



ROMEXIS PMBRIDGE V.2.6.0 SPECIFICATION

JKN/26.10.1998
EJL/16.08.1999
EJL/14.10.1999
EJL/05.03.2001 (review)
MKH/2005 (review)
VS/2010 (review)
MKH/2014(review)

1.	ROMEXIS PMBRIDGE v2.6.0 Interface Specification	4
1.1	Background.....	4
1.2	The big picture	4
1.3	Planmeca implementation of PMBRIDGE for ROMEXIS	5
2.	Using the PMBRIDGE	6
2.1	Library	6
2.2	Functions	6
3.	PMBRIDGE Functions.....	6
3.1	Session Related Functions.....	6
3.1.1	At the beginning of the session.....	6
3.1.2	Terminating the session.....	7
3.1.3	Application window behavior.....	7
3.1.4	Set location of the main window	7
3.1.5	Force Romexis to be on top.....	7
3.1.6	Hide Toolbar & Statusbar of ROMEXIS.....	7
3.1.7	Define Limits for the ROMEXIS Window	8
3.2	Image Related Functions.....	8
3.2.1	Radiograph request	8
3.2.2	Opening the image acquisition dialog	9
3.2.3	Opening the image acquisition dialog with study option	9
3.2.4	Opening the image selection dialog.....	9
3.2.5	Getting information about images of the current patient.....	9
3.2.6	Setting Image Retrieval Window Size.....	10
3.2.7	Receiving of the image data	10
3.2.8	Getting a thumbnail image	11
3.3	User and Patient Related Functions.....	11
3.3.1	Adding a new patient	11
3.3.2	Modifying patient information	12
3.3.3	Removing patient	12
3.3.4	Deselection of the current patient	12
3.3.5	Setting ROMEXIS User.....	12
4.	PmBridge / Romexis Information Fields	12
4.1	PatientID	12
4.2	FamilyName.....	13
4.3	FirstName	13
4.4	Birthdate	13
4.5	Doctor(optional)	13
5.	Best practices	13
5.1	Opening patients.....	13
5.2	High Level Bridge DxStart	13
5.3	Using DxStart.....	14
6.	Installation.....	14
6.1	Windows Operating System	14
6.2	Macintosh Operating System	15
6.3	Conflicts with DIMAXIS PMBRIDGE	15
7.	Other	15
7.1	Example Program DxClient	15
7.2	Unicode Compatibility	27



ROMEXIS PMBRIDGE 2.6.0 SPECIFICATION

Planmeca reserves the right to make changes to this specification without a further notice. Planmeca disclaims any liability, including without limitation consequential or incidental damages related to the use the contents of this document. There are no express or implied copyrights or patent licenses granted hereunder to copyrighted material or applicable patents of Planmeca or others.

Having said that, we hope the information presented here will be useful and we will proceed to improve both the documentation and the PMBRIDGE standard it describes, as well as other support material for the PMBRIDGE interface. All comments and suggestions are highly appreciated -- we hope to establish open and productive relationships with third party software developers as well as other dental imaging device manufactures.

Please send comments etc to:

Email: [markku.hatakka @planmeca.com](mailto:markku.hatakka@planmeca.com)

1. ROMEXIS PMBRIDGE v2.6.0 Interface Specification

This document describes the PMBRIDGE interface for PLANMECA ROMEXIS imaging software. Although it has been defined exclusively by Planmeca Group, it is an open system that we hope will be widely used by third party software developers and dental imaging device manufacturers.

The version number of the PMBRIDGE interface refers to current ROMEXIS version number at the time that the interface has been modified. Possible modifications will always be done so that the library will remain compatible with older versions. All the functions presented in this document refer to this version of the PMBRIDGE.

Note that there are separate versions of the PMBRIDGE for PLANMECA DIMAXIS and PLANMECA ROMEXIS Imaging Software. Additionally there are separate files for Windows and Mac OS versions of PMBRIDGE for ROMEXIS. They are delivered on the installation DVD/CD's of the respected products and are not compatible with both imaging applications. For more information of the PMBRIDGE for DIMAXIS, see the PMBRIDGE specification on the DIMAXIS installation CD.

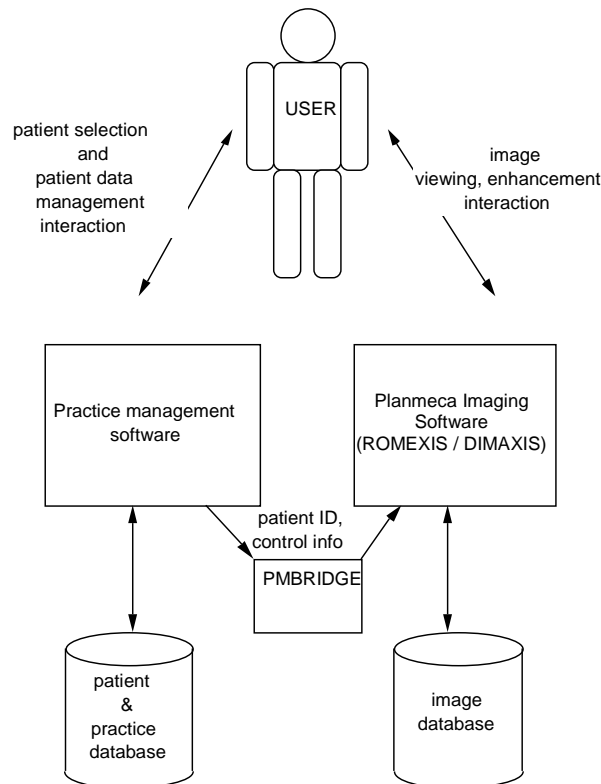
Note also that the version numbers of the respective DLL files depend on the imaging software version number (DIMAXIS PMBRIDGE versions go up to 4.5.0, whereas ROMEXIS PMBRIDGE at the moment up to 2.2.0)

1.1 Background

From the beginning Planmeca recognized that to be a successful part of a dental practice the digital imaging devices must be seamlessly integrated with the rest of the computer system. To make total integration possible Planmeca has previously created an interface standard called DIDAPI, including supporting libraries. DIDAPI gives direct access to image acquisition, which makes it possible for software houses to seamlessly integrate all aspects of the digital imaging into their practice management software. However, it has turned out that not all of the software houses are willing to invest in the time and money to develop a system with that level of integration. Instead many software houses are looking for simple way to "bridge" their practice management software with the imaging software supplied by Planmeca (called ROMEXIS & DIMAXIS). For this purpose Planmeca has defined a standard called PMBRIDGE and the supporting libraries and documentation.

1.2 The big picture

The co-operation of the practice management software and the image manipulation software is illustrated in the following diagram.



In this integration model the Planmeca Imaging Software will handle all the image manipulation and viewing related operations and user interactions. All the patient/practice data related operations (including patient selection) and interactions are to be handled by the practice management software. This system makes it very simple and quick for the practice management software house to create this link or bridge to the Planmeca Imaging Software. The actual coding time is measured in hours.

All that the practice management software has to do is to start the Planmeca Imaging Software (minimized) at start of the session. When the user clicks at a button (labeled e.g. "X-RAY"), the patient ID is passed to the imaging software by the PMBRIDGE library along with the request for the imaging software to show its windows. That's all. The imaging software will then display a dialog where all the patient images are listed and the user continues the interaction with the imaging software viewing, manipulating, printing etc.

The downside of this type of bridging is that it is not totally seamless which means that the user does see two programs with possibly different look-and-feel.

1.3 Planmeca implementation of PMBRIDGE for ROMEXIS

Currently the PMBRIDGE v2.6.0 for Planmeca ROMEXIS is supported on the following platforms:

Windows 8.1 or lower, and Mac OS 10.9.4 or lower.

The behavior of the ROMEXIS software is subtly changed when used via PMBRIDGE system: the splash screen at the launch of the application is omitted and the patient selection related functions are disabled as this will be handled by the practice management software via the bridge.

2. Using the PMBRIDGE

2.1 Library

The PMBRIDGE is supplied in the form of a DLL library (Windows)/DYLIB(OS X) , associated header file 'PmBridge.h' and static library '.LIB'(Windows).

Please note that the latest PmBridge.DLL / PMBRIDGE.DYLIB that comes with ROMEXIS installation (Windows folder Romexis\Pmbridge\Program, OSX folder /usr/lib) should always be used (Except when developing on OS X HOME/lib can be used also). The libraries may not be copied or moved to other locations, because otherwise the link will not work.

(However sometimes an application can not find the library and 'Romexis.bat' start program, although the Windows system variable PATH contains a correct folder path. Then one needs to copy PmBridge.dll and Romexis.bat files into application work folder.)

It is recommended to use LoadLibrary() and GetProcAddress() calls for using PmBridge library instead of linking with static PmBridge.LIB. This way you can keep your program compatible with older PMBRIDGE versions which don't have the new functions.

2.2 Functions

A sample code, which can be used to simulate and investigate the practice management software side of bridging, is supplied with SDK for PMBRIDGE. The code is supplied in form of a complete MS Visual Studio 2010 project and XCode project on OS X.

All the functions return a result code that has either the value PMB_OK or signals some kind of abnormal condition.

3. PMBRIDGE Functions

3.1 Session Related Functions

3.1.1 At the beginning of the session

At the beginning of the session start the ROMEXIS software as follows:

```
err = DxStart();
```

3.1.2 Terminating the session

It is possible to terminate ROMEXIS by calling

```
err = DxExit ();
```

This could be done when the user is closing the administrative application. If ROMEXIS has modal dialog open, it can't be terminated. In such a case the return code of `DxExit()` is `PMB_DX_BUSY`.

3.1.3 Application window behavior

The practice management software may want to minimize the ROMEXIS application when the practice management software is switched to the foreground. This can be done with a call:

```
err = DxSetWindowPlacement (SW_SHOWMINIMIZED);
```

ROMEXIS can be configured to remember the window position after exiting the program. This feature is often useful when using ROMEXIS via PMBRIDGE. It is also possible to configure ROMEXIS to stay "Always on top". This is useful for example when a dentist wants to write diagnose of an image in the practice management software while viewing the image in a ROMEXIS window.

3.1.4 Set location of the main window

It is possible to explicitly specify the location of the main window of Romexis by calling

```
DxMoveWindow (x0, y0, nWidth, nHeight );
```

where `x0` and `y0` specify the upper left corner of the window and `nWidth` and `nHeight` specify the size of the window.

3.1.5 Force Romexis to be on top

Romexis can be forced to be always on top by calling

```
short nTop=1;
```

```
DxAlwaysOnTop (nTop);
```

3.1.6 Hide Toolbar & Statusbar of ROMEXIS

It is possible to hide Toolbar and Statusbar of Romexis to get more room for the images

```
extern "C" short WINAPI DxShowToolbar (  
    short nShow // 1 = show, 0 = hide  
);  
  
extern "C" short WINAPI DxShowStatusbar (  
    short nShow  
);
```

3.1.7 Define Limits for the ROMEXIS Window

Define limits for the Romexis window when the window is maximized. Using this function it is possible to make sure that the calling application's menu is visible when the window is resized. A value will not be used if it ≤ 0 . The values are relative to the actual screen size. (This has been tested only in single monitor systems). Romexis also does a 'sanity' check for the values and uses value only if it does not limit screen usage more than 10%.

```
extern "C" short WINAPI DxSetMaxLimits (  
    long nMinTop,  
    long nMaxBottom,  
    long nMinLeft,  
    long nMaxRight  
);
```

With function

```
extern "C" short WINAPI DxSetWorklistEntry ();
```

it is possible to pass to Romexis the same information as is used in DICOM Worklist. See PmBridge.h for details.

3.2 Image Related Functions

3.2.1 Radiograph request

When the user requests to see the radiographs of a particular patient, pass the patient information to ROMEXIS as follows:

```
// Do what ever you need to get the ID  
  
char* patientID = GetCurrentPatientID();  
  
short noOfImages;  
  
err = DxSelectPatient (patientID, &nNumberOfImages);  
  
if (err==PM_OK)
```


Please note that the patient ID must exactly match the ID used in ROMEXIS. In ROMEXIS it is possible to specify minimum patient ID length. If the given ID is shorter than the specified minimum, ROMEXIS adds leading zeroes to the ID. Default value for minimum patient ID in ROMEXIS is 1, so no leading zeros will be added by default.

Immediately after that force ROMEXIS to show its window:

```
err = DxSetWindowPlacement(SW_SHOWNORMAL);
```

That's all that there is to the basic pattern of interaction.

3.2.2 Opening the image acquisition dialog

Image acquisition dialog can be opened by calling `DxCaptureImage()`, for example

```
short nImageType = DX_IT_PAN ;  
DxCaptureImage (nImageType);
```

3.2.3 Opening the image acquisition dialog with study option

Image acquisition dialog with study selection window can be opened by calling `DxCaptureImageIntra()`

```
extern "C" short WINAPI DxCaptureImageIntra (  
    short nUseStudy, short nAllowWithoutStudy  
);  
  
nUseStudy ... 1 open Study Selection window, 0 don't use Study  
Selection window  
nAllowWithoutStudy ... 1 allow capture without Study, 0  
don't allow capture without study
```

3.2.4 Opening the image selection dialog

A dialog for selecting images can be opened by calling

```
DxSelectImage ();
```

Note that there are several ways of selecting images in Romexis. The default method can be configured in Romexis Settings.

3.2.5 Getting information about images of the current patient

It is possible to get information about images of the current patient by calling

```
DxInquireImage (
```

```
short nIndex,           // 0 .. nImageCnt-1
long  arrImageParam [], // Size must be at least DX_MAX_INDEX
short arrToothMask  []  // 1-based table for dentistry
);
```

nIndex is the 0 –based index of the images of the patient. The number of the images the patient has can be received when calling DxSelectPatient()

arrImageParam[] is a 1-based array of the image parameters. The size of the array must be at least DX_MAX_INDEX. The indices of the table have been defined in "PmBridge.h".

3.2.6 Setting Image Retrieval Window Size

Image Navigator (Image Retrieval) is one of the ways of selecting images in Romexis . It is possible to set the Image Retrieval Window size by calling

```
extern "C" short WINAPI DxSetImageRetrievalWindowSize (
short nWindowSize // DX_RWND_SIZE_SMALL or DX_RWND_SIZE_FULL
);
```

3.2.7 Receiving of the image data

The image data can be received by calling

```
DxGetImage (
short nIndex,
short nConvertTo8bit,
short nEnhance,
short nFileFormat,
const char *pszFileName
);
```

nIndex see DxInquireImage() above

nConvertTo8bit is a flag. The x-ray images in Romexis are normally stored with 16(12) bits. If this flag is set to 1, the image will be mapped to 8 bits. Most applications are only able to handle 8 bit images.

nEnhance is a flag. If this flag is set to 1, the contrast of the image will be enhanced before saving the image to file. For cephalometric images the soft tissue filter will be applied. If the image has been explicitly saved, any other image processing functions (filtering etc) will be applied to the image.

nFileFormat defines the image file format; see "PmBridge.h" for definitions of the formats.

pszFileName is the output file name. It should include the extension (.TIF, .JPG or .BMP depending on nFileFormat).

or by calling

```
extern "C" short WINAPI DxGetImageByID (
    int nImageID,
    short nConvertTo8bit,
    short nEnhance,
    short nFileFormat,
    const char *pszFileName
);
```

Otherwise the same as `DxGetImage()` except the `nImageID` is the real image number which is fetched from the database with the function `DxInquireImage()` into the table `arrImageParam[]` with index value 10.

3.2.8 Getting a thumbnail image

Besides image data, it is possible to get the thumbnail (overview) image by calling

```
extern "C" short WINAPI DxGetOverviewImage (
    short nIndex,
    short nFileFormat,
    const char *pszFileName
);
```

See `DxGetImage` for definition of the parameters.

or by calling

```
extern "C" short WINAPI DxGetOverviewImageByID (
    int nImageID,
    short nFileFormat,
    const char *pszFileName
);
```

Otherwise the same as `DxGetOverviewImage()` except the `nImageID` is the real image number which is fetched from the database with the function `DxInquireImage()` into the table `arrImageParam[]` with index value 10.

`(#define DX_INDEX_IMAGE_ID 10 // ID for the patient's image in the Romexis/Dimaxis database)`

3.3 User and Patient Related Functions

3.3.1 Adding a new patient

Of course, when a new patient is added to the practice management database the corresponding information needs to be passed to ROMEXIS as well. To do that:

```
err = DxInsertPatient
(patientID, familyName, firstName, birthDate, doctor);
```

3.3.2 Modifying patient information

When patient information is modified in the practice management system, (name changes due to marriage etc.) the corresponding information can be passed to ROMEXIS:

```
err = DxUpdatePatient  
(patientID, familyName, firstName, birthDate, doctor);
```

3.3.3 Removing patient

Unneeded patients can also be removed from the ROMEXIS database via PMBRIDGE, but only if the patient don't have any images in the database:

```
err = DxDeletePatient (patientID);
```

3.3.4 Deselection of the current patient

The current patient can be deselected by calling `DxDeselectPatient()`.

3.3.5 Setting ROMEXIS User

The ROMEXIS user can be set by calling

```
DxSetUser (  
    char *pszUserName,  
    unsigned short nAccessLevel  
);
```

`pszUserName` will be used as the default for the user name which will be saved with the new images

`nAccessLevel` can be used to limit the rights of the user. It is possible to define if the user can acquire images, delete images or do administrative tasks (like change the settings) in Romexis . The access level parameters are defined in "PmBridge.h". The lowest access level allows the user to view images. The access level parameters are used as a bit mask; so they will be added together. For example if the user is allowed to make exposures and delete images, the access level is

```
unsigned short nAccessLevel = DX_ACL_ACQUIRE_IMAGE +  
                               DX_ACL_DELETE_IMAGE ;
```

4. PmBridge / Romexis Information Fields

4.1 PatientID

PatientID can be any string of a maximum length of 64 characters.

4.2 FamilyName

FamilyName can be any string of a maximum length of 64 characters.

4.3 FirstName

FirstName can be any string of a maximum length of 64 characters.

4.4 Birthdate

Birthdate should be a string of 8 numbers of format YYYYMMDD. However, also formats YYYY*MM*DD and DD*MM*YYYY are recognized and converted to the right format.

4.5 Doctor(optional)

Doctor can be any string of a maximum length of 64 characters.

5. Best practices

5.1 Opening patients

It is recommended to use the following PMBRIDGE command sequence regarding the opening of patients in ROMEXIS (see DxStart for more info):

Select Patient

-> If Patient not found, try to insert patient

-> If Patient found, try to update patient

5.2 High Level Bridge DxStart

Some integrators have complained that PMBRIDGE is a bit complicated to use. Therefore a higher level bridge, DxStart, has been developed by Planmeca. DxStart encapsulates the complexities of PMBRIDGE, but the functionality is limited. See document "DxStart.doc" for more details.

5.3 Using DxStart

It is advisable to call the `DxStart()` function always before interacting with ROMEXIS i.e. call it not only at the beginning of the session but also when the patient is selected and you want to ROMEXIS to display the radiographs. This is not absolutely necessary but is good practice as it ensures that the ROMEXIS is running (the user may have quitted the application since you started it). Calling `DxStart()` causes minimal overhead if the ROMEXIS application is already running.

6. Installation

6.1 Windows Operating System

See the ROMEXIS installation manual for up to date installation instructions for ROMEXIS PMBRIDGE.

Please make sure that you only have one copy of `PmBridge.DLL` in the PC! The DLL must be located in the following ROMEXIS directory:

C:\Program Files\Planmeca\Romexis\Pmbridge\Program

You should not copy `PmBrige.DLL` to your own directory because the dependency with the windows registry will then be broken.

You can obtain the path to ROMEXIS PMBRIDGE directory from the registry under the following key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\Dimaxis.exe]

The path is of form

"Path"="C:\Program Files\Planmeca\Romexis\pmbridge"

Thus the subdirectory "Program" must be added to the string:

```
RegQueryValueEx(..., pszDimaxisPath, ...);  
  
GetCurrentDirectory(sizeof(szCurrDir), szCurrDir);  
  
SetCurrentDirectory(pszDimaxisPath);  
  
SetCurrentDirectory("Program");  
  
  
HMODULE hPmBridge = LoadLibrary(pszPath);  
  
SetCurrentDirectory(szCurrDir);
```

Note that all the PMBRIDGE functions use calling convention `_stdcall`. Thus also programs written in other languages than C/C++ can use the library (if they are able to use C style NULL-terminated strings). An early preliminary version of the library used `_cdecl` calling conversion. This caused compatibility problems for the early PMBRIDGE interface users.

6.2 Mac OS X Operating System

The PMBRIDGE library file `pmbridge.dylib` is installed to the folder `usr/lib`.

The library calls the script `Romexis.sh` which is located in the following folder

`/Applications/Planmeca/Romexis/pmbridge`

The `Romexis.sh` starts up the ROMEXIS application.

6.3 Conflicts with DIMAXIS PMBRIDGE

The `pmbridge.dll` file that is provided on the ROMEXIS installation DVD is only meant to be used with ROMEXIS imaging application.

In the basic setup, Romexis and Dimaxis cannot be used both at the same time with PMBRIDGE.

While installing ROMEXIS, the PMBRIDGE registry information that points to the Dimaxis directory is changed to point to the ROMEXIS `pmbridge.dll` directory.

In order to use DIMAXIS one has to change the registry information to point back to the DIMAXIS `pmbridge.dll` directory.

7. Other

7.1 Example Program DxClient

There is an example program `DxClientW/DxClientOSX` available that clarifies the usage of PMBRIDGE. The example uses a wrapper class `CPmbWrap` that dynamically loads the library in the constructor and frees it in the destructor.

'Select Patient' task

Client for Dimaxis/Romexis

Patient ID: 234-45

Family Name:

First Name:

Birth Date: Gender:

Doctor:

Login User (set)

<AD-user> OR <Romexis-user>@[<pwd>]

Set User Select Patient Deselect Patient Exit

Insert Patient Update Patient Delete Patient Set Worklist

Capture Image

Panoramic.. Cephalometric.. Intraoral.. Intracam..

TWAIN ... CR Scanner... CBCT 3D

Show Dimaxis/Romexis Show Max. Move Window Select Image

Set Image Retrieval Window Size

Small Full Close

Toolbar

Hide Show

Get Image

☐ Force to 8 bits/pixel ☐ Enhance ☐ Overview Index: 0

Image File Name:

2D Image File Format

☒ TIFF ☐ JPEG ☐ BMP

Inquire Images Open Image Get Image (by Index)

Get Image (by ID)

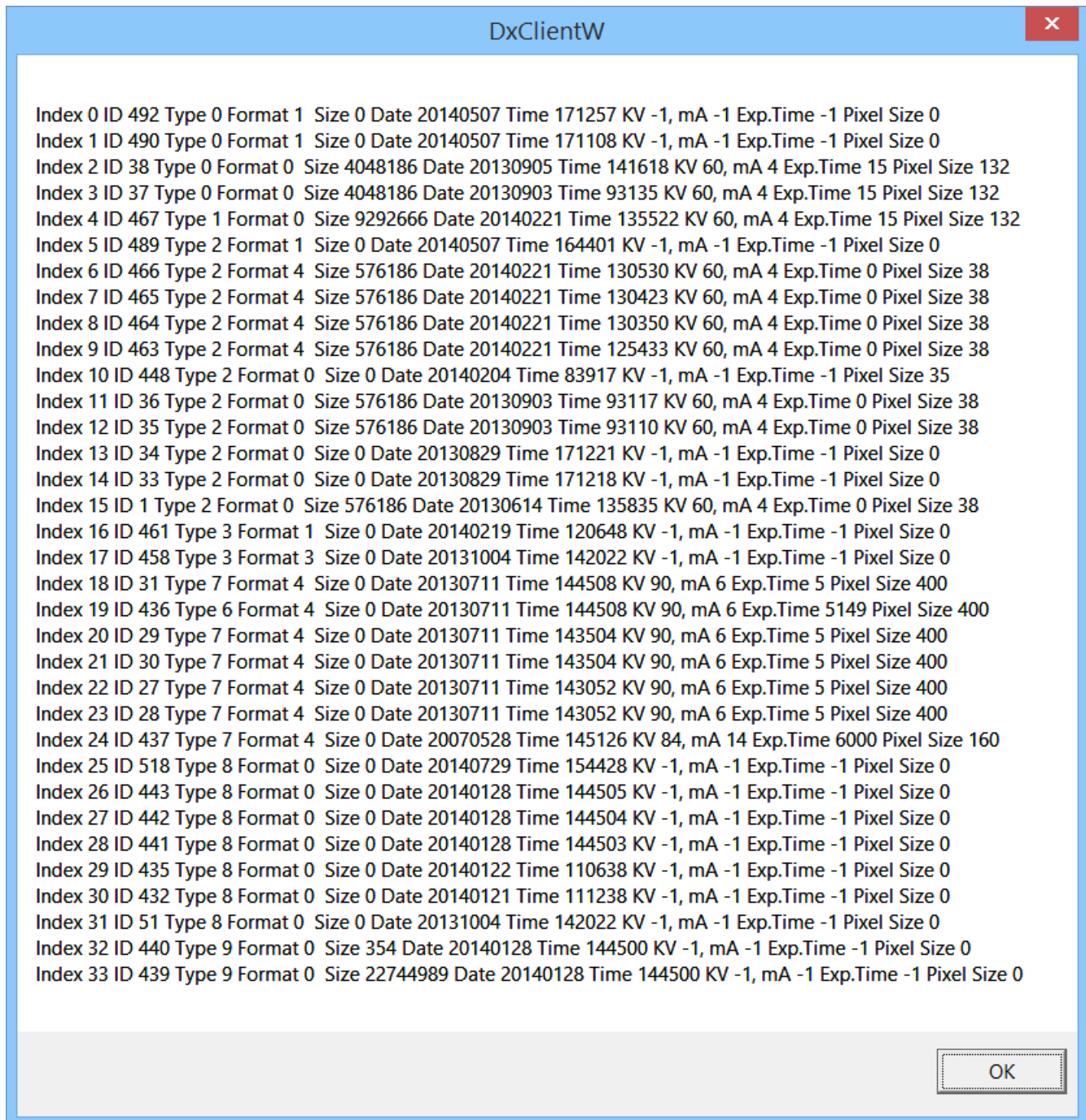
Exit

DxClientW

Return code: 0, Image Count: 34

OK

'Inquire Images' –task



Overview (=thumbnail) of image index 2 (type 0, Panoramic) using 'Get Image (by Index) –task :

Client for Dimaxis/Romexis

Patient ID: 234-45

Family Name:

First Name:

Birth Date: Gender:

Doctor:

Login User (set) <AD-user> OR <Romexis-user>@[<pwd>]

Set User Select Patient Deselect Patient Exit

Insert Patient Update Patient Delete Patient Set Worklist

Capture Image

Panoramic.. Cephalometric.. Intraoral.. Intracam..

TWAIN ... CR Scanner... CBCT 3D

Show Dimaxis/Romexis Show Max. Move Window Select Image

Set Image Retrieval Window Size Small Full Close

Toolbar Hide Show

Get Image

☐ Force to 8 bits/pixel ☐ Enhance ☒ Overview Index: 2

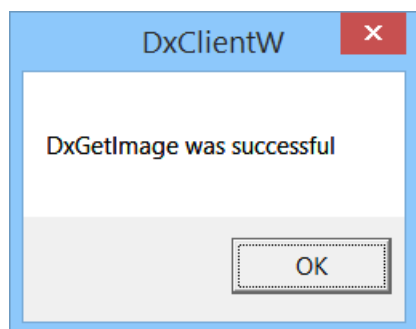
Image File Name: c:\temp\ov2

2D Image File Format ☐ TIFF ☒ JPEG ☐ BMP

Inquire Images Open Image **Get Image (by Index)**

Get Image (by ID)

Exit



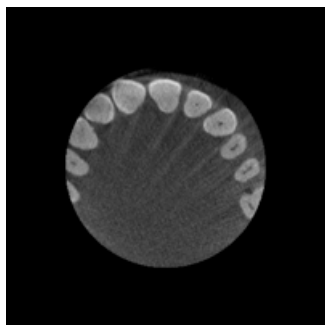
Overview of index 4 (Type 1, Cephalometric):



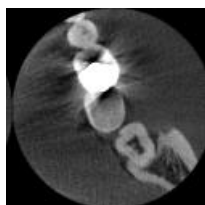
Overview of index 5 (Type 2, IntraOral):



Overview of index 19 (Type 6):



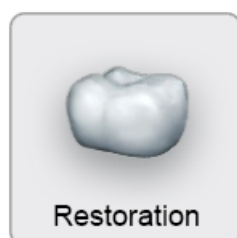
Overview of index 21 (Type 7, Volume):



Overview of index 25 (Type 8, Surface ProFace subtype):



Overview of index 26 (Type 8, Surface IOScan subtype):



Overview of index 32 (Type 9, attachment file):



2 D Images (Type 0 ... 5) whole image output format TIFF/JPEG/BMP

3D Volume Image output DCM (Dicom) format

3D Surface Image output format STL

Get volume image using 'Get Image(by Index)' of index 21:

Client for Dimaxis/Romexis

Patient ID: 234-45

Family Name:

First Name:

Birth Date: Gender:

Doctor:

Login User (set)

<AD-user> OR <Romexis-user>@[<pwd>]

Set User Select Patient Deselect Patient Exit

Insert Patient Update Patient Delete Patient Set Worklist

Capture Image

Panoramic.. Cephalometric.. Intraoral... Intracam..

TWAIN ... CR Scanner... CBCT 3D

Show Dimaxis/Romexis Show Max. Move Window Select Image

Set Image Retrieval Window Size

Small Full Close

Toolbar

Hide Show

Get Image

☐ Force to 8 bits/pixel ☐ Enhance ☐ Overview Index: 21

Image File Name: c:\temp\volume21

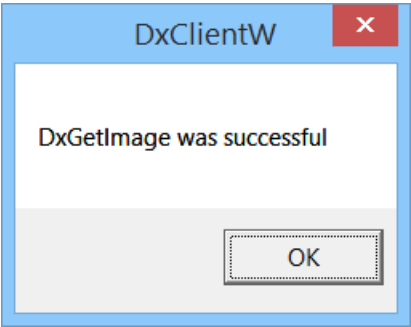
2D Image File Format

☐ TIFF ☒ JPEG ☐ BMP

Inquire Images Open Image Get Image (by Index)

Get Image (by ID)

Exit



Get surface image of index 25:

Client for Dimaxis/Romexis

Patient ID: 234-45

Family Name:

First Name:

Birth Date: Gender:

Doctor:

Login User (set)

<AD-user> OR <Romexis-user>@[<pwd>]

Set User Select Patient Deselect Patient Exit

Insert Patient Update Patient Delete Patient Set Worklist

Capture Image

Panoramic.. Cephalometric.. Intraoral... Intracam..

TWAIN ... CR Scanner... CBCT 3D

Show Dimaxis/Romexis Show Max. Move Window Select Image

Set Image Retrieval Window Size

Small Full Close

Toolbar

Hide Show

Get Image

☐ Force to 8 bits/pixel ☐ Enhance ☐ Overview Index: 25

Image File Name: c:\temp\surface25

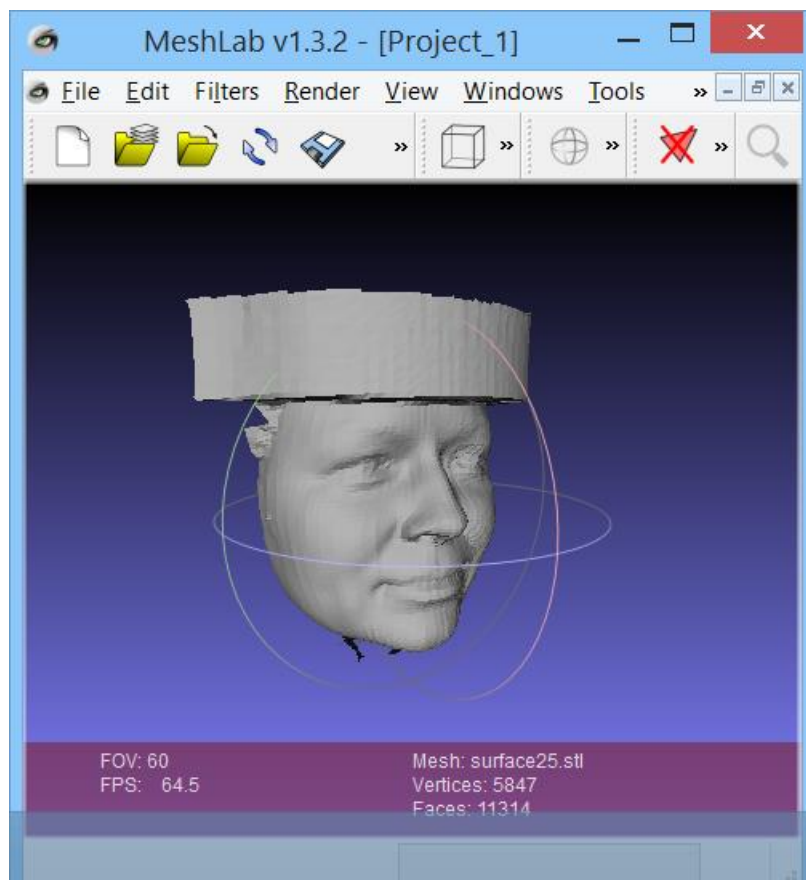
2D Image File Format

☐ TIFF ☒ JPEG ☐ BMP

Inquire Images Open Image Get Image (by Index)

Get Image (by ID)

Exit



Get attached file of index 32:

Client for Dimaxis/Romexis

Patient ID: 234-45

Family Name:

First Name:

Birth Date: Gender:

Doctor:

Login User (set)
<AD-user> OR <Romexis-user>@[<pwd>]

Set User Select Patient Deselect Patient Exit

Insert Patient Update Patient Delete Patient Set Worklist

Capture Image

Panoramic.. Cephalometric.. Intraoral... Intracam..

TWAIN ... CR Scanner... CBCT 3D

Show Dimaxis/Romexis Show Max. Move Window Select Image

Set Image Retrieval Window Size

Small Full Close

Toolbar

Hide Show

Get Image

☐ Force to 8 bits/pixel ☐ Enhance ☐ Overview Index: 32

Image File Name: c:\temp\index32

2D Image File Format

☐ TIFF ☒ JPEG ☐ BMP

Inquire Images Open Image Get Image (by Index)

Get Image (by ID)

Exit

→ result file 'INDEX32.xml' suffix of attached file read from database



7.2 Unicode Compatibility

PMBRIDGE 3.0.0 is Unicode compatible. Unicode compatible versions of `DxInsertPatient()` and `DxUpdatePatient()` are `DxInsertPatientW()` and `DxUpdatePatientW()`. Note ! If Unicode compatible functions are used, then `PatientID` and `BirthDate` variables must be ANSI strings. All other must be UNICODE strings.

However, the calling application doesn't have to be Unicode compatible to use PMBRIDGE.