VisiQuick link DLL specifications

Description

VisiQuick can be linked to in many ways. Using the link DLL is the most flexible method.
The link DLL is normally installed by the VisiQuick install technician, into the same folder as the exe of your app.
Local path

- C:\Program Files\YourApp

Network path

- \\Server\Share\YourApp

Alternatively, you may choose to place the link DLL in a separate folder, however, that requires you to add a configuration field to your app, to allow the installer to indicate where the DLL is located.

The DLL supports **1-way** or **2-way** linking. 2-way linking allows for sending events from VisiQuick back to your app, for instance, to auto-add treatments in your app, when VisiQuick acquires a new x-ray, or to receive image data from VisiQuick, that you use to display an image in your app.

For support, email to support@thomsystems.com, or call +31-20-6722610.

Functions in the DLL

The DLL is a WIN32 x86 dll.
All functions are declared with WINAPI (stdcall).
All strings are zero-terminated ANSICODE.

```
WINAPI int VISI_COMMAND(char* Cmd)
```

- Sends a command to an already running VisiQuick. This is the main method for communicating with VisiQuick.
- Returns error code.

```
WINAPI int VISI_GETERRORSTRBUF(char* Buf, int BufLen)
```

- Fills a buffer with more error information, or with a result string.
- Returns error code.

```
WINAPI char* VISI_GETERRORSTR(void)
```

- Returns a pointer to a string with more error information, or with results.

```
WINAPI int VISI_EXIT(void)
```

- Shuts down VisiQuick and performs cleanup.
- Returns error code.

```
WINAPI int VISI_INIT2W(char* IniP, PROCMSGS2W Pm2W, HWND PrevWnd)
```

- Initializes the DLL and launches VisiQuick.
    - IniP = path to vqdde32.ini. Can be set to null, then vqdde32.ini will be located in the same folder as vqdde32.dll.
    - Pm2W = function pointer, pointing to static function in your app. This function will receive events, for 2-way linking. Pm2W can be null.
    - PrevWnd = Window Handle to the Window VisiQuick will return to, when the green Back-button is clicked. This is normally the handle to your main window. PrevWnd can be null; the active window is then used.
- This function allows for 2-way linking.
- Returns error code.

```
WINAPI int VISI_INITI(char* IniP, PROCMSGS Pm, HWND PrevWnd)
```

- Initializes the DLL and launches VisiQuick.
    - IniP = path to vqdde32.ini. Can be set to null, then vqdde32.ini will be located in the same folder as vqdde32.dll.

    - Pm = function pointer, pointing to static function in your app. This function will receive events. Pm can be null.
    - PrevWnd = Window Handle to the Window VisiQuick will return to, when the green Back-button is clicked. Normally the handle to your main window. PrevWnd can be null; the active window is then used.
- This function allows for 1-way linking.
- Returns error code.

```
WINAPI int VISI_INIT(PROCMSGS Pm)
```

- Initializes the DLL and launches VisiQuick.
    - Pm = function pointer, pointing to static function in your app. This function will receive events. Pm can be null.
- This function allows for 1-way linking.
- Returns error code.

```
WINAPI int VISI_GETVER(void)
```

- Returns DLL internal version number. 123 = version 1.23.
- This function is seldomly used.

Error codes

- >= 0 = no error
- -1000 = Unable to launch VisiQuick
- -1001 = Error
- -1002 = Busy
- -1003 = Timeout
- -1004 = Missing seperator
- -1005 = Bad res number

Callback functions

```
typedef void WINAPI (*PROCMSGS)(void);
typedef int WINAPI (*PROCMSGS2W)(int Msg, int WParam, int LParam);
```

App start

- Do nothing

Link

- If not loaded
    - Load DLL
    - Call VISI_INIT2W
        - (VisiQuick is launched)
- Build command string
    - If opening patient chart (getting teeth availability data)
        - Only when your app wants to indicate teeth availability data.
        - `%VQCMD%=searchtstatus|100~PatID`
        - Error string will indicate teeth availability data.
    - If clicking linking button
        - `String=”%VQCMD%=setreturnwnd|11~HWND”{CR-LF} +`
        - `“%VQCMD%=newpat|100~PatientID|101~FirstName|102~LastName|103~DateOfBirth”`
- Call VISI_COMMAND(String)
- If error (optional, can skip)
    - Call VISI_GETERRORSTR
    - Display error

App end

- If loaded

- Call VISI_EXIT
  - VisiQuick is ended
- Unload DLL

Notes

- Is a DentistID field required?
  - VisiQuick can handle multiple patient lists, wherein each patient list can be physically seperated, and each list contains unique PatientIDs.
  - If there is only one patient list in your app, do not specify a DentistID.
  - If nothing is specified, the default DentistID is 1 (32-bit number).
  - If your app needs to specify a particular DentistID, use field 50
    - `|50~DentistID`
    - Example for newpat (1 string)
      - `%VQCMD%=newpat|50~DentistID|100~PatientID|…`
  - All commands with a PatientID also requires a DentistID if specific DentistIDs are used.
- 2-way linking
  - VisiQuick sends an event to your app when a new x-ray image is saved. For instance, your app can decide to automatically create a new entry in the list of treatments.
  - The static function `PROCMSGS2W` in your app is called from the DLL when VisiQuick sends an event. Msg contains an event code, and WParam, LParam contains event data.
- {CR-LF} = carriage return and line feed. Seperates one command from the next.
- DateOfBirth must be formatted as configured in the current Windows Regional Settings. Specifically, a date must be formatted just like explorer.exe displays file dates in Detail View.
- Setreturnwnd is used to pass the Window Handle, to which VisiQuick will return to, when the Green Back-button is clicked.
- When linking to VisiQuick, all fields are automatically overwritten, except for PatientID.
- There are more fields your app can pass:
  - 104~Gender (M/F)
  - 105~Street
  - 106~ZipCode
  - 107~City
  - 108~Country
  - 109~Telephone
  - 110~Fax
  - 111~Email
  - 112~Notes (use ^ to separate multiple lines)
  - 113~SocialSecurityNumber
  - 114~SourcePatID (not used)
  - 115~OrderNumber
  - 116~Diagnose (use ^ to separate multiple lines)
  - 117~PatientGuid (not used)
  - 118~ReadOnlySSN (as field 113, but readonly, and optionally checked with an 11-test as well)

Command strings

- openpat
  - Opens existing patient, and opens indicated image ID.
  - `%VQCMD%=openpat|100~PatientID|200~ImageID`
  - Error if PatientID or ImageID does not exist.
- closepat
  - Closes current patient.
  - `%VQCMD%=closepat`
  - VisiQuick closes the patient and clears all views.
- searchphotos
  - Multiple purpose image searching function.
    - Searches for, and optionally opens, specific types of images.
    - Can cause VisiQuick to switch to a floating mini-mode, displaying a single tooth image, or two bitewings, etc.

        - Can cause VisiQuick to switch to a particular viewing mode, and the display specific types of images (bitewings, PAN, etc), all on a secondary monitor. Purpose: click on specific icon in your app, on primary monitor, and VisiQuick display specific images on the second monitor.
        - Can return a list of image IDs.
    - `%VQCMD%=searchphotos|100~PatID|10~Flags|11~Index|201~SearchText`
    - Flags & spf_onlycheck == 0
        - Not only search, but also actually opens the found images.
    - Flags & spf_onlycheck == 1
        - Only search.
    - Index == View to switch to in VisiQuick.
        - spi_Bitewings
        - spi_CompareView
        - spi_XrayView
        - spi_ColorView
        - spi_PanView
        - spi_CephView
        - spi_CtView
        - spi_DocView
        - spi_FileView
        - spi_IrsProView
- searchphotoids
    - Searches for FDI tooth codes, or for custom tooth codes, and returns list of matching ImageIDs.
    - `%VQCMD%=searchphotoids|100~PatID|10~Flags|11~Index|201~SearchText`
    - Flags must be spis_toothpos if Index = '0'. For Index='1' it is ignored.
    - Index can be '0' (toothpos in searchtext) or '1' (all images)). Default is '0'.
- searchtstatus
    - Returns teeth availability string.
    - `%VQCMD%=searchtstatus|100~PatID|11~Index`
    - Index must be 1.
    - The availability string is retreived with VISI_GETERRORSTR.
    - Each character indicates how many images there are for a tooth or modality.
        - First 32 characters = 32 teeth (11-18,21-28,31-38,41-48).
        - 1 = PAN
        - 1 = CEPH
        - 1 = BWR1
        - 1 = BWL1
        - 1 = BWR2
        - 1 = BWL2
        - 1 = Pas (passport image)
        - 1 = ColR (color right)
        - 1 = ColM (color middle)
        - 1 = ColL (color left)
        - 1 = OcclU (occlusal upper)
        - 1 = OcclL (occlusal lower)
- newpat
    - Main linking function.
    - Creates and opens patient. If patient already exists, overwrites all fields, except PatientID.
    - `%VQCMD%=newpat|100~PatID|101~FirstName|102~LastName etc.`
    - Minimum fields
        - `100~PatID`
        - `101~FirstName`
        - `102~LastName`
        - `103~DateOfBirth`
    - You should pass all available patient fields, because when printing images in VisiQuick, the patient data is also printed, with configurable fields.
    - See note below about the format of a date field.
- newphoto
    - Creates and opens patient. If patient already exists, overwrites all fields, except PatientID.

    - `%VQCMD%=newphoto|100~PatID|101~FirstName|102~LastName|201~ToothCode(s)|600~0`
    - All fields are as for command “newpat”, with field 201 added.
    - Initiates acquiring new x-ray on the specified tooth (Modality).
    - Allowed values for field 600 (XrayIndex) are '0' , '1', '2' and '3'. This indicates the index of the X-raybutton (default=0).
- updatepat
    - Not used.
- dialpat
    - Cause VisiQuick to use Windows Dialler to call the telephone number of the patient.
    - `%VQCMD%=dialpat|100~PatID`
- dialnumber
    - Cause VisiQuick to use Windows Dialler to call the specified telephone number.
    - `%VQCMD%=dialnumber|109~020-6722610`
- worklink
    - Not used.
- getimage
    - This function will cause VisiQuick to send back image data to your app.
        - Can be used to display a passport photo of the patient.
        - Can be used to display the newest PAN x-ray.
        - Can be used to display the most recent bitewings.
        - Can be used to display popup images with each tooth.
    - `%VQCMD%=getimage|100~PatID|10~Flags|11~Index|12~HWNDorHDC|201~SearchText`
    - Flags & gif_photoid == 0
        - Search using exact ImageID
    - Flags & gif_photoid != 0
        - Search using FDI code
    - Index == gii_paintrect
        - Paint in window rect in HWND
    - Index == gii_paintrectdc
        - Paint in rect in HDC
    - Index == gii_paintclient
        - Paint in client rect in HWND
    - Index == gii_copydata
        - Send WM_COPYDATA with HGLOBAL with DIB.
    - HDC/HWND
        - DC or HWND to paint in
    - SearchText
        - ImageID or FDI code to search for.
        - FDI codes
            - Bitewing L/R = bwl/bwr
            - PAN = pan
            - CEPH = ceph
            - Passport image = pas
- selectimages
    - This function cause VisiQuick to activate, allowing the user to select one or more images by dragging into a selection popup window. The IDs of the selected images are then sent back in an event. The function getimage can then be used for each ID, to get a copy of the image, uncompressed or as a jpeg.
        - Can be used to get images to add as attachments to an email in your app.
        - Can be used to add images to a print layout in your app.
    - `%VQCMD%=selectimage|100~PatID|12~HWND|13~Prompt`
    - HWND = Window handle to return to, when done selecting.
    - Prompt = Text to display in selection function.
- setadminlinkdllpath
    - Not used.
    - `%VQCMD%=setadminlinkdllpath|12~dllpath`
- setreturnwnd
    - Sets the Window Handle that VisiQuick activates, when the green Back-button is clicked.
    - `%VQCMD%=setreturnwnd|11~hwnd_dec_or$hex`

    - Field 11
        - HWND of window.
        - Decimal = 12345
        - Hex = $12345
- quit
    - Cause VisiQuick to terminate.
    - `%VQCMD%=quit|10~Flags`
    - Flags not used.
- getxrayinfo
    - Retrieve semi-colon separated list of x-ray module names
    - `%VQCMD%=getxrayinfo`