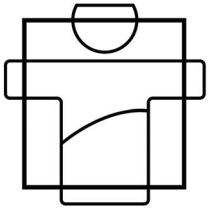
OsiriX MeVisLab Bridge 3.1

Felix Ritter



Fraunhofer
MEVIS

Introduction

To facilitate the integration of the MeVisLab macOS edition into the clinical workflow, you may use the OsiriX Advanced DICOM PACS workstation or its descendant Horos[1] to retrieve and store medical image datasets. The solution introduced in this document enables a bidirectional communication between MeVisLab and OsiriX version 3.3 or later. Images may not only be sent from OsiriX to MeVisLab for processing, but furthermore, images can be sent back to OsiriX for PACS storage as well.

1. In the following only OsiriX is described. Currently, there are no significant differences between Horos and OsiriX regarding this data exchange bridge.

Integrating MeVisLab with OsiriX

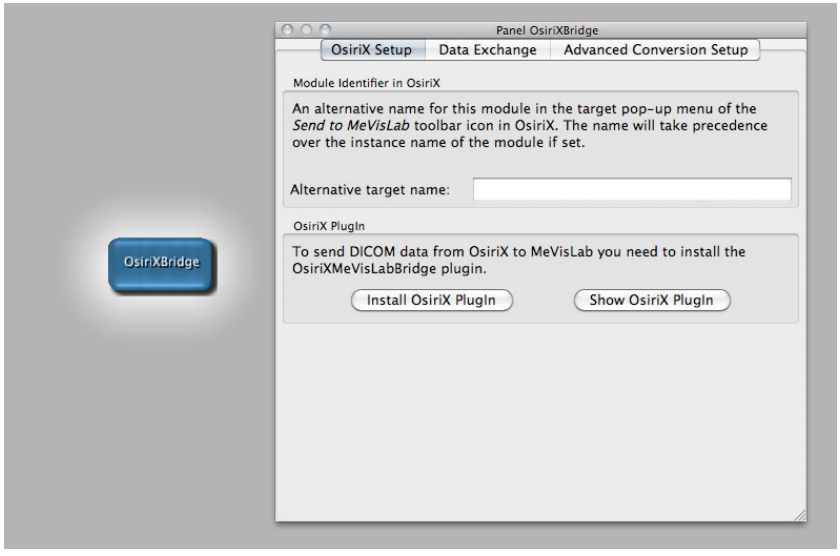
The following sections explain the setup and adoption of the OsiriX MeVisLab bridge for your MeVisLab installation.

Installation of the OsiriX plugin

The bridge between MeVisLab and OsiriX consists of the OsiriX plugin OsiriXMeVisLabBridge and the MeVisLab module OsiriXBridge. Currently, MeVisLab and OsiriX must be running on the same computer.

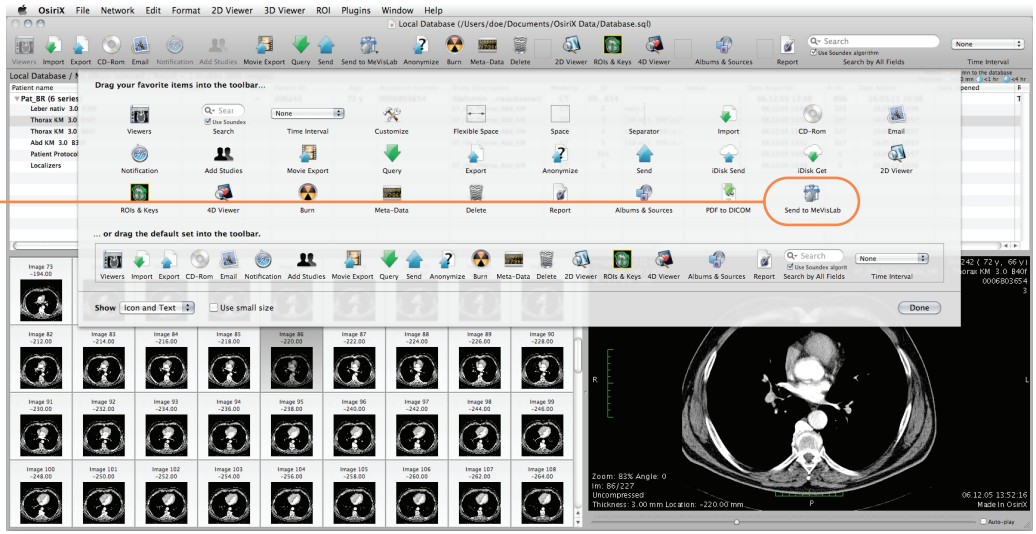
To install the OsiriX plugin, start MeVisLab, place the module OsiriXBridge onto the network window and open its panel by double-clicking the module. The *OsiriX Setup* tab features an *Install OsiriX PlugIn* button, that will hand the plugin to OsiriX for installation (Fig. 1).

Figure 1 Installation of the OsiriX plugin via the MeVisLab module OsiriXBridge



After the OsiriXMeVisLabBridge plugin has been loaded by OsiriX, customize the OsiriX toolbar by selecting *Customize Toolbar…* from the *Format* menu in OsiriX. Then simply drag and drop the *Send to MeVisLab* icon into the toolbar (Fig. 2).

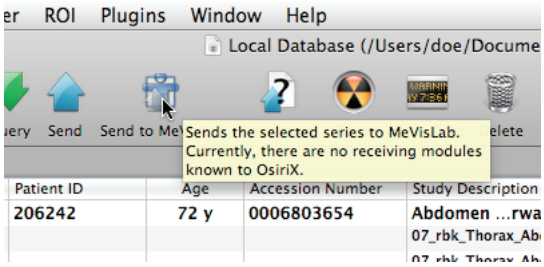
Figure 2 Adding a *Send to MeVisLab* icon to the OsiriX toolbar



Sending image series from OsiriX to MeVisLab

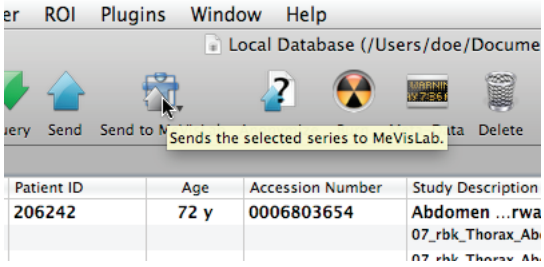
All interaction with the OsiriXMeVisLabBridge plugin is performed via the *Send to MeVisLab* toolbar icon. The icon's appearance and tooltip indicate the current state of the connection between OsiriX and MeVisLab. If OsiriX does not detect any MeVisLab modules which are able to receive data, the arrow in the icon will be translucent (Fig. 3).

Figure 3 OsiriX could not detect any OsiriXBridge modules



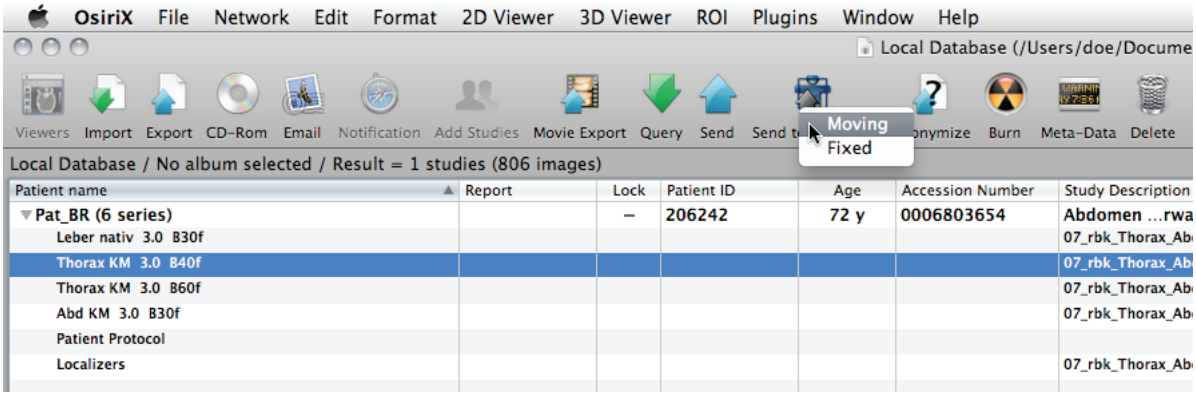
If at least one target module could be detected, the arrow will be shown fully opaque (Fig. 4).

Figure 4 At least one OsiriXBridge module is available



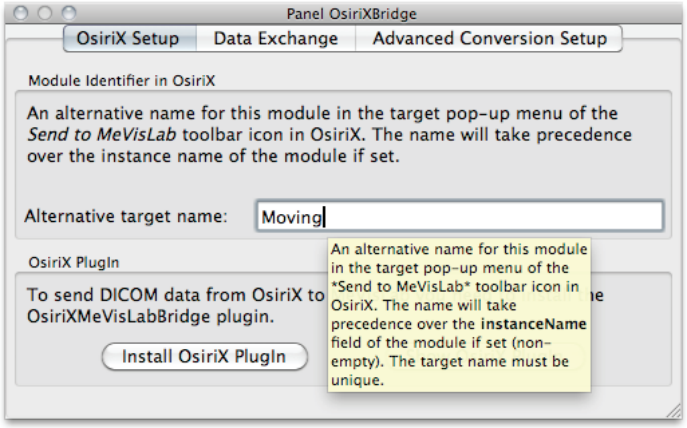
To send a specific set of images to MeVisLab select the name of the series you would like to send in the patient and study list. Then click on the *Send to MeVisLab* icon in the toolbar. A pop-up menu will open and allow the user to select one of the detected targets (Fig. 5).

Figure 5 Sending a selected series via the target pop-up to MeVisLab



Each target represents one OsiriXBridge module or macro module that employs an OsiriXBridge module. The target name can be set in the module's panel *OsiriX Setup* tab in MeVisLab (Fig. 6). Initially the instance name of the OsiriXBridge module is used as the target name but you are free to change that name to better reflect the targets meaning. Target names must be unique. If a target name has already been registered with OsiriX, the name is suffixed by a counting number.

Figure 6 Setup of the name of the target pop-up menu item in OsiriX



Using MeVisLab to process images sent from OsiriX

Direct Dicom Import or Conversion?

MeVisLab supports two approaches to load DICOM data. The first, traditional approach, loads DICOM data after a conversion step performed by the EatDicom tool. EatDicom analyses and sorts the DICOM slices, builds series and finally creates two files per series, a DICOM header file and a 3D-TIFF file. Both files can be loaded by passing the DICOM header file to an ImageLoad module.

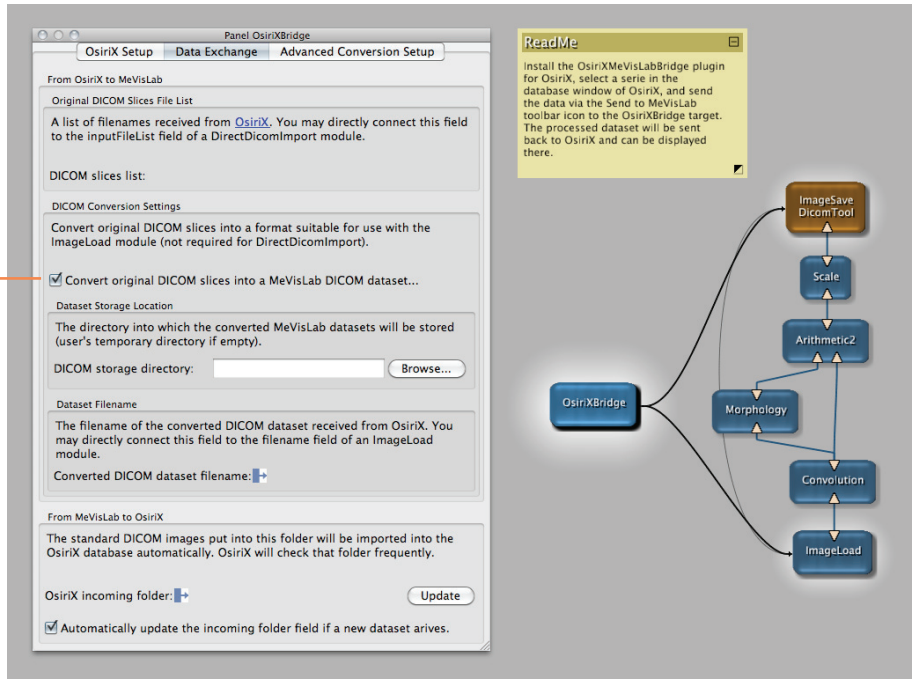
The second, more modern approach, reads the DICOM slices directly without writing an explicit intermediate format file. Using a DirectDicomImport module, the DICOM slices are analyzed and sorted. Multiple volumes can be provided sequentially or in parallel. You may query the module for the properties of the loaded series. See the module help for further information.

EatDicom

The DICOM images sent by OsiriX are automatically converted by the OsiriXBridge module into a dataset that can be loaded by the ImageLoad module of MeVisLab. By connecting the filename field of the OsiriXBridge module to the filename field of an ImageLoad module, the converted dataset will be loaded by the ImageLoad module as soon as it becomes available.

The EatDicom mode of the OsiriXBridge module is determined by the boolean state of the useEatDicom field (Fig. 7).

Figure 7 Example network of the OsiriXBridge module using an ImageLoad module to load DICOM data that have been converted by the EatDicom tool



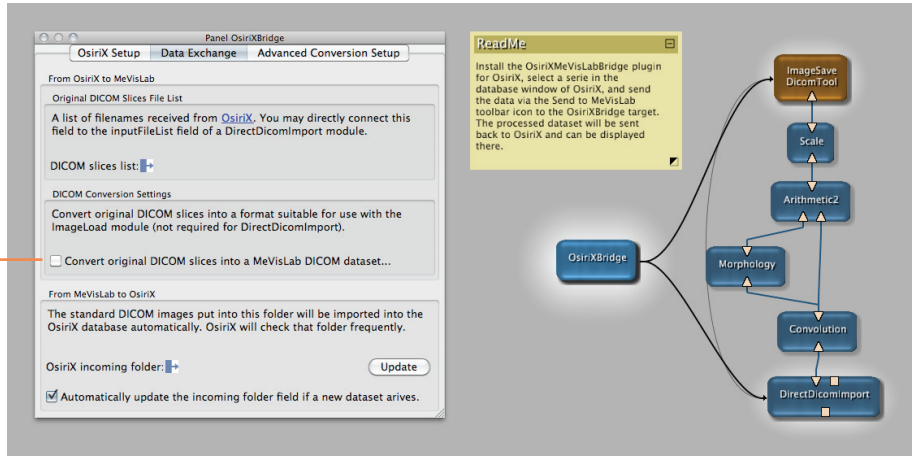
DirectDicomImport

To interface the OsiriXBridge module with the DirectDicomImport module, first connect the OsiriXBridge.slicesFileList field to the DirectDicomImport.inputFileList field. Second, connect the DirectDicomImport.inputFileList field to the DirectDicomImport.dplImport field. This will automatically trigger the import process if a list of files has been passed to DirectDicomImport. As a convenience to the user, the OsiriXDirectDicomImport macro module already provides this configuration.

The DirectDicomImport mode does not require the conversion step outlined in the Direct Dicom Import or Conversion? section. Starting

with MeVisLab 2.2 the conversion if off by default in the OsiriXBridge module panel (Fig. 8).

Figure 8 Example network of the OsiriXBridge module using a DirectDicomImport module to load the DICOM data directly from the OsiriX database



Sending images back to OsiriX

If you want to store a processed dataset into the OsiriX database, use the DicomTool macro module and save the dataset slice by slice to the Incoming folder of the OsiriX database. The path to this folder can be obtained from the osirixIncomingDir field of the OsiriXBridge module. Simply connect the osirixIncomingDir field to the exportBaseDir field of the DicomTool macro module and trigger the saveSlices field.

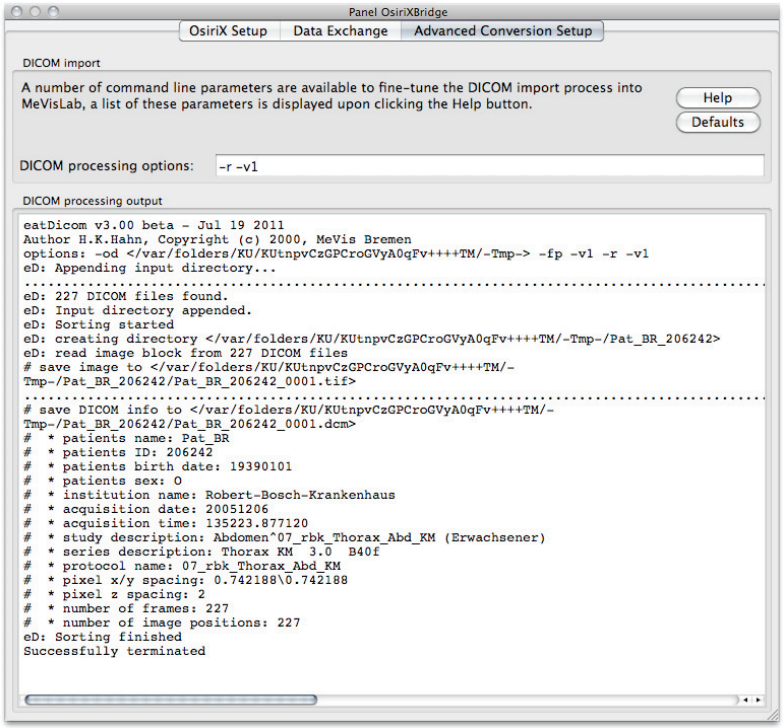
The shown example networks of the OsiriXBridge module use the MeVisLab-OsiriX bridge to pipe a series sent by OsiriX through a MeVisLab filter chain. The filtered dataset is sent back to OsiriX (Fig. 7 & Fig. 8).

Operation details

The OsiriXBridge uses Objective-C distributed objects to establish a bidirectional communication between OsiriX and MeVisLab. OsiriX sends the filenames of all slices of the current dataset which are then converted by the EatDicom utility of MeVisLab depending on the state of the useEatDicom field. The resulting filename is passed to the filename field of the OsiriXBridge module.

The import of the DICOM slices sent by OsiriX is performed by the EatDicom import tool, a command line utility that can also be used independently of MeVisLab (e.g. as a background process). A number of command line parameters are available to fine-tune the import process, a list of these parameters is displayed by clicking the *Help* button in the *Advanced Conversion Setup* tab of the OsiriXBridge module panel (Fig. 9).

Figure 9 Setup and output of the DICOM import process



Troubleshooting

If the import of DICOM images sent by OsiriX fails using the conversion approach, you may be required to customize the process. See the help page of the EatDicom tool available by clicking the Help button in the *Advanced Conversion Setup* tab of the OsiriXBridge module panel for tuning options.

In addition, the standard and error output of the EatDicom import tool are displayed in the *Advanced Conversion Setup* tab. This may help to identify import issues (Fig. 9).

XML-RPC support

OsiriX integrates an HTTP server with XML-RPC protocol support. The OsiriX MeVisLab Bridge extends the callable methods by the following:

Method:	SendSelectedSeriesToMeVisLab
Parameters:	{target: "OsiriXBridge"}; name of the target
Response:	{error: "0"}; -1: service not found, -2: service transmission exception, -3: communication protocol mismatch, -4: unknown target name

Obtaining the source code

The source codes of both the OsiriX plugin and the MeVisLab module are available as part of the GitHub MeVisLab Community project.

Document Revision History

Date	Notes
2017-01-23	Obtaining the source code updated
2012-07-24	Sending image series from OsiriX to MeVisLab updated XML-RPC support updated Obtaining the source code updated
2011-08-02	Sending image series from OsiriX to MeVisLab added Using MeVisLab to process images sent from OsiriX updated
2011-05-16	Installation of the OsiriX plugin updated XML-RPC support updated Obtaining the source code updated
2009-12-21	Installation of the OsiriX plugin updated XML-RPC support added
2009-04-22	New document that introduces the OsiriX MeVisLab Bridge