine-readable output of the Trueview Server is stored in the private DICOM tag `(0019,1001)`, in JavaScript Object Notation (JSON) format.

More detail about the produced DICOM tags and pixel data format can be found in the DICOM conformance statement (`IBX_SWD_PRS_001`) and the technical specification (`IBX_SWD_PRS_006`).

3.5 Using the Configuration Interface

In this section, `<ip>` refers to the IP address or host name of the computer hosting the Trueview Server, and `<port>` refers to the HTTP server port (i.e., 8080 or, for sandboxed installations, the outer HTTP port as configured in Section 3.2.4). Any connection must be established using the `http://` protocol, and not the `https://` protocol. Therefore, the connections to the configuration interface are not secure, and must not be established over an untrusted network.

The configuration interface is password-protected using basic HTTP authentication. The default admin username and password are:

- Username: `TrueviewConfig`
- Password: (set on installation, see Section 3.2.3 for sandboxed installations, or Section 3.3.2 for native installations)

The admin username can be changed, either on installation or at any time hereafter with the HTTP configuration interface API (see Section 3.5.1). The default password is initialised when the Trueview Server is installed (see Section 3.2.3 for sandboxed installations, or Section 3.3.2 for native installations). It can be changed at any time using the HTTP configuration API.

The admin user has full access to all components of the configuration interface. If necessary, additional users can be created with more limited access rights, using the HTTP configuration API. For details, see the technical specification (`IBX_SWD_PRS_006`).

3.5.1 Using the HTTP Configuration API

The password used to access the configuration interface can be changed for the currently authenticated user by sending an HTTP PUT request at the following address, with the new password as body (in plain text):

```
http://<ip>:<port>/api/v1/conf/passwd
```

Note that this PUT request requires authentication with the *old* password, not the new password. If the old password is lost and cannot be recovered, simply delete the volume or reset the local data (see Section 3.10) and follow once more the instructions in Section 3.2.3 for sandboxed installations or Section 3.3.2 for native installations to re-generate the default configuration. Note that this will also delete any uploaded calibration data (Section 3.8); these will have to be re-uploaded and re-validated before the system can be used again.

The configuration itself is accessible at the following HTTP address:

```
http://<ip>:<port>/api/v1/conf
```

The current configuration can be queried with an HTTP GET request. It can also be updated with an HTTP PUT or PATCH request at the same address, with the modified JSON configuration as payload. If using a PATCH request, the JSON payload may contain only the elements of the configuration that need updating; any element not included in the payload will remain unchanged in the current configuration.

When the configuration is changed, the Trueview Server will restart some of its internal services. Therefore, the server will not be able to respond to requests immediately.

The default configuration is:


```
{
  "receiver": {
    "aet": "TrueviewReceiver",
    "calling_aet_filter": "",
    "queue_size": 20,
    "port": 2534
  },
  "sender": {
    "aet": "TrueviewSender",
    "dest_aet": "HostReceiver",
    "dest_host": "127.0.0.1",
    "dest_port": 2535,
    "retry_delay_seconds": 5
  },
  "licensing": {
    "use_agent": true,
    "agent_host": "127.0.0.1",
    "agent_port": 7777
  },
  "processing": {
    "calibration_qa_mode": false,
    "qa_expiry_days": 365,
    "kvp_offset_measured_from_nominal": 0.0,
    "tscore_threshold": -2.5,
    "normative_data": {
      "UDR": {
        "male": {"mean": 0.42025, "std": 0.046735},
        "female": {"mean": 0.36676, "std": 0.041923}
      },
      "TDR": {
        "male": {"mean": 0.78715, "std": 0.070091},
        "female": {"mean": 0.69539, "std": 0.065608}
      }
    }
  },
  "report": {
    "series_description": "IBEX Bone Health report",
    "series_number_policy": "increment_source",
    "series_number": 1,
    "require_presentation_dataset": true,
    "osteoporosis_title": "OSTEOPOROSIS IDENTIFIED",
    "osteoporosis_desc": "Osteoporosis detected at wrist.",
    "normal_title": "NORMAL",
    "normal_desc": "No osteoporosis detected.",
    "error_title": "NOT ASSESSED",
```

```
    "error_desc": "An error occurred. Osteoporosis not assessed.",
    "udr_region_title": "Region: UD",
    "tdr_region_title": "Region: TD",
    "abmd_label": "aBMD = {:.3f} g/cm²",
    "tscore_male_label": "T (Male) = {:.1f}",
    "tscore_female_label": "T (Female) = {:.1f}",
    "error_codes_title": "<font face=bold>Errors: {}</font>",
    "warning_codes_title": "<font face=bold>Warnings: {}</font>",
    "too_many_issues_reported_label": "<font face=bold>Too many issues reported. List truncated.</font>
    "version_label": "Version: {}",
    "script": "Latn",
    "draw_data_panel": true,
    "report_scale": 1.0,
    "report_overlay_scale": 1.0,
    "report_bone_edges_scale": 1.0,
    "report_brightness": 1.0,
    "xray_image_brightness": 0.8,
    "xray_image_rotation_step_in_degrees": 0,
    "xray_image_clip_to_collimator": true,
    "watermark_enabled": {
      "206": true,
      "211": true
    },
    "watermark_label": {
      "206": "Outside Indication for Use",
      "211": "Experimental Protocol"
    },
    "watermark_position": "bottom_right",
    "watermark_xray_only": true,
    "watermark_brightness": 0.85,
    "watermark_scale": 1.0
  },
  "report_qa": {
    "qa_passed_title": "QA PASS",
    "qa_passed_desc": "QA successful. System is operational.",
    "qa_failed_title": "QA FAIL",
    "qa_failed_desc": "An error occurred. QA failed."
  },
  "system": {
    "max_log_files": 10,
    "max_log_file_size_kb": 5120,
    "max_request_payload_size_kb": 1048576,
    "max_calibration_package_size_kb": 1048576,
    "pipeline_retention_days": 30,
    "data_encryption_scheme": "aes256_v1",
    "retain_datasets": false
  }
}
```


The meaning of each setting is as follows:

- `receiver.aet`: AET of the DICOM C-STORE SCP of the Trueview Server.
- `receiver.calling_aet_filter`: AET of the client(s) allowed to send C-STORE and C-ECHO commands

to the DICOM SCP of Trueview Server. If empty, all clients are allowed. Multiple AETs can be specified, separated by a newline character (`\n`).

- `receiver.queue_size`: Maximum number of input data sets queued for processing. Each pair of "ForProcessing" and "ForPresentation" DICOMs counts as one data set. If this number is reached, any further C-STORE command will be rejected.
- `receiver.port`: Network port of the DICOM C-STORE SCP of the Trueview Server. This is only meant to be changed in native installations; for sandboxed installations, use Docker port mapping instead.
- `sender.aet`: AET of the DICOM C-STORE SCU of Trueview Server.
- `sender.dest_aet`: AET of the destination DICOM C-STORE SCP, where output DICOMs are sent.
- `sender.dest_host`: Hostname or IP address of the destination DICOM C-STORE SCP.
- `sender.dest_port`: Network port of the destination DICOM C-STORE SCP.
- `sender.retry_delay_seconds`: Delay (in seconds) before re-trying sending the output DICOM, in case the first sending attempt failed.
- `licensing.use_agent`: A toggle to switch between agent-based licensing and standalone licensing; see Section 3.7.
- `licensing.agent_host`: Hostname or IP address of the license agent; see Section 3.7.
- `licensing.agent_port`: Network port of the license agent; see Section 3.7.
- `processing.calibration_qa_mode`: Whether the system is in QA mode, accepting only QA phantoms for calibration QA purpose; see the service manual (`IBX_PRD_SOP_003`).
- `processing.qa_expiry_days`: Number of days for which a calibration QA result is considered valid. After this time period, a warning will be raised prompting a rerun of the QA procedure.
- `processing.kvp_offset_measured_from_nominal`: Additive offset to apply to the nominal KVP reported in the input DICOM, in order to obtain a value closer to the real KVP output of the system (computed as `measured - nominal`). Note: this setting has been retired from the configuration web page, but is still accessible through the API. Its use is not recommended.
- `processing.tscore_threshold`: Threshold below which a given T-score will be considered as indicative of osteoporosis.
- `processing.normative_data.UDR`: Normative data for the UD region. The normative data must be provided for males and females, and must include the mean aBMD and standard deviation of aBMD for the respective target population.
- `processing.normative_data.TDR`: Normative data for the TD region, see above for description of the expected fields.
- `report.series_description`: Value to use for the 'Series Description' tag in the output secondary capture.
- `report.series_number_policy`: Method to use for generating the 'Series Number' tag in the output secondary capture: `"increment_source"` will increment the value of this tag read from the source DICOM, `"fixed"` will use a fixed number (specified below).
- `report.series_number`: The fixed 'Series Number' to use, if the policy is set to `"fixed"` (unused otherwise).
- `report.require_presentation_dataset`: Whether a 'For Presentation' dataset is required for creating the report (`true`) or whether the 'For Processing' dataset can be used as a fallback (`false`).

- `report.osteoporosis_title`: Heading of the graphical DICOM report when osteoporosis has been detected.
- `report.osteoporosis_desc`: Paragraph displayed in the graphical DICOM report when osteoporosis has been detected.
- `report.normal_title`: Heading of the graphical DICOM report when osteoporosis has not been detected.
- `report.normal_desc`: Paragraph displayed in the graphical DICOM report when osteoporosis has not been detected.
- `report.error_title`: Heading of the graphical DICOM report when an error occurred and osteoporosis could not be assessed.
- `report.error_desc`: Paragraph displayed in the graphical DICOM report when an error occurred and osteoporosis could not be assessed.
- `report.draw_data_panel`: Whether to draw the data panels on the graphical report (true) or not (false). Note: the data panel is the panel which displays text boxes, including in particular the numerical values for aBMD and T-score.
- `report.udr_region_title`: Header of the UD region box.
- `report.tdr_region_title`: Header of the TD region box.
- `report.abmd_label`: Label for reported aBMD value in each region box. The aBMD value is inserted at the location of the `{...}` symbol, and is formatted following the format flags inside the curly braces. See below for information on format flags.
- `report.tscore_male_label`: Label for reported male T-score in each region box. The T-score value is inserted at the location of the `{...}` symbol, and is formatted following the format flags inside the curly braces. See below for information on format flags.
- `report.tscore_female_label`: Label for reported female T-score in each region box. The T-score value is inserted at the location of the `{...}` symbol, and is formatted following the format flags inside the curly braces. See below for information on format flags.
- `report.error_codes_title`: Label for reported errors.
- `report.warning_codes_title`: Label for reported warnings.
- `report.too_many_issues_reported_label`: Placeholder to display when too many issues (errors or warnings) are reported.
- `report.version_label`: Label for product version number.
- `report.script`: The default font script to use when rendering text in the graphical report. The script affects the selected font file and rendering of characters, and must be set appropriately to correctly display text in non-Latin languages.
- `report.report_scale`: The scale factor of the report elements, such as text boxes and logos (e.g., 1.0: standard size, 2.0: twice standard, 0.5: half standard).
- `report.report_overlay_scale`: The scale factor of overlay elements, such as arrows and ROI borders (e.g., 1.0: standard size, 2.0: twice standard, 0.5: half standard).
- `report.report_bone_edges_scale`: The scale factor of bone edges drawn inside ROIs (e.g., 1.0: standard size, 2.0: twice standard, 0.5: half standard).
- `report.report_brightness`: Maximum brightness level (0: black, 1: white) for the report text and boxes.

- `report.xray_image_brightness`: Maximum brightness level (0: black, 1: white) for the background x-ray image.
- `report.xray_image_rotation_step_in_degrees`: The step size for the automatic X-ray image rotation, in degrees (a step size of 0 allows all rotations, while a step size of 90 degrees will only rotate the image by multiples of 90 degrees; a step size of 360 effectively disables automatic rotation).
- `report.xray_image_clip_to_collimator`: Whether to automatically clip the X-ray image to the detected collimated area (`true`) or not (`false`).
- `report.watermark_enabled.206`: Whether a watermark will be generated on the output image if the warning 206 "Bone health assessment not indicated" is raised.
- `report.watermark_enabled.211`: Whether a watermark will be generated on the output image if the warning 211 "Experimental protocol" is raised.
- `report.watermark_label.206`: Text to display on the watermark when warning 206 is raised.
- `report.watermark_label.211`: Text to display on the watermark when warning 211 is raised.
- `report.watermark_position`: Position of the watermark on the output image. Possible values are: `"top_left_repeat"`, `"top_left"`, `"top_right"`, `"bottom_left"`, `"bottom_right"`, `"center"`.
- `report.watermark_xray_only`: Whether the watermark is displayed on the X-ray image only, or on the whole output image.
- `report.watermark_brightness`: Maximum brightness level (0: black, 1: white) for the watermark text.
- `report.watermark_scale`: The scale factor of the watermark text (e.g., 1.0: standard size, 2.0: twice standard, 0.5: half standard).
- `report_qa.qa_passed_title`: Heading of the graphical DICOM report for QA exposures when QA has passed.
- `report_qa.qa_passed_desc`: Paragraph displayed in the graphical DICOM report for QA exposures when QA has passed.
- `report_qa.qa_failed_title`: Heading of the graphical DICOM report for QA exposures when QA has failed.
- `report_qa.qa_failed_desc`: Paragraph displayed in the graphical DICOM report for QA exposures when QA has failed.
- `system.max_log_files`: Maximum number of log files to create; beyond this limit, the oldest log files will be deleted.
- `system.max_log_file_size_kb`: Maximum size (in KB) of a log file before the logs rotate to a new file.
- `system.max_request_payload_size_kb`: Maximum size (in KB) of an HTTP request sent to the software (longer requests will be denied).
- `system.max_calibration_package_size_kb`: Maximum size (in KB) of a calibration package after unzipping. Note: See also `system.max_request_payload_size_kb` which effectively places a limit on the maximum size of a calibration package, but *before* unzipping.
- `system.pipeline_retention_days`: Maximum number of days for which to keep entries in the pipeline database; older entries will be automatically cleaned up.
- `system.data_encryption_scheme`: Encryption method to use when storing input data. Possible values: `"none"` (no encryption), `"aes256_v1"` (AES encryption with 256 bit key).

- `system.retain_datasets`: Whether to retain received and sent DICOMs for troubleshooting purposes (`true`) or not (`false`). Note: when this option is enabled, the DICOMs are retained until their corresponding entry is removed from the pipeline database (see `system.pipeline_retention_days`) and are stored with the encryption scheme defined in `system.data_encryption_scheme`.

For settings making use of format flags, please refer to the format string syntax documentation available at the following address: `https://fmt.dev/10.1.0/syntax.html`. All settings controlling text displayed on the output image can make use of formatting tags (e.g., `<font face=bold>...</font>`; see the technical specification for details (`IBX_SWD_PRS_006`).

3.5.2 Using the HTML Configuration Page

The HTML configuration page is accessible with a web browser at the following HTTP address:

`http://<ip>:<port>/conf`

Please refer to the service manual (`IBX_PRD_SOP_003`) for instructions on how to use this web page.

3.6 Using the Mockup Configuration Interface

This interface is only available in the Mockup image/application, and is intended to be used only for development and testing. It can be accessed over HTTP with the same IP and port as described in Section 3.5, but with no authentication. The mockup configuration is accessible at the following HTTP address:

`http://<ip>:<port>/api/v1/mockup`

The current configuration can be queried with an HTTP GET request. It can also be updated with an HTTP PUT or PATCH request at the same address, with the modified JSON configuration as payload. If using a PATCH request, the JSON payload may contain only the elements of the configuration that need updating; any element not included in the payload will remain unchanged in the current configuration. When the mockup configuration is changed, the Trueview Server will restart some of its internal services. Therefore, the server will not be able to respond to requests immediately.

The default configuration is:

```
{
  "receiver_listening": true,
  "output_delay_seconds": 5,
  "errors": [],
  "output": {
    "outcome": "default",
    "calibration_qa_success": true,
    "osteoporosis": false,
    "detected_rois": ["UDR", "TDR"]
  }
}
```

The meaning of each setting is as follows:

- `receiver_listening`: If set to false, the DICOM C-STORE SCP will not listen to incoming requests.
- `output_delay_seconds`: Specifies the delay (in seconds) for a mock output DICOM to be created and sent back.
- `errors`: List of error codes to generate (not implemented in this release).

- `output.outcome`: If set to "default", the mockup will behave as close to the production setup as possible, such that the processing will stop and report failure whenever an error occurs, and report success otherwise. If set to "success", the output will always report success, even if an error occurred. If set to "failure", the output will always report failure, even if no error occurred.
- `output.calibration_qa_success`: If set to true, the mockup calibration QA pipeline will report a "pass", otherwise it will report a "fail".
- `output.osteoporosis`: If set to true, the mock output DICOM will contain data indicative of osteoporosis. Otherwise, the data will be indicative of a normal bone health.
- `output.detected_rois`: List of ROIs that are detected in the mock output. This may be set only to `["UDR","TDR"]`, `["UDR"]`, or `[]`.

3.7 Licensing

The Trueview Server requires a license of IBEX BH.

When running inside a sandboxed environment (such as Docker or a virtual machine), this license is managed by a "license agent", which is a companion application running on the native operating system. Before, during, and after processing, the Trueview Server will contact the license agent to check if a valid license is configured and available. If so, the license will be "borrowed" by the Trueview Server, and will become unavailable to other processes. Otherwise, if no license is available, processing stops. Therefore, the license agent application must be launched and remain alive while the Trueview Server services are needed.

For native installations, the license agent is not necessary; a simpler "standalone" licensing (without a license agent) must be used instead. In this mode, the functionalities of the license agent are integrated directly into the Trueview Server application, and there is no need to manage a separate agent process. A setting is provided to select the licensing method, see Section 3.5.1.

The license agent (or, in native installations, the main application) manages the "base" IBEX license only. "Additional" license restrictions can be configured on the Trueview Server. Please see the service manual (`IBX_PRD_SOP_003`) for more information.

Note: The Mockup Image does not require a license.

3.7.1 Installing the License Agent

This is only necessary for sandboxed installations. The license agent application consists of a standalone executable file, which can be downloaded from a running instance of the Trueview Server at the address:

`http://<ip>:<port>/api/v1/licensing/agent`

(replacing `<ip>` and `<port>` by the values described in Section 3.5.1). This executable file must be downloaded into any folder on the host file system. A copy of this executable is included alongside the release package, if downloading from the container is not a viable option.

3.7.2 Starting the License Agent

This is only necessary for sandboxed installations. The license agent is a command-line application and has no user interface. It can be configured using the following command-line arguments:

- `--port <number>`: Specify the networking port to use to listen to requests from the Trueview Server. The default is `7777`. This may be changed if this port is already in use. If so, please make sure to update the agent port in the Trueview Server configuration (see Section 3.5.1).

- Other options are available for debugging purposes; please contact IBEX for further details.

To reduce cybersecurity risks, the license agent must not be executed with elevated ("root") privileges. The license agent requires read and write access to the chosen installation folder and read access to the operating system folders accessible to non-root users.

The license agent must run natively on the host operating system. It cannot run from inside a container or a virtual machine. Only one instance of the license agent will be allowed to run on a given system. Only one instance of the Trueview Server can connect to the license agent.

Any attempt to circumvent these limitations will be detected, and the license agent will either stop or refuse to provide a valid license.

Once started, the license agent listens to requests until it is closed. It is therefore recommended to run the license agent as a background service, using the system administration tools of the host operating system. The license agent outputs logs in the current working directory.

3.7.3 Setting Up or Renewing the IBEX BH License

Please refer to the service manual (`IBX_PRD_SOP_003`) for information on how to activate or renew a license using the licensing web page.

3.7.4 Querying the Status of the License

Please refer to the service manual (`IBX_PRD_SOP_003`) for instruction on how to query the license status from the licensing web page. This information can also be accessed programmatically by making a `GET` request at the following address:

```
http://<ip>:<port>/api/v1/licensing/status
```

The corresponding response will contain the information encoded in a JSON object, for example:

```
{
  "agent_fingerprint": "IBEXDEV04-QaYrV3-mtZGw6-vHJFw4-gYFyD5",
  "fingerprint": "IBEXDEV06-xdFFe-masdw6-ffg5sf-s5sref",
  "time": "2023-12-06T11:00:40.329Z",
  "installed": true,
  "valid": true,
  "used": true,
  "last_renewal": "2023-12-05T15:04:47.120Z",
  "expiry": "2023-12-28T00:00:00.000Z",
  "first_use": "2023-12-01T01:05:07.000Z",
  "remaining_uses": 2,
  "max_uses": 20,
  "error": "..."
}
```

The meaning of each entry is as follows:

- `agent_fingerprint`: Present if using agent-based licensing only. The activation ID of the license agent. Necessary only for renewals.
- `fingerprint`: Present if using standalone licensing only. The activation ID of the device running the Trueview Server. Necessary only for renewals.
- `time`: The current time as measured on the Trueview Server.

- `installed`: Whether a license is installed (`true`) or not (`false`), regardless of whether the license is valid or not. E.g., an installed license may be expired.
- `valid`: Whether the installed license is valid (`true`) or not (`false`). A license is invalid if it goes past its configured expiry date, or runs out of uses, or is not installed.
- `used`: Whether the installed license is currently in use for processing (`true`) or not (`false`).
- `last_renewal`: The date of last renewal. Only present if `"installed"` is `true`.
- `expiry`: The expiry date of the license. `null` if the license does not expire (perpetual license). Only present if `"installed"` is `true`.
- `first_use` indicates the time at which the license was first used to process data. `null` if the license has not yet been used. Only present if `"installed"` is `true`.
- `remaining_uses`: The number of remaining uses of the license. `null` if there is no limit. Only present if `"installed"` is `true`.
- `max_uses`: The maximum number of uses allowed at the last renewal. `null` if there is no limit. Only present if `"installed"` is `true`.
- `error`: Present only if an error occurred when trying to establish the complete license status.

3.8 Using the Calibration Interface

As discussed in Section 2, the Trueview Server processing requires calibration on each deployed X-ray system. This is expected to be performed by a trained engineer, either at the factory where the system is assembled, or on site when the system is first deployed.

3.9 Using the Engineering Interface

The Engineering interface is a debugging interface, offered mostly for the benefit of the field service engineer and integrators. It uses the same HTTP/HTML protocol and authentication as the configuration interface (see the service manual; `IBX_PRD_SOP_003`).

3.9.1 Using the HTTP Pipeline API

The Trueview Server pipeline API can be accessed at the following addresses (using HTTP `GET` requests, with authentication; see Section 3.5 for more information on accessing the HTTP interface):

`http://<ip>:<port>/api/v1/pipeline/datasets` (for fetching the list of all data sets)

`http://<ip>:<port>/api/v1/pipeline/dataset/X/details` (replacing `X` with the unique ID of a given data set; for fetching individual data sets in more detail)

`http://<ip>:<port>/api/v1/pipeline/dataset/X/files` (replacing `X` with the unique ID of a given data set; for fetching received and generated files, only reliably usable when the `system.retain_datasets` option is enabled)

This API allows querying the status of the processing pipeline, listing data sets that were processed or are being processed, accessing the list of generated errors and warnings for each data set, and accessing underlying files. This interface is meant mostly to be used by the dashboard page (see Section 3.9.2), but it can also be used manually for debugging during integration. The details of this API are available in the technical specification (`IBX_SWD_PRS_006`).

The list of data sets included in this API is periodically cleaned up to remove the oldest entries, to avoid excessive build up of entries, and to comply with local data retention regulations. The retention duration is configurable, see Section 3.5.1. If a manual cleanup is required, the following API can be used to erase all records (using an HTTP `DELETE` request, with authentication):

`http://<ip>:<port>/api/v1/pipeline/datasets`

Finally, the following API can be used to flush the pipeline, i.e., to cancel processing of any data set queued or being processed (using an HTTP `POST` request, with authentication):

`http://<ip>:<port>/api/v1/pipeline/flush`

3.9.2 Using the HTML Dashboard Page

Please refer to the service manual (`IBX_PRD_SOP_003`) on how to access and use the dashboard web page to display the live pipeline status.

3.9.3 Using the HTTP Logs API

The Trueview Server log API can be accessed at the following addresses (using HTTP `GET` requests, with authentication; see Section 3.5 for more information on accessing the HTTP interface):

`http://<ip>:<port>/api/v1/logs` (for fetching the list of all log files and their unique IDs)

`http://<ip>:<port>/api/v1/logs/X` (replacing `X` with the unique ID of the file; for fetching individual log files)

These logs can be used for deeper debugging of issues, when the errors and warnings displayed on the dashboard are not sufficient to understand a problem. Note that the logs are periodically rotated, such that the oldest log entries are eventually deleted to make room for newer entries. This rotation is configurable, see Section 3.5.1.

3.9.4 Using the HTML Logs Page

Please refer to the service manual (`IBX_PRD_SOP_003`) on how to access and use the logs web page to display the current Trueview Server logs.

3.10 Update or Uninstall the Device

3.10.1 For Sandboxed Installation (Docker)

Trueview Server images can be updated by following the instructions in Section 3.2.1 to install the new image, then following the instructions below to uninstall the old image. When the container is restarted after the update, it will preserve any existing local data (configuration, password, etc.) and will automatically perform any required migration (addition of new settings, etc.).

To uninstall an image, run the following command (replacing `<image>` by the tag of the image to uninstall):

```
docker rmi -f <image>
```

If uninstalling the Trueview Server permanently (rather than updating to a new version), the persistent storage should be removed with the following command:

```
docker volume rm trueview
```

Please consult the Docker documentation for more in-depth information on managing Docker images and volumes.

3.10.2 For Native Installation

Updating or uninstalling the software must not be attempted while the application is in use.

Updates to the software are performed by unpacking (unzipping) the new archive onto a temporary location, and if the unpacking is successful (no error reported), copy the extracted files onto the original installation (see Section 3.3.1). When the application is restarted after the update, it will preserve any existing local data (configuration, password, etc.) and will automatically perform any required migration (addition of new settings, etc.).

Uninstalling the software is performed by removing the folder storing local data (see Section 3.3.1 for instructions on locating this folder), and removing the installation folder.

4 Risks

The Trueview Server should run on a computer (or virtual machine) that is independent and not responsible for operating the X-ray equipment. Network communications with the device (using either the DICOM or HTTP APIs) are not encrypted, and must operate on a trusted network. For native installations, the Trueview Server should run under an unprivileged user (i.e., not administrator or root), and the installed files should be protected against modification (see 3.3.1). If these instructions are followed, this device may be used safely without risk to patient, radiographer, and health care provider IT infrastructure.

No Personally Identifiable Information should be included in the DICOMs sent to the Trueview Server, beyond the data required for processing (e.g., patient age and sex). The Trueview Server will make no attempt at anonymizing the received DICOMs, hence it is the responsibility of the integrator to ensure that such data is not sent, in order to mitigate any risk of sensitive data being leaked or harvested by an attacker.

If the Trueview Server fails to process a given input for any reason, no usable output will be generated. If the failure is fatal and the Trueview Server is forced to terminate, no output will be generated. Otherwise, the Trueview Server will output a DICOM object describing the nature of the error. In all cases, the input data will be discarded and lost. It is the responsibility of the integrator to back up the original input data if necessary.

Finally, for the Bone Health specialism, the Trueview Server outputs an outlier flag (similarity check), which can be used to highlight potential issues with the processing or with the input. See Section 2 for more information.

5 Data Backup Warning

The input data provided to the Trueview Server will not be automatically backed up. It remains the responsibility of the integrator to ensure, when appropriate, that a copy of any input data is preserved in case hardware or software failure halts the processing.

Likewise, outputs created by the Trueview Server will not be automatically backed up. It remains the responsibility of the integrator to retrieve this data and save copies and backups as appropriate.

6 Accessing the Medical Device Label

The medical labels of all software devices used to implement the Trueview Server can be accessed with a web browser using the following address (see Section 3.5 for details on how to access the HTTP API):

```
http://<ip>:<port>/label
```

This does not require any authentication. An example label is shown in Figure 1, highlighted with the symbol meanings. These labels must be made available to the end user of the software, either in a printed form, or displayed in electronic format.

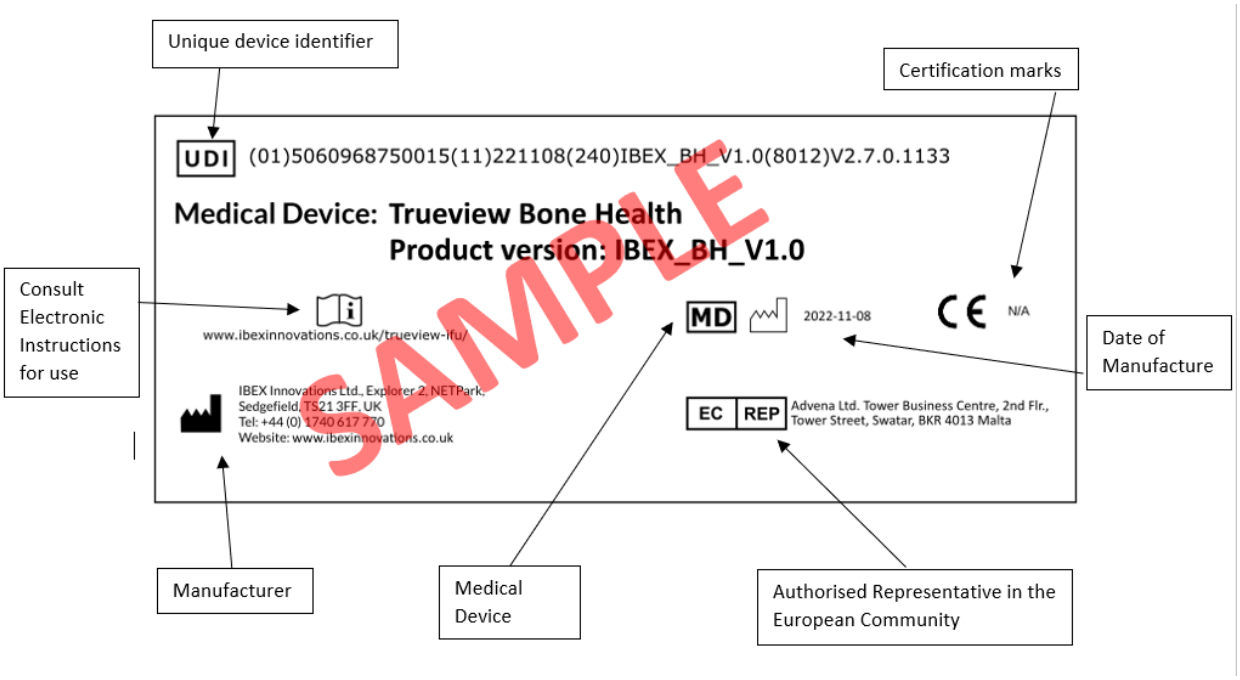

Figure 1: Sample Medical Device label.

7 Accessing the Unique Device Identifier

Each specialism included in the Trueview Server is assigned its own Unique Device Identifier (UDI). The UDI can be found on the respective label (see Figure 1) and can also be queried in a machine-readable format using the HTTP API at the following address (see Section 3.8 for details on how to access the HTTP API):

`http://<ip>:<port>/api/v1/status/udis`

The UDIs of all devices in use by the Trueview Server can be queried with an HTTP GET request at the address above, in JSON format. This does not require any authentication.

8 Appendix

8.1 Medical Device Label

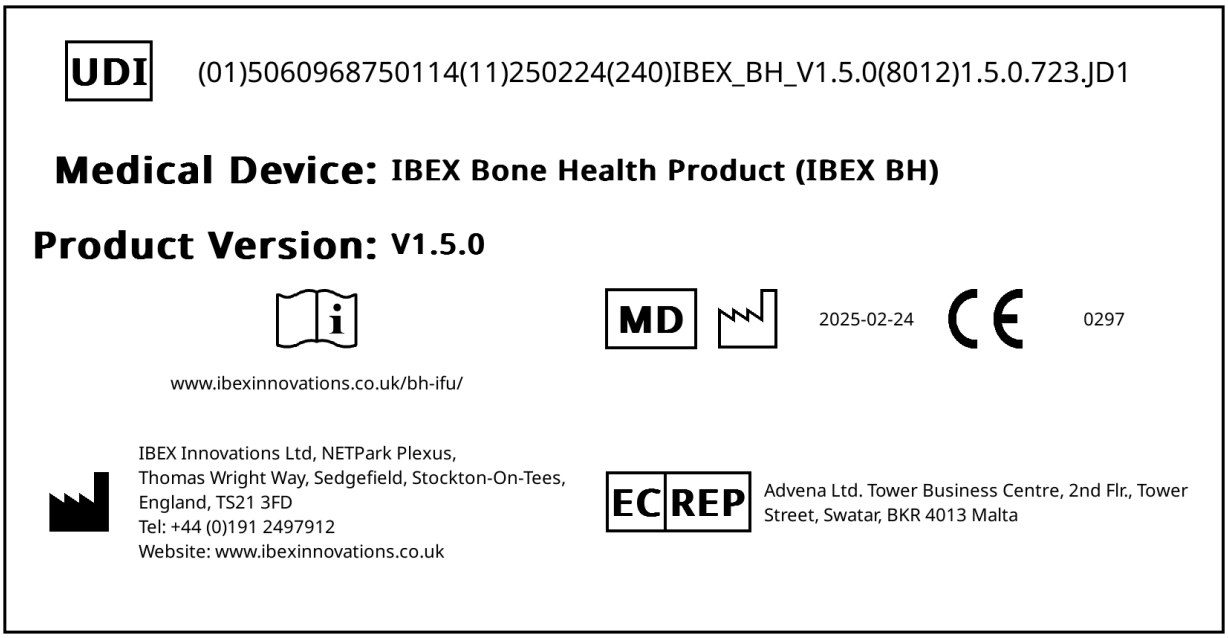

Figure 2: Product label for integration on Nvidia Jetson Xavier NX (Docker, Linux).

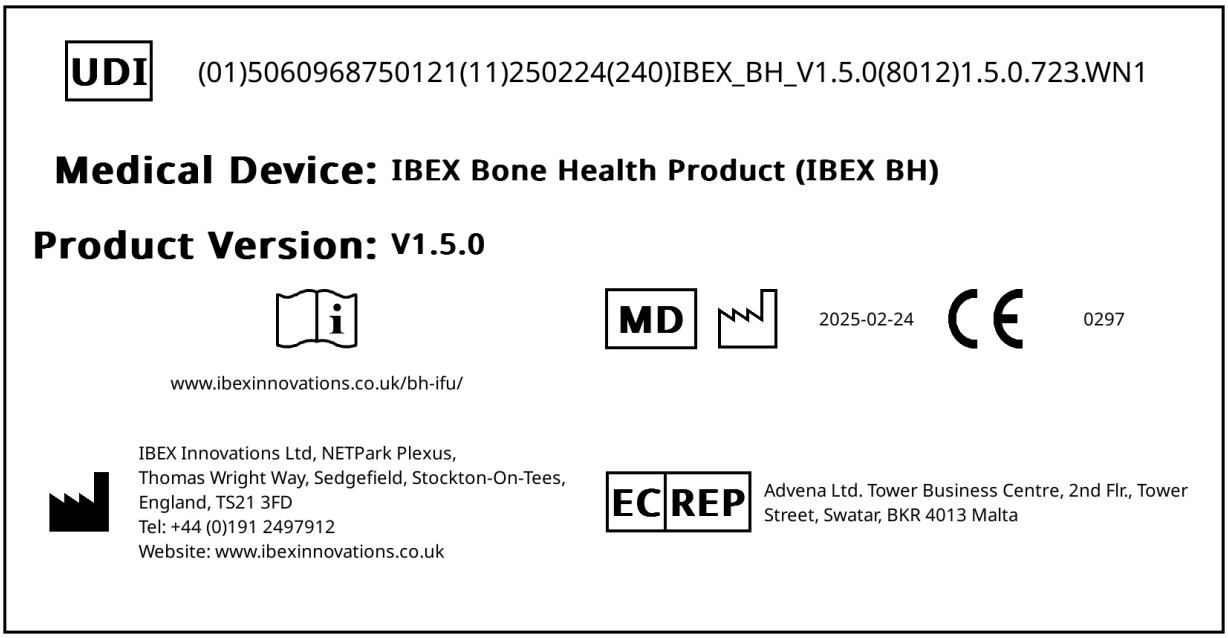

Figure 3: Product label for integration on generic desktop PC (Native, Windows).

8.2 Regulatory Clearance

IBEX BH is a CE marked class IIa according to Regulation (EU) 2017/745 (EU MDR) and approved for use in applicable jurisdictions.

- 2017/745 — Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 concerning Medical Devices.
- EN ISO 13485:2016+A11:2021 — Medical Devices — Quality Management Systems — Requirements for Regulatory Purposes.
- EN ISO 14971:2019+A11:2021 — Medical Devices — Application of Risk Management to Medical Devices.
- EN ISO 15223-1:2021 — Medical devices. Symbols to be used with information to be supplied by the manufacturer — General requirements.
- EN ISO 20417:2021 — Medical devices. Information to be supplied by the manufacturer.

8.3 Patents

The IBEX BH software is protected by following patents: JP6598087, US10588592, GB2576772, GB2608900, GB2611231, GB2589643, GB2563115, US 10820878 (Japan, South Korea Pending, EU, China Pending).

Instructions For Use - Integration Manual - IBEX BH v1.5.0

Final Audit Report 2025-03-03

Created:	2025-03-03
By:	Corentin Schreiber (c.schreiber@ibexinnovations.co.uk)
Status:	Signed
Transaction ID:	CBJCHBCAABAAqAwhiX4YBQzYln_g2REuIZD70wHqiA9d

"Instructions For Use - Integration Manual - IBEX BH v1.5.0" History

Document created by Corentin Schreiber (c.schreiber@ibexinnovations.co.uk)
2025-03-03 - 2:58:54 PM GMT

Document emailed to d.joyce@ibexinnovations.co.uk for signature
2025-03-03 - 3:00:15 PM GMT

Email viewed by d.joyce@ibexinnovations.co.uk
2025-03-03 - 3:03:28 PM GMT

Signer d.joyce@ibexinnovations.co.uk entered name at signing as David Joyce
2025-03-03 - 3:05:32 PM GMT

Document e-signed by David Joyce (d.joyce@ibexinnovations.co.uk)
Signature Date: 2025-03-03 - 3:05:34 PM GMT - Time Source: server

Document emailed to Paul Scott (p.scott@ibexinnovations.co.uk) for signature
2025-03-03 - 3:05:36 PM GMT

Email viewed by Paul Scott (p.scott@ibexinnovations.co.uk)
2025-03-03 - 3:12:57 PM GMT

Document e-signed by Paul Scott (p.scott@ibexinnovations.co.uk)
Signature Date: 2025-03-03 - 3:13:13 PM GMT - Time Source: server

Agreement completed.
2025-03-03 - 3:13:13 PM GMT