



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 1 van 13

Introduction

This document is intended for developers of Practice Management Systems (PMS) and describes the mechanism of linking from a PMS (your app) to VisiQuick, opening the same patient.

Note

It is not allowed to transfer or redistribute any part of this SDK to a 3rd party, except as defined in this SDK, without prior approval from Thomas Monitor Systems. If you have any questions regarding your rights to use this SDK, please contact Thomas Monitor Systems by sending email to info@thomsystems.com.

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 installation technician, into the same folder as the executable of your app. There are 32-bit and 64-bit versions of the link DLL. The name of the link DLL is "vqdde32.dll" or "vqdde64.dll".

Alternatively, the link DLL can be placed in a separate folder, however, that requires a configuration field to your app, to allow configuring the location of the link DLL.

Several link DLL sample applications are available for various platform to help getting started. Contact us for more information if needed.

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, when VisiQuick acquires new x-rays, or to receive image data from VisiQuick, that you use to display an image in your app.

For support, email to info@thomsystems.com.

Published functions

All functions are declared with WINAPI (std_call). All strings are zero-terminated ANSICODE.

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

Sends a command to an already running VisiQuick and returns a result code. This is the main method for communicating with VisiQuick.

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

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



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 2 van 13

Returns an 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 an error code.

```
WINAPI int VISI_INIT2W(char* IniP, PROCMSG2W 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 an error code.

```
WINAPI int VISI_INITI(char* IniP, PROCMSG 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(PROCMSG 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 an error code.



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 3 van 13

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

Returns DLL internal version number. 123 = version 1.23.

This function is seldomly used.

Result codes

- ≥ 0 = Successful
- -1000 = Unable to launch VisiQuick
- -1001 = Error
- -1002 = Busy
- -1003 = Timeout
- -1004 = Missing separator
- -1005 = Bad res number

Callback functions

The callback functions are only required if your app is implementing 2-way linking.

```
typedef void WINAPI (*PROCMSG) (void);  
typedef int WINAPI (*PROCMSG2W) (int Msg, int WParam, int LParam);
```

Using enhanced security (User-Roles)

If VisiQuick is configured for enhanced security, VisiQuick requires a RoleID before any patient can be opened. The RoleID is an internal ID in VisiQuick. The RoleID is set via one of the following methods:

- Via the link.
- Via Active Directory.
- Via a default selection.

When the method for setting the RoleID is via the link, every link command using a PatientID must also include a so-called ExtRoleID. The ExtRoleID value is determined outside VisiQuick, in your app. VisiQuick will use this ID and map it to VisiQuick internal RoleID. This is configured in VisiQuick User-Roles settings. Passing the ExtRoleID parameter is described below with the link command NewPat on page 7.

When advanced security requires logging the user selected in your app, it is required to pass the name of that user via the link. Passing the LinkUser parameter is described below with the link command NewPat on page 7.

Calling sequence

App start

Do nothing.



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 4 van 13

App open patient

If your app requires displaying which images are available for each tooth, use command SearchTStatus, and then indicate, for each tooth, if images are available, for instance using color.

App close patient

Use the command ClosePat to also close the patient in VisiQuick. If the user is using two monitors and VisiQuick is on the second monitor, the previous patient should not be visible there.

App switch patient

Use the same code as for open patient. The reason for this requirement is that if VisiQuick is running on another monitor than your app, VisiQuick can automatically open the same patient when your app is switching patient (in this case, done when SearchTStatus is executed).

App click link function

If the link DLL is not loaded:

- Load the link DLL (LoadLibrary).
- Retrieve procedure addresses (GetProcAddress).
- Call VISI_INIT2W (VisiQuick is launched).

If the link DLL is already loaded, do nothing.

Build command string:

- If opening patient chart (getting teeth availability data):
 - Only when your app wants to indicate teeth availability data
 - Command string: %VQCMD%=searchtstatus|100~PatID
 - Error string (using VISI_GETERRORSTR) will indicate teeth availability data.
- If clicking linking button in your app:
 - Command string: %VQCMD%=setreturnwnd|11~HWND{CR-LF}%VQCMD%=newpat|100~PatientID|101~FirstName|102~LastName|103~DateOfBirth

Execute Result=VISI_COMMAND(Command string)

If Result>=0 (optional) Execute ErrorString=VISI_GETERRORSTR(), Display error.

App end

If the link DLL is loaded:

- Call VISI_EXIT (VisiQuick is ended).
- Unload the link DLL.



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 5 van 13

Notes

Is a ClinicID field required?

VisiQuick can handle multiple patient lists, wherein each patient list can be physically separated, and each list contains unique PatientIDs, for installations where more than one dentist is using the database where each dentist has their own list of patients. A dentist is a Clinic.

If there is only one patient list in your app, specifying a ClinicID is not required.

If nothing is specified, the default ClinicID is 1 (32-bit number).

If your app needs to specify a particular Clinic, use field 50 when linking:

- |50~ClinicID

Example for NewPat:

- Command string: %VQCMD%=newpat|50~ClinicID|100~PatientID|...

All commands with a PatientID also requires a ClinicID if specific ClinicIDs are used.

2-way linking

VisiQuick sends an event to your app when a new X-ray image is saved, together with modality information. For instance, your app can decide to automatically create a new entry in the list of treatments.

The static function PROCMSG2W 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.

Separates one command from the next.

Date of Birth

This parameter 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. It is possible to force using another date format by adding an entry in VisiQuick.ini. Contact us for more information is needed.

SetReturnWnd

This 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 patient fields are automatically overwritten, except for PatientID.

There are more fields your app can pass:



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 6 van 13

Field	Description	Since
40~ExtRoleID	The ID (integer) of the Role of the user in your app.	3.0.1.???
50~ClinicID	The ID (integer) of the Clinic that should be opened (where PatientID is defined).	3.0.1.1
100~PatientID	The ID (string) of the patient that should be opened.	3.0.1.1
101~FirstName	The first name of the patient.	3.0.1.1
102~LastName	The last name of the patient.	3.0.1.1
103~DateOfBirth	The date of birth of the patient. Must be in the format Windows is currently using.	3.0.1.1
104~Gender	M/F	3.0.1.1
105~Street	The street address of the patient.	3.0.1.1
106~ZipCode	The zip code of the patient.	3.0.1.1
107~City	The city name of the patient.	3.0.1.1
108~Country	The country name of the patient.	3.0.1.1
109~Telephone	The telephone number of the patient.	3.0.1.1
110~Fax	The fax number of the patient.	3.0.1.1
111~Email	The e-mail address of the patient.	3.0.1.1
112~Notes	The notes text of the patient. Use ^ to separate multiple lines.	3.0.1.1
113~SocialSecurityNumber	The SSN of the patient.	3.0.1.1
114~SourcePatID	The ID of the patient in the original software, used in converted databases.	3.0.1.1
115~OrderNumber	The order number of the patient. Used when dental labs are exchanging data.	3.0.1.1
116~Diagnose	The diagnose text of the patient. Use ^ to separate multiple lines.	3.0.1.1
117~PatientGuid	The GUID of the patient. Not used.	



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 7 van 13

118~ReadOnlySSN	The SSN of the patient. If this field is used instead of field 113, it cannot be modified in VisiQuick (read-only, and optionally validated with an 11-test as well, configured in VisiQuick).	
119~LinkUser	The ID of the user within your app.	3.0.1.795

Command strings

This is the list of all commands that can be sent via VISI_COMMAND. Normally, only the NewPat command is used.

Command	Description	Since
OpenPat	<p>Opens an existing patient, and opens indicated image ID. This command is normally not used (use NewPat instead).</p> <p>Command string: %VQCMD%=openpat 100~PatientID 200~ImageID</p> <ul style="list-style-type: none"> ○ Optional parameter 50: the ID of the Clinic to which the referred patients belong. ○ Parameter 100: the ID of the patient to open. ○ Parameter 200: the ID of the image to open. <p>Error if PatientID or ImageID does not exist.</p>	3.0.1.1
ClosePat	<p>Closes the current patient. Can be called when your app is closing the view of the patient.</p> <p>Command string: %VQCMD%=closepat</p> <ul style="list-style-type: none"> ○ (no parameters) <p>VisiQuick closes the patient and clears all views.</p>	3.0.1.1
NewPat	<p>Main linking function. Creates (if needed) and opens the patient. If the patient already exists, all fields are overwritten, except PatientID.</p> <p>Command string: %VQCMD%=newpat 100~PatID 101~FirstName 102~LastName etc.</p> <p>Parameters:</p> <ul style="list-style-type: none"> ○ Optional parameter 50: the ID of the Clinic to which the referred patients belong. 	



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 8 van 13

	<ul style="list-style-type: none"> ○ Parameter 100: the ID of the patient to open. ○ Parameter 101: the patient's first name. ○ Parameter 102: the patient's last name. ○ Parameter 103: the patient's date of birth. <p>Required fields if using enhanced security</p> <ul style="list-style-type: none"> ○ Parameter 40: ExtRoleID (the ID of the role in your app, for instance PowerUser). ○ Parameter 119: LinkUser (the user name or ID of the selected user in your app, when your app has user-selection) <p>You should pass all available patient fields, because when printing, mailing or exporting images from VisiQuick, the patient data is also included, with configurable fields.</p> <p>See note below about the format of a date field.</p>	
<p>NewPhoto</p>	<p>Creates and opens the patient. If the patient already exists, the function overwrites all fields, except PatientID. Then it initiates acquisition of a new X-ray on the specified tooth (Modality).</p> <p>Command string: <code>%VQCMD%=newphoto 100~PatID 101~FirstName 102~LastName 201~ToothCode(s) 600~0</code></p> <p>All fields are as for command NewPat, with field 201 added.</p> <ul style="list-style-type: none"> ○ Parameter 600 (X-ray Index): 0-3, specifying the X-ray profile to activate (default=0=the yellow X-ray button). 	<p>3.0.1.1</p>
<p>MergePat</p>	<p>Merges one or more patients to a patient. The used IDs cannot be in use. If the function fails because data is in use, the function should be retried after the locked data has been freed.</p> <p>Command string: <code>%VQCMD%=mergepat 100~TargetPatID 301~SourcePatientIDs 302~Action</code></p> <ul style="list-style-type: none"> ○ Optional parameter 50: the ID of the Clinic to which the referred patients belong. ○ Parameter 100: the ID of the patient to which all the other patients must be merged into. If it does not exist, it is 	<p>3.0.1.???</p>



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 9 van 13

	<p>automatically created, and its data will be filled with data of the first source patient.</p> <ul style="list-style-type: none"> ○ Parameter 301: a list of IDs of the other patients which should be merged. The IDs in the list are separated using ‘;’ as in this example: 1111;2222;3333. ○ Parameter 302: a value describing the type of operation the command should execute. <ul style="list-style-type: none"> ▪ 0: (not used) ▪ 1: Move <p>Returns a result code depending on the function:</p> <ul style="list-style-type: none"> ○ 0: Operation succeeded. All data of the source patient(s) was moved into the target patient. ○ -1006: Operation failed because the data is in use. This can occur if any patient has an edit window open, or if any file within the File Database (FDB) folders is opened by another application. The Error String (obtain with VISI_GETERRORSTR) will contain additional information which can be used to trace the data which is in use. 	
<p>SearchPhotos</p>	<p>Multiple-purpose image searching function. Searches for, and optionally opens, specific types of images.</p> <p>Can cause VisiQuick to switch to a floating (on top) mini-mode, displaying a single tooth image, or two bitewings, etc.</p> <p>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.</p> <p>Can return a list of image IDs.</p> <p>Command string: %VQCMD%=searchphotos 100~PatID 10~Flags 11~Index 201~SearchText</p> <ul style="list-style-type: none"> ○ Parameter 100: the ID of the patient. ○ Parameter 10: the flags value. 	<p>3.0.1.1</p>



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 10 van 13

	<ul style="list-style-type: none"> ○ Parameter 11: the index value. ○ Parameter 201: the search parameter. <p>The flags parameter:</p> <ul style="list-style-type: none"> ○ Flags & spf_onlycheck == 0 = Not only search, but also actually opens the found images. ○ Flags & spf_onlycheck == 1 = Only search. <p>The index parameter: View to switch to in VisiQuick.</p> <ul style="list-style-type: none"> ○ 1 = spi_BiteWings (2x2 or 4x2 BiteWing view) ○ 2 = spi_DefaultView ○ 3 = spi_CompareView ○ 4 = spi_XrayView ○ 5 = spi_ColorView ○ 6 = spi_PanView ○ 7 = spi_CephView ○ 8 = spi_DocView ○ 9 = spi_FileView ○ 11 = spi_CtView 	
<p>SearchPhotoIDs</p>	<p>Searches for FDI tooth codes, or for custom tooth codes, and returns list of matching ImageIDs.</p> <p>Command string: %VQCMD%=searchphotoids 100~PatID 10~Flags 11~Index 201~SearchText</p> <p>The index parameter:</p> <ul style="list-style-type: none"> ○ 0 = FDI code in SearchText. ○ 1 = all images. <p>The flags parameter:</p> <ul style="list-style-type: none"> ○ Must be spis_toothpos if Index = 0. For Index 1, it is ignored. 	<p>3.0.1.1</p>
<p>SearchTStatus</p>	<p>Returns a string that indicates which images each tooth type has.</p> <p>Command string: %VQCMD%=searchtstatus 100~PatID 11~Index</p> <ul style="list-style-type: none"> ○ Index must be 1. 	<p>3.0.1.1</p>



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 11 van 13

	<p>The string is retrieved with VISI_GETERRORSTR.</p> <p>Each character indicates how many images there are for a tooth or modality.</p> <ul style="list-style-type: none"> ○ 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) 	
<p>GetImage</p>	<p>This function will cause VisiQuick to send an image bitmap to your app.</p> <p>Can be used to display a passport photo of the patient.</p> <p>Can be used to display the newest PAN or CEPH X-ray.</p> <p>Can be used to display the most recent bitewings.</p> <p>Can be used to display popup images with each tooth.</p> <p>Command string: %VQCMD%=getimage 100~PatID 10~Flags 11~Index 12~HWNDorHDC 201~SearchText</p> <p>The Flags parameter is a 32-bit integer with a bit mask of the following values:</p> <ul style="list-style-type: none"> ○ Flags & gif_photoid == 0: Search using exact ImageID ○ Flags & gif_photoid != 0: Search using FDI code <p>The Index parameter is a 32-bit integer with one of the following values:</p> <ul style="list-style-type: none"> ○ 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 	<p>3.0.1.1</p>



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 12 van 13

	<ul style="list-style-type: none"> ○ Index == gii_copydata: Send WM_COPYDATA with HGLOBAL with DIB. ○ HDC/HWND: DC or WND 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	<p>This function causes VisiQuick to activate, and then allows the user to select one or more images by dragging them 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.</p> <p>The command can be used to get images to add as attachments to an email in your app.</p> <p>The command can be used to add images to a print layout in your app.</p> <p>Command string: %VQCMD%=selectimage 100~PatID 12~HWND 13~Prompt</p> <ul style="list-style-type: none"> ○ HWND = Window handle to return to, when done selecting. ○ Prompt = Text to display in selection function. 	3.0.1.1
SetReturnWnd	<p>Sets the Window Handle that VisiQuick activates, when the green Back-button is clicked to return to your app. Normally, this command is sent automatically by the Link-DLL.</p> <p>Command string: %VQCMD%=setreturnwnd 11~HWND_Dec_or\$Hex</p> <ul style="list-style-type: none"> ○ HWND = Window Handle to return to. Decimal = '12345', hex = '\$12345'. 	3.0.1.1
UpdatePat	<p>Deprecated (Not used).</p>	3.0.1.1
DialPat	<p>Deprecated. Cause VisiQuick to use Windows Dialer to call the telephone number of the patient.</p> <p>Command string: %VQCMD%=dialpat 100~PatID</p>	3.0.1.1
DialNumber	<p>Deprecated. Causes VisiQuick to use Windows Dialer to call the specified telephone number.</p> <p>Command string: %VQCMD%=dialnumber 109~020-6722610</p>	3.0.1.1



VisiQuick-Link Developers Documentation

Link DLL SDK

Date: 16-08-2018

Version: VisiQuick 3.0.1.810

Page 13 van 13

SetAdminLinkDllPath	Deprecated (Not used). Command string: %VQCMD%=setadminlinkdllpath 12~dllpath	3.0.1.1
Quit	Causes VisiQuick to terminate. It is not necessary to send this command directly, as it is done automatically by VISI_EXIT, when your app is quitting. Command string: %VQCMD%=quit 10~Flags <ul style="list-style-type: none">○ Flags = not used.	3.0.1.1
WordLink	Deprecated (Not used).	3.0.1.1