



Document Ref	RecIFU_TV
Revision	1.10

Instructions for Use:

Trueview 2.0.0 Software Development Kit

1. Description of device

For the purpose of these instructions the device consists of the Trueview software and its Software Development Kit (SDK). The Trueview SDK contains the following items:

- The Trueview software runtime libraries (DLLs) and their dependencies.
- The Trueview common assets required for running the Trueview software.
- The Trueview Application Programming Interface (API) headers for the C and C++ languages and the CLR assembly for the C# language.
- The Trueview code examples demonstrating the usage of the API in multiple languages (C++ and C#) and for all supported Trueview specialisms: Digital Radiography (DR), Mammography (Mammo), Bone Health (BH), and Cone Beam Computed Tomography (CBCT).
- The Trueview License Manager tool required for offline licensing of the software.
- The Trueview Demonstrator application, for the DR, BH, and CBCT specialisms only.

In what follows, we refer to the “development machine” as the computer on which the application using the Trueview software is built, and the “production machine” as the computer that will be running Trueview in production. Instructions on how to setup Trueview for both development and production environments can be found in sections 5.1 and 5.3.

1.1. Software

The Trueview software improves a standard diagnostic digital radiograph by removing the need for an anti-scatter grid, normally used during image acquisition, whilst enabling subsequent processing of the radiograph to produce a scatter-corrected image. In the BH specialism, the Trueview software also provides an indicator for bone health. These functionalities are provided by the file `Lib\IbexSDK.dll` and its dependencies, and `Lib\IbexSDK.NET.dll` for the CLR interface. The dependencies are all the other DLLs listed in the same directory. All these DLLs must be present on the machine for Trueview to operate. See section 5.1 for more information.

1.2. Common assets

The Trueview common assets are binary data required for the general operation of the Trueview software. This currently consists of the files in the `Lib\Segmentation` folder. Except for the Mammo specialism and for the CBCT specialism with forward projection support, this folder must be present on the machine for Trueview to operate, and its folder structure relative to the `IbexSDK.dll` file must not be altered (both in development and production). New assets may be added to this folder in future releases of the software.

1.3. API

The Trueview API provides programmatic access to the Trueview software functionalities. The `IbexSDK.NET.dll` file contains all the CLR assembly information required for programming against this API with the C# language. Other languages such as C or C++ should link directly to



`IbexSDK.dll` and require additional textual headers to access this API, which are listed in the `Include` folder. For the C language the header is named `IbexTrueview.h` and for the C++ language the header is named `IbexTrueviewCPP.h`. These header files also contain documentation on all functions required for accessing the Trueview software functionalities. Headers are only required on the development machine.

1.4. Examples

The Trueview SDK examples demonstrate how to integrate the various parts of the Trueview API together to form an application. These examples are provided in the `Examples` folder. More information on integration can be found in section 5.2.

1.5. License Manager

The License Manager is a standalone command-line utility that can be used to license the Trueview software for development or production machines, including machines that do not have an internet connection. See the section 5.4 for more information on licensing.

1.6. Demonstrator

The Trueview demonstrator only showcases the DR, BH, and CBCT Trueview specialisms. This graphical application performs end-to-end scatter correction of a radiograph or CT scan and, for the BH specialism, the measurement of the bone health in selected regions. It should only be used for demonstration purposes. This application is not covered further by this document.

2. Intended and Indications for Use

The device's intended use is to produce scatter-corrected radiographs without the need for an anti-scatter grid (ASG). For the BH specialism, the intended use also includes the measurement of the bone health in selected regions of the radiographs.

2.1. Intended User

The intended user of this medical device is a software developer who is required to integrate the Trueview software with their medical software.

3. Cleaning and sterilisation

There are no sterility requirements as this is a software-only device.

4. Accuracy

The scatter-corrected radiographs produced by Trueview are in the form of images. The bone health values measured in the BH specialism are estimates of the projected bone thickness (in centimetres) and are reported as single average values for each selected region. No measurement of the accuracy of the outputs are made by the software. The end-user operator is responsible for verifying the quality of the outputs, and for determining whether a scatter-corrected image is of diagnostic quality.

Calibration must be performed prior to using the Trueview software. The calibration process consists of taking a series of radiographs of manufactured samples at varying thickness levels using the settings for each of the imaging protocols. These calibration samples are not part of



the product. Once the calibration samples radiographs are obtained, they must be sent to IBEX to create a calibration database for the chosen imaging protocol. This database enables the Trueview scatter-correction and bone health functionalities for radiographs obtained with this protocol. Please contact IBEX for more information on this process.

Periodic recalibration of the system may be required to compensate for changes in the X-ray system. The frequency of recalibration is dependent on the stability of the X-ray system, so recalibration intervals cannot be specified here.

5. How to use the device

5.1. Setting up the Trueview SDK for development

To be able to develop an application using the Trueview SDK, the Trueview installer must be configured to install at least the following features:

- Trueview ® SDK (*mandatory*).
- If integrating with the C or C++ interface:
 - C/C++ Development Resources (*mandatory*).
 - C/C++ Examples (*recommended*).
- If integrating with the CLR interface:
 - .NET Support (*mandatory*).
 - .NET Examples (*recommended*).
- License Manager (*recommended*).

This will install the required files (DLLs, headers, import libraries, ...) required for compiling programs making use of the Trueview API. The examples, although optional, are a recommended resource for learning how to use the Trueview API; see section 5.2. The “License Manager” feature will install the Trueview License Manager, which can assist in licensing Trueview on the development machine; see section 5.4.

The installer adds the `Lib` folder to the `PATH` environment variable. If this folder needs to be moved for any reason after Trueview is installed, this environment variable must be updated. If the `PATH` environment variable cannot be updated or used, it is the responsibility of the integrator to ensure that all the Trueview DLLs can be found in the DLL search path of their application, at all times. For example, this can be achieved by placing these DLLs in the same directory as the application executable.

The Trueview SDK will assume that the common assets and the license file are located in the same folder as the `IbexSDK.dll` file.

Because of limitations on path lengths in Windows, the Trueview SDK cannot be installed or moved to a location that would lead to path lengths longer than `MAX_PATH` (260 characters). The Trueview installer will not allow installing the SDK at such locations, and the integrator must observe the same restriction if they manually move the SDK to an alternative location.

To prevent Windows from timing out long-lasting GPU computations, hence interrupting the Trueview processing, the Trueview installer modifies (or creates) the registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\GraphicsDrivers\TdrDelay` and sets it to a large value (360 seconds).



Warning: This registry key may be overwritten when updating graphics card drivers, and care must be taken to ensure it always remains set to a large value.

The Trueview SDK is built against CUDA 11.2 (Update 1). This requires an Nvidia graphics card with up-to-date GPU drivers (at least 461.09, released January 2021). Both “Standard” and “DCH” drivers are supported. However, to support “Standard” drivers, the Trueview installer will add the %ProgramW6432%\NVIDIA Corporation\NVSMI folder to the PATH environment variable to enable loading the file `nvml.dll`.

The Trueview SDK and its dependencies require the Microsoft Visual Studio C++ 2019 redistributable. This will automatically be configured by the installer. If copying the Trueview SDK DLLs to another machine, care should be taken to ensure that this new machine also has the appropriate redistributable installed. To use the CLR interface, the .NET Framework v4.6.2 should also be installed (this is installed by default starting from Windows 10 version 1607).

At runtime, the Trueview SDK will attempt to write cache files to the %LOCALAPPDATA%\Local\Temp\ArrayFire folder. If this directory is accessible and writable, the cached data will enable Trueview to run at optimal performance.

Warning. In a fresh setup, the above cache will not yet exist, and the first execution of Trueview will need to create it. This will incur a performance penalty. This will only affect noticeably the first run of Trueview, and the few follow up runs in a lesser extent, until the cache is fully built. The runtime performance of Trueview during the first runs will therefore not be representative of the performance in normal use.

5.2. Integrating with the Trueview API

The Trueview API contains functions for performing scatter correction (and other measurements) for different specialisms. The workflow of the API can be summarized as follows:

1. Initialise the Trueview Context. This must be executed before any interaction with the Trueview API.
2. Load inputs for the chosen protocol. This must be executed each time a new protocol is selected (or once when the application starts, if there is only a single protocol available).
3. Process one or more images obtained with that protocol. This can be repeated for as many images as required, provided they were obtained with the same protocol. If a different protocol must be used, go back to step 2.
4. Close the Trueview Context. This must be executed once the Trueview API is no longer needed.

It is optimal to initialise and close the Trueview Context only once in an application rather than to repeatedly initialise and close it for each processing job.

The Trueview API does not support any form of concurrent access. All interactions with the Trueview API should be carried on in the same execution thread, and only a single process should be using the Trueview API at any point in time on a given machine.

The inputs required for a given protocol are provided by IBEX and include a calibration database (specific to the X-ray device being used, see section 4) and a scatter correction



database. These databases can either be loaded and used explicitly in the API, or they can be loaded and selected automatically based on the capture conditions of the input images.

The input images must be dark-field-corrected and flat-field-corrected, and they must not contain any invalid pixel (infinite, not-a-number, negative, or zero). The input images can optionally be dose-normalised (i.e., the image is divided by the dose, or mAs). If the input images are not dose-normalised, the dose must be provided to the Trueview API. Details on how to load these data in a form suitable for the Trueview API and how to launch the scatter correction procedure are provided in the examples.

The C# examples can be accessed by opening the Visual Studio solution file `Examples\CSharp\Examples.sln`. For detailed information, please refer to:

- **(CBCT ONLY)** The ConeBeamComputedTomography project.
- Other modalities: C# examples are not provided for the moment.

The C++ examples are listed in the `Examples\CPP` folder. They are configured using CMake, which can be used to produce a Visual Studio solution file or to produce build scripts for any other build system supported by CMake. The CMake scripts use the `TRUEVIEW_ROOT_DIR` variable to locate the Trueview API (i.e., the folder containing the `Include` and `Lib` sub-folders). By default, this variable is initialised assuming the standard folder structure created by the Trueview installer. Therefore, no extra command line argument needs to be provided to CMake unless the example files have been copied to an alternative location. The generated executables assume that the `Lib` folder has been added to the `PATH` environment variable. For detailed information, please refer to:

- **(DR ONLY)** The file `DigitalRadiography\DigitalRadiography.cpp`.
- **(BH ONLY)** The file `BoneHealth\BoneHealth.cpp`.
- **(MAMMO ONLY)** The file `Mammography\Mammography.cpp`.
- **(CBCT ONLY)** For integrating Trueview with forward projection data, see the file `ConeBeamComputedTomography\ForwardProjectionRun.cpp`. Else, see the file `ConeBeamComputedTomography\StandaloneRun.cpp`.

5.3. Installing Trueview on a production machine

To be able to run an application developed with the Trueview SDK, the Trueview installer must be configured to install at least the following features:

- Trueview ® SDK (mandatory).
- If Trueview is integrated into a CLR application:
 - .NET Support (*mandatory*).
- License Manager (*recommended*, or *mandatory* when no internet connection is available).

This will install the required DLLs for running an application making use of the Trueview API. The “License Manager” feature will install the Trueview License Manager, which is required if the machine has no internet connection and has not already been licensed for using the Trueview SDK. This tool may also be necessary if the software that integrates the Trueview SDK does not provide its own solution for licensing. See the section 5.4 for more information.



The instructions from section 5.1 regarding the `PATH` environment variable, the location of the Trueview SDK DLLs, the restrictions on path lengths, the required modifications to the Windows registry, the GPU drivers, the Microsoft Visual Studio redistributable, and the .NET Framework also apply to production machines.

Warning. As discussed in section 5.1, the very first execution of Trueview will need to create cache files in the `%LOCALAPPDATA%\Local\Temp\ArrayFire` folder. This will incur a one-time performance penalty, and can be an issue if consistent, optimal performance is a critical requirement. In such cases, pre-built cached data can be copied onto the production machine from another machine with the same GPU device and drivers, using the above location.

5.4. Licensing

The Trueview software requires a machine to be licensed before that machine can use any of the software's functionalities. IBEX provides two types of licenses: development licenses and production licenses. Development licenses are typically capped to a small number of activated machines and have a fixed expiration date. Conversely, production licenses are not capped and will never expire.

The Trueview licensing involves creating a license file for the machine running the software. This file is checked by the Trueview software to verify the license validity before the software's functionalities can be accessed. The procedure to generate this file depends on whether the machine has access to the internet or not. If internet access is available, the Trueview software can use a fully automated procedure which is documented in the code examples. This procedure requires providing a license key and token to the Trueview API. Otherwise, the License Manager utility must be used. More information on licensing can be found by running this utility with the `-Help` command line option.

Once a license file is created, no internet connection is required until the license needs re-validating. A license needs re-validating if it expires (only applicable to development licenses), or if the machine's hardware changes significantly since the file was first created. If no internet access is available when the license needs re-validating, the License Manager utility must be used.

In addition to the Trueview license, users of the Trueview SDK are bound to the licenses of the third-party dependencies used by the Trueview SDK. These can be accessed in the `Licenses` folder created by the installer, along with the Trueview End User License Agreement (EULA). Some of these third-party licenses have specific requirements that must be fulfilled by users of the Trueview SDK. The most common requirement is to include the text of the corresponding third-party license in any distribution of derivative products (which includes any product built against the Trueview SDK). Other requirements may apply; please refer to the individual licenses for more information.

5.5. Update or uninstall the device

The installer can be used to perform updates or to uninstall the Trueview software or its SDK. Updates require running a new installer which will be provided by IBEX.

As discussed in section 5.1, the Trueview SDK writes cache files to the `%LOCALAPPDATA%\Local\Temp\ArrayFire` folder. It is recommended to clear this cache in the



following situations: when graphics card drivers are updated, or when the graphics card is physically replaced by another model.

Warning. If the cache is cleared after an update, the runtime performance of Trueview will be noticeably impacted until the cache is fully re-built (see sections 5.1 and 5.3). Therefore, this operation should only be performed in an environment where runtime performance is not critical. Alternatively, an updated cache can be installed following the procedure discussed in 5.3.

Warning. When updating or uninstalling the Trueview software, make sure no application is running on the operating system that could make use of the Trueview software. This includes applications built with the Trueview API, as well as development environments that are used to build such applications (e.g., Visual Studio).

6. Restrictions on use

The body type of certain people may fall outside the calibration that has been conducted. This may lead to a reduction in the ability of the software to image fine detail.

7. Risks

The software device will be run on a Microsoft Windows 10 Professional computer that is independent and not directly connected to the X-ray equipment. If these instructions are followed this device may be used safely without risk to patient or radiographer.

If the Trueview software fails to process a given radiograph for any reason, no scatter-corrected radiograph and no bone health indicator will be generated. The Trueview API or the operating system will report such failures, and it is the responsibility of the integrator to capture and act on these errors. In particular, it is the responsibility of the integrator to back up the original image data in case an unrecoverable error forces the process to close.

Finally, the Trueview API associates to each output a quality indicator. This indicator can be used to flag potential issues with the processing or with the input image, but it is only indicative.

8. Hazard summary

There are no hazards related to use of this device during demonstration due to the following:

- Essential performance of the X-ray system does not depend on this device.
- The device is being run on a PC independent from the X-ray equipment.

9. Cautionary notices

None

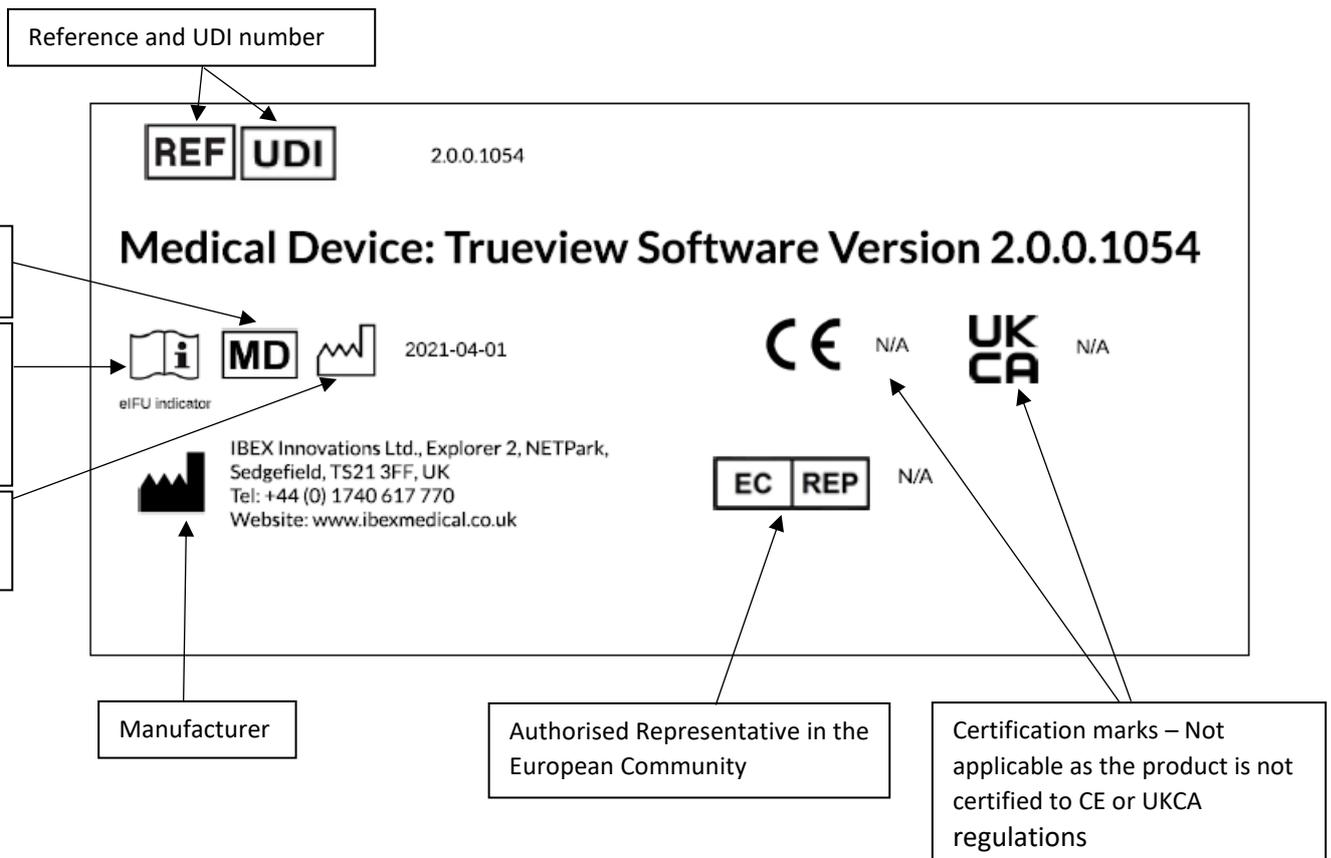
10. Packing and transport

Not applicable.



11. Symbols, Labelling and Manufacture

The label to be used with this version is shown below; it is highlighted with the symbol meanings.





12. DOCUMENT APPROVALS

This document approved by:

Reviewer:

Name (print):

Corentin Schreiber

Signature:

Date:

Quality Manager:

Name (print):

Lisa March

Signature:

Date:



13. REVISION HISTORY

Version number	Changes made	Author(s) initials
1.0	Original release for Trueview version 1.2.0	KH
1.1	Updated for Trueview version 1.4.0	KH, CS
1.2	Addition of user description	KH
1.3	Updated for Trueview version 1.4.1	CS
1.4	Updated for Trueview version 1.4.2	CS
1.5	Updated for Trueview version 1.5.0. Added limitation of file path lengths. Added information on alternative loading of protocol data. Added limitation on multi-threading.	CS
1.6	Updated for Trueview version 1.5.1. Added more information on environment variables when using the CLR interface. Added information on TdrDelay registry key.	CS
1.7	Updated for Trueview version 1.6.0. Updated for removal of CommonLibs folder, installer feature for system environment, IBEX_TRUEVIEW_PATH environment variable, and Windows GAC. Added instructions for third-party software licenses.	CS
1.8	Updated for Trueview version 1.6.1	CS
1.9	Updated for Trueview version 1.6.2. Added information on Visual Studio redistributable. Fixed formatting.	CS
1.10	Updated for Trueview version 2.0.0. Updated dependency information to show that VS C++ 2015 runtime is no longer needed. Clarified what happens when a Trueview license needs re-validating. Added more precise pointers to the relevant example code for each modality. Updated after splitting the C++/C interface and .NET Framework interfaces into separate DLLs. Added more information on the impact of the ArrayFire cache, and how to mitigate the impact. Updated for inclusion of CBCT into the demonstrator. Updated after renaming BMD to Bone Health. Updated the label.	BR, CS