QuickVision Link programmer's guide

Revision	Date	Modification	Writer
1	13/07/2006	Document creation	Y. Rekiek
2	20/06/2012	New commands EXPORT_PATIENTIMAGELIST,EXPORT_PATIENTIMAGE	Y. Rekiek
3	28/07/2014	VB sample code deleted Manifest information added New document format	F. Rey Martín
4	27/01/2017	Two new APIs	F. Rey Martín
5	21/12/2017	QuickVision does not need to run with high privileges anymore since V5.04 VDDS information added	F. Rey Martín

1. Table of contents

1. General presentation

1.1. Purpose and Scope:

The aim is to interface a dentistry database software with QuickVision or VILLA - QuickVision, adding to the database software abilities for acquiring, storing and displaying X-ray images.
To achieve this, the database software establishes a link with QuickVision and provides informations like patient name and file number.

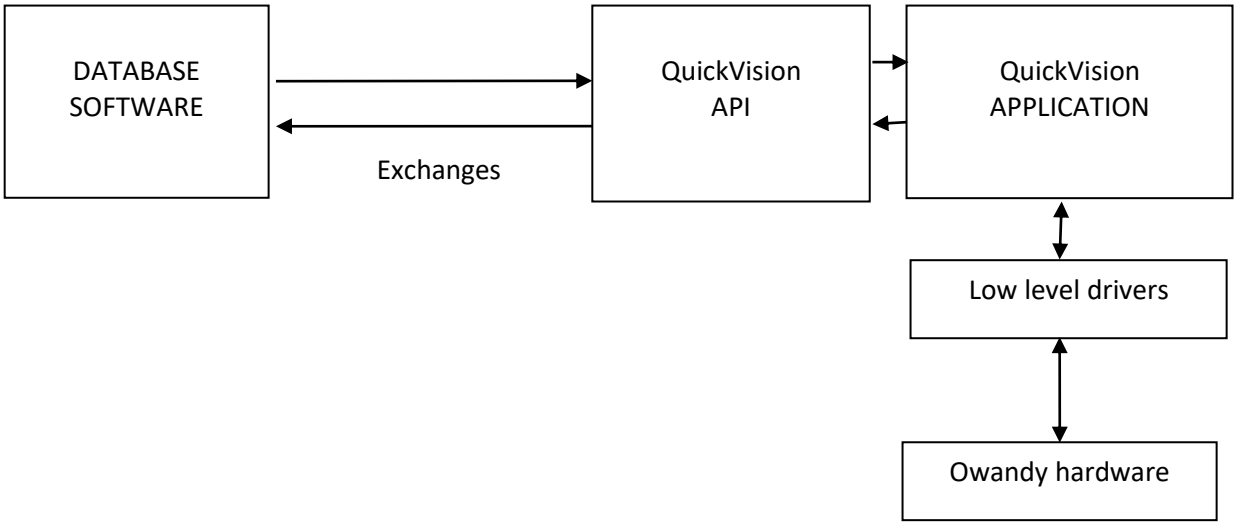
Unless specified otherwise, the word “QuickVision” in this document stands for both “QuickVision” and “VILLA – Quickvision” softwares.

1.2. Required configuration

See QuickVision manual. This document covers the supported APIs by QuickVision version 5.02 and following.
QuickVision must be installed and functional. The APIs are installed with QuickVision, there is no need to install any further software.

QuickVision versions (and therefore the APIs) older than 5.04 needed to run with high privileges (or in administration mode). To do this programmatically, you can add a manifest to your application. You will find an example in the appendix “Manifest with high privileges”.
This constraint does not exist anymore since QuickVision version 5.04.

1.3. Global view



Four different APIs exist; they are equivalent and the external database software can use any of them:

- The first one is based on message exchanges.
- The second one is based on dll communication.
- The third one is based on DOS commands.
- The fourth one is the VDDS protocol.

QuickVision is a 32 bits application, but an external database software in 32 or 64 bits can use any of the existing APIs.

The following chapter describes the way to integrate each API.

2. SDK description

2.1. SDK related files

The developer's SDK is composed by the files below.

Filename	Description
Documentation	This file
Sample\Demo folder	Source code of a C sample program that shows how to use each of the three different APIs. You can choose the API you want in API.h file.

2.2. API based on messages

For an example, please refer to API_Sig.cpp file in the Sample\Demo code.

1.1.1 Starting QuickVision

First, the external database software have to start QuickVision creating a new process (using CreateProcess, WinExec or system functions). Since the destination QuickVision folder can be choosen by the user on installation, you should retrieve the full exe name from the Windows registry. Read the "MjExec" value from the following key:

- For Owandy QuickVision: HKEY_LOCAL_MACHINE\SOFTWARE\Owandy\QuickVision
- For VILLA QuickVision: HKEY_LOCAL_MACHINE\SOFTWARE\VILLA SM\QuickVision

Since QuickVision in a 32-bit application, the external database software must use the KEY_WOW64_32KEY flag if it is a 64-bit application.

The result will be, for example, "C:\OWANDY\QuickVision\MjExec.exe"

The external database software will create a new process using the retrieved application with no parameter or the /HIDE parameter. See the details in the "Commands" chapter.

Examples:

```
system("C:\OWANDY\QuickVision\MjExec.exe")
system("C:\OWANDY\QuickVision\MjExec.exe /HIDE")
```

1.1.2 Sending commands to QuickVision

Afterwards, the external database software must retrieve the QuickVision window handle and send messages to it (refer to the "Commands" chapter).

For example:

```
HWND hWndQV = FindWindow("MjLinkWndClass", NULL);
char msg[] = "/END";
```

```
SendMessage(hWndQV, WM_SETTEXT, 0, (LPARAM)msg);
```

2.3. API based on DLL

For an example, please refer to API_Dll.cpp file in the Sample\Demo code.

1.1.3 Starting QuickVision

First, the external database software have to load the dll and retrieve the function pointers from it.
Since the destination QuickVision folder can be choosen by the user on installation, you should retrieve the full dll name from the Windows registry. Read the "MjExecDll32" value (or "MjExecDll64" value if the external database software is a 64-bit application) from the following key:

- For Owandy QuickVision: HKEY_LOCAL_MACHINE\SOFTWARE\Owandy\QuickVision
- For VILLA QuickVision: HKEY_LOCAL_MACHINE\SOFTWARE\VILLA SM\QuickVision

Since QuickVision in a 32-bit application, the external database software must use the KEY_WOW64_32KEY flag if it is a 64-bit application.

The result will be, for example, "C:\OWANDY\QuickVision\MjExec.dll"

Then, you can load the library and get the function pointers:

```
HINSTANCE hApiDll = LoadLibrary(szAPI);

typedef int (_cdecl *tMJExec_RunMJ  )(LPSTR lpszCmdLine);
typedef int (_cdecl *tMJExec_ExecCmd)(LPSTR lpszCmdLine);

tMJExec_RunMJ     pMJExec_RunMJ   = NULL;
tMJExec_ExecCmd   pMJExec_ExecCmd = NULL;

pMJExec_RunMJ  = (tMJExec_RunMJ  )GetProcAddress(hApiDll, "MJExec_RunMJ");
pMJExec_ExecCmd= (tMJExec_ExecCmd)GetProcAddress(hApiDll, "MJExec_ExecCmd");
```

The external database software will launch Quickvision using the MJExec_RunMJ function using no parameter or the /HIDE parameter (see the details in the "Commands" chapter).

Examples:

```
pMJExec_RunMJ("");
pMJExec_RunMJ("/HIDE");
```

1.1.4 Sending commands to QuickVision

Afterwards, the external database software will use the MJExec_ExecCmd function to send commands to QuickVision (refer to the "Commands" chapter).

Example:

```
pMJExec_ExecCmd("/END");
```

2.4. API based on DOS command line

For an example, please refer to API_Exe.cpp file in the Sample\Demo code.

First, the external database software have to retrieve the application to which send commands from the registry, because the destination QuickVision folder can be choosen by the user on installation. Read the "MjExec" value from the following key:

- For Owandy QuickVision: HKEY_LOCAL_MACHINE\SOFTWARE\Owandy\QuickVision
- For VILLA QuickVision: HKEY_LOCAL_MACHINE\SOFTWARE\VILLA SM\QuickVision

Since QuickVision in a 32-bit application, the external database software must use the KEY_WOW64_32KEY flag if it is a 64-bit application.

The result will be, for example, "C:\OWANDY\QuickVision\MjExec.exe"

The external database software will send commands to the retrived application using the appropriate commands as parameters (see the details in the "Commands" chapter), with the CreateProcess, WinExec or system functions.

Examples:

```
WinExec("C:\OWANDY\QuickVision\MjExec.exe")
WinExec("C:\OWANDY\QuickVision\MjExec.exe /HIDE")
WinExec("C:\OWANDY\QuickVision\MjExec.exe /END")
```

2.5. VDDS interface

QuickVision is compatible with stage 1 of the "Interface specification VDDS-media 1.4".

Refer to the VDDS specification for the detailed information.

2.6. Commands

1.1.5 HIDE

Syntax
"/HIDE"

Description
This command can only be sent when opening QuickVision. QuickVision main window will be hidden. The mouth window remains visible after having selected a patient file. It can be convenient to show only images related patient information.

Parameters
None

Returned value:
- 1 if the command succedded
- 0 if the command failed

Example
“/HIDE”

1.1.6 END

Syntax
"/END"

Description
Closes all windows and ends QuickVision.

Parameters
None.

Returned value:
- 1 if the command succedded
- 0 if the command failed

Example
“/END”

1.1.7 P

Syntax
"/P:<ID>,<LASTNAME>,<FIRSTNAME>,<SOCIALSECURITY>,<DD-MM-YYYY>,<CIV>,<STREET_LINE_1>,
<STREET_LINE_2>,<CITY>,<ZIP>,<PHONE_PERSONAL>,<PHONE_PROFESSIONAL>"

Description
Selects or creates a patient file.
If the patient file is not found, it will be created and selected.
All arriving images goes to the selected patient file.

Parameters
Mandatory fields for creating or searching a patient file:

- <ID> is a external patient identifier up to 128 alfanumeric characters. This number must be unique.

Optional fields for creating a patient file:

- <LASTNAME> is the last name of the patient (20 alphanumeric characters maximum).
- <FIRSTNAME> is the first name of the patient (20 alphanumeric characters maximum).
- <SOCIALSECURITY> is the number of insurance or identity number. Alphanumeric, unformatted. It can be 20 characters at the most.
- <DD-MM-YYYY> is the birth date. Numeric. Date format is DD-MM-YYYY.
- <CIV> is the situation of the patient:
    - 1 for Mr, 2 for Mrs, 3 for Ms, 4 for Dr. The situation will be translated to the language selected by the user in QuickVision.
    - Otherwise <CIV> can be any three alphabetic characters, but they will not be translated to the language selected by the user in QuickVision.
- <STREET_LINE_1> is the patient address. It can be 30 characters at the most.
- <STREET_LINE_2> is the complementary patient address. It can be 30 characters at the most.
- <CITY> is the patient address city. It can be 30 characters at the most.
- <ZIP> is the patient address Zip code. It can be 5 characters at the most.
- <PHONE_PERSONAL> is the patient personal telephone number. It can be 18 characters at the most.
- <PHONE_PROFESSIONAL> is the patient professional telephone number. It can be 18 characters at the most.

Remarks

- If used without parameters, "/P:" unselects the last selected patient field.
- If the <ID> exists already inside QuickVision, the patient file will be open. The other fields will not be used to search or to update the patient file.
  If the ID does not exist, the patient file will be created with the information from all the fields.
- The comma character ",' is used as separator. It cannot be used inside the fields.

Returned value:

- 0 if the command failed
- Otherwise, the internal QuickVision patient number, which is different from the external <ID>.

Examples
"/P:17,SMITH,DAVID,48XDE45,25-06-1970,1,90 3th avenue,3rd floor,NYC,10003,0612345678,0123456789"
"/P:17,SMITH,DAVID,48XDE45,25-06-1970,Prf"
"/P:17,SMITH,DAVID, , ,4"

“/P:17,SMITH,DAVID”
“/P:17”
“/P:”

1.1.8 N

Syntax
"/N:<LASTNAME>,<FIRSTNAME>,<SOCIALSECURITY>,<DD-MM-YYYY>,<CIV>,<STREET_LINE_1>,
<STREET_LINE_2>,<CITY>,<ZIP>,<PHONE_PERSONAL>,<PHONE_PROFESSIONAL>"

Description
Modifies the informations of the selected patient. If no patient file is selected, this command is invalid.

Parameters
- <LASTNAME> is the new last name of the patient (20 alphanumeric characters maximum).
- <FIRSTNAME> is the new first name of the patient (20 alphanumeric characters maximum).
- <SOCIALSECURITY> is the new number of insurance or identity number. Alphanumeric, unformatted. It can be 20 characters at the most.
- <DD-MM-YYYY> is the new birth date. Numeric. Date format is DD-MM-YYYY.
- <CIV> is the situation of the patient:
    - 1 for Mr, 2 for Mrs, 3 for Ms, 4 for Dr. The situation will be translated to the language selected by the user in QuickVision.
    - Otherwise <CIV> can be any three alphabetic characters, but they will not be translated to the language selected by the user in QuickVision.
- <STREET_LINE_1> is the new patient address. It can be 30 characters at the most.
- <STREET_LINE_2> is the new complementary patient address. It can be 30 characters at the most.
- <CITY> is the new patient address city. It can be 30 characters at the most.
- <ZIP> is the new patient address Zip code. It can be 5 characters at the most.
- <PHONE_PERSONAL> is the new patient personal telephone number. It can be 18 characters at the most.
- <PHONE_PROFESSIONAL> is the new patient professional telephone number. It can be 18 characters at the most.

Remarks:
- The comma character “,’ is used as separator. It cannot be used inside the fields.
- To keep the QuickVision current value of one of the fields, use an empty string.
  For example, “/N:SMITH,MONICA,,31-12-1950” will:
    - update the last name, first name and birthday,
    - keep the insurance number, situation, address, city, Zip and phone numbers already existing in QuickVision.
- To empty one of the fields, use the value “<empty>”.
  For example, “/N:SMITH,MONICA,<empty>,31-12-1950” will:
    - update the last name, first name and birthday,
    - empty the insurance number,
    - keep the situation, address, city, Zip and phone numbers already existing in QuickVision.

Returned value:
- 0 if the command failed

- Otherwise, the internal QuickVision patient number

Examples
“/N:SMITH,MONICA”
“/N:SMITH,MONICA,48XDE45,25-06-1970,2,90 3th avenue,3rd floor,NYC,10003,0612345678,0123456789”
“/N:SMITH,MONICA,,31-12-1950”
“/N:SMITH,MONICA,<empty>,31-12-1950”

1.1.9 D

Syntax
"/D:<ID>"

Description
Deletes a patient file and all its images.

Parameters
- <ID> is the unique external patient identifier that was used in the call /P:.

Returned value:
- 1 if the command succedded
- 0 if the command failed

Example
“/D:17”

1.1.10 T

Syntax
"/T:<num>"

Description
Selects a tooth. This is intended to prepare image acquisition. Any acquired image is assigned to the selected tooth.

Parameters
<num> is a tooth number in the FDI (Fédération Dentaire Internationale) World Dental Federation ISO-3950 notation (see Appendix “FDI two-digit notation”.

Returned value:
- 1 if the command succedded
- 0 if the command failed

Example
“/T:21”

1.1.11 X

Syntax
"/X"

Description
Opens the x-rays image window for the last selected tooth of the last selected patient.

Parameters
None.

Returned value:
- 1 if the command succedded
- 0 if the command failed

Example
“/X”

1.1.12 CFG

Syntax
"/CFG"

Description
Shows the QuickVision configuration dialog. The user can select a sensor and define standard paths.

Parameters
None.

Returned value:
- 1 if the command succedded
- 0 if the command failed

Example
“/CFG”

1.1.13 O

Syntax
"/O:< flag1>,< flag2>,< flag3>,< flag4>,< flag5>,< flag6>,< flag7>"

Description
This command sets various preferences flags.
All flags can take only 2 values: ascii char "0" or "1".

Parameters
- <flag 1>: "Retain image adjustments"
- <flag 2>: "Beep at acquisition"
- <flag 3>: "Propose to delete"
- <flag 4>: "Rotate image from upper arcade"
- <flag 5>: "Rotate image from lower arcade"
- <flag 6>: "Flip image from left arcade"
- <flag 7>: "Flip image from right arcade"

Returned value:
- 1 if the command succedded
- 0 if the command failed

Example
“/O:0,1,1,0,0,0,0”

1.1.14 EXPORT_PATIENTIMAGE_LIST

Syntax
"/EXPORT_PATIENTIMAGE_LIST:<pathfilename>"

Description
This keyword is intended to retrieve a list of the current patient's images. It creates the <pathfilename> text file in CSV format containing a list of all current patient's images. That csv file can be opened by Microsoft Excel.

The format of the CSV file is as follows:

- Character encoding: ANSI (Windows 1252).
- Separator:Semicolon ";".
- All fields are double quoted.
- The first line is a header (always present) which contains column names of the fields:
  "Image type";"Image subtype";"Patient #";"Image #";"Date";"Hour";"Tooth"
- Next, follows lines (records) where each line describes an image. In one line, the fields are:

  Image type: 1 character fixed length from:
  "I" for intraorals,
  "P" for panoramics,
  "C" for camera images,
  "V" for volumes.

  Image subtype: additional information about the image subtype (e.g. "TMJ biaxial" or "3D maxillary". It can be blank.

  Patient #: variable length.
  The patient number in the internal QuickVision database.
  Only this data can be used to retrieve individual images using the command /EXPORT_PATIENTIMAGE.

  Image #: variable length.
  The image unique number. Only this data can be used to retrieve individual images.

  Date: 8 characters
  Date of image creation. Format is yyyymmdd (without separator)

  Hour: 6 characters
  Hour of image creation. Format is hhmmss (without separator)

  Tooth: 2 characters.
  The tooth number associated to the image, in international notation (ISO-3950 notation. See appendix).

Parameters

- <pathfilename> is the full path name of the csv file. If not specified, it is set to default: <FOLDER>\QVImageList.csv, where <FOLDER> is the temporary folder given by Windows when calling the function GetTempPath.

Returned value:

- 1 if the command succedded
- 0 if the command failed

Examples

"/EXPORT_PATIENTIMAGE_LIST:C:\MyFile.csv"
"/EXPORT_PATIENTIMAGE_LIST:"

1.1.15 EXPORT_PATIENTIMAGE

Syntax
"/EXPORT_PATIENTIMAGE:<PatientNumber>,<ImageNumber>,<ImageType>,<FilterOption>,<PathFilename>"

Description
This command exports the specified QUICKVISION image (panoramic, intra-oral or camera) to the specified file "PathFilename" in BMP format. 3D export is not currently supported.

Parameters
- <PatientNumber>: the internal QuickVision patient number from the previously returned csv file (variable length).
- <ImageNumber>: the image number from the previously returned csv file (variable length).
- <ImageType>: the image type from the previously returned csv file (1 character).
- <FilterOption>: image filtering (sharpening) option flag.
  If the image is wanted filtered, the value must be "Fn", where "n" if a number from 0 to 100.
  if "n" is 0, the applied filter strength is the one chosen in QUICKVISION.
  if "n" is between 1 and 100, it is the filter strength that will be applied to this one image.
  If the image is wanted unfiltered, the value must be “0”
- <PathFilename>: the path and file name of the resulting bmp file.

Returned value:
- 1 if the command succedded
- 0 if the command failed

Example
To export the panoramic image “003” for the patient 10 filtered with a gain=9 to the folder C:\Folder\MyImage.bmp
“/EXPORT_PATIENTIMAGE:10,003,P,F9,C:\Folder\MyImage.bmp”

To export the intraoral “051” for the patient 10 unfiltered to the default export file
“/EXPORT_PATIENTIMAGE:10,051,I,0”

3. Appendix

3.1. Alternate database commands

QuickVision is able to manage two different databases, for example a default database stored on the local hard disk and a second on a network server. Here are the steps to create a double database environment.

Firstly, create a text file named "jwaltdb.ini " in QuickVision folder, then edit it with notepad.

Put the following lines into it:
[AltDbase]
UseAltDbase=1
PatientPath=E:\JULIEW\
XrayPath=E:\ MYXPLACE\
ImagesPath=E:\MYXPLACE\
3DPath=E:\MYXPLACE\
ArchXrayPath=
ArchImgPath=

To activate the alternate database, use this flag:
UseAltDbase=1
set to 0 to disable the alternate db, or 1 to enable it.

The next fields are self explanatory: they indicate the location of the alternate database.
You must provide valid paths, otherwise the result will be unpredictable.
In this example, we store the second data patient file on the E drive, all images on a directory called "MYXPLACE" on the same drive E.

Then run QUICKVISION. To switch to the alternate database, the user has to do a CTRL-RIGHT CLICK onto the name/date bar of QUICKVISION. (on top, just under the menu bar).

QUICKVISION will display an asterisk following its name, indicating we are on the alternate database.
Another CTRL - RIGHT CLICK returns to the main database, and the asterisk disappears.

Paths can be controlled not only by the "jwaltdb.ini" file, but also dynamically by the commands below:

"/SET_PATIENT_PATH_TO: <path>"
"/SET_XRAY_PATH_TO: <path>"
"/SET_IMG_PATH_TO: <path>"
"/SET_3D_PATH_TO: <path>"
"/SET_ARCH_XRAY_PATH_TO: <path>"
"/SET_ARCH_IMG_PATH_TO: <path>"

<path> must be a valid MS Windows path in the form: "<Drive letter>:\Folder\"

As explained, they change dynamically the current disk and/or path.
These commands do not change default pathes. They change only alternative paths.

3.2. FDI two-digit notation

Adults:
In the FDI (Fédération Dentaire Internationale) World Dental Federation ISO-3950 notation 1s are central incisors, 2s are laterals, 3s are canines, 4s are 1st premolars etc., up through 8s which are 3rd molars.
The permanent teeth quadrants are designated 1 to 4 such that 1 is upper right, 2 is upper left, 3 is lower left and 4 is lower right, with the resulting tooth identification a two-digit combination of the quadrant and tooth (e.g. the upper right central incisor is 11 and the left is 21).
The lower left permanent first molar is 36; however, it is not said thirty-six, but rather three six. 11 is one one, not eleven.
The currently accepted convention to view the FDI notation chart is from the perspective of the patient's right on the left:

Permanent Teeth

FDI Two-Digit Notation

			upper right								upper left				
18	17	16	15	14	13	12	11	21	22	23	24	25	26	27	28
48	47	46	45	44	43	42	41	31	32	33	34	35	36	37	38
			lower right								lower left				

Baby Teeth*

FDI Two-Digit Notation

			upper right								upper left	
			55	54	53	52	51	61	62	63	64	65
			85	84	83	82	81	71	72	73	74	75
			lower right								lower left	

Children:
In the deciduous dentition the numbering is correspondingly similar except that the quadrants are designated 5, 6, 7 and 8.

3.3. Manifest with high privileges

YourApp.manifest file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
  <security>
    <requestedPrivileges>
     <requestedExecutionLevel
      level="highestAvailable"
      uiAccess="false"/>
    </requestedPrivileges>
   </security>
 </trustInfo>
</assembly>
```

4. Troubleshooting

4.1. How to check if QuickVision is installed?

Test if the following registry key entry exists:

- For Owandy QuickVision: HKEY_LOCAL_MACHINE\SOFTWARE\Owandy\QuickVision
- For VILLA QuickVision: HKEY_LOCAL_MACHINE\SOFTWARE\VILLA SM\QuickVision

Since QuickVision in a 32-bit application, the external database software must use the KEY_WOW64_32KEY flag if it is a 64-bit application.

4.2. Is the target application a single instance or a multiple instance?

QuickVision is a single instance application.

End of the document