

## User's Guide

### Contents

1. Introduction.....	1
1.1 Scope.....	1
1.2 Normative references .....	2
2. SDK composition .....	2
2.1 Static Library Package – “QRC_PRO_Linux_64_o” .....	2
2.2 Dynamic Library Package – “QRC_PRO_Linux_64_so” .....	2
3. Deployment.....	3
3.1 Static Library Package – “QRC_PRO_Linux_64_o” .....	3
3.2 Dynamic Library Package – “QRC_PRO_Linux_64_so” .....	3
4. Licensing / Evaluation .....	4
4.1 Licensing from TwoDtgLicense app.....	4
4.1.1 Executing TwoDtgLicense.....	4
4.1.2 Online Activation.....	4
4.1.3 Manual Activation.....	5
4.2 Licensing from User's Application.....	5
4.2.1 Licensing system initialization.....	6
4.2.2 On-line library activation on 2DTG website.....	6
4.2.3 Manual activation (no access to the Internet from your PC).....	6

## 1. Introduction.

### 1.1 Scope

This document is applicable to the **Professional** edition of the QR Code Decoding SDK compiled in Linux 64 bit environment.

SDK delivery package is notated as **QRC\_PRO\_Linux\_64\_X**, where X:=”o” or “so”, differentiates between dynamic and static libraries, and notation “64” means 64-bit version of Linux. Both static and dynamic libraries are available:

# QR Code Decoding SDK

## (Professional Edition, Linux 64 bit)

---

- **icQRCode\_Pro.o** – static library
- **libcQRCode\_Pro.so** – dynamic library

Library interface and its features are the same as for the Windows version. They described in detail in the User's Guide "**QR Code Decoding SDK (Professional Edition)**".

The library is designed to decode QR Codes in accordance with ISO/IEC 18004 Symbology specification. Symbol quality assessment is provided in accordance with ISO/IEC 15415 standard.

## 1.2 Normative references

ISO/IEC 18004 - Symbology specification - QR Code

ISO/IEC 15415 - Symbol quality - Two-dimensional symbols

## 2. SDK composition

### 2.1 Static Library Package – “QRC\_PRO\_Linux\_64\_o”

Decoding SDK package contains:

1. Static Library **icQRCode\_Pro.o**, designed to perform symbol search, recognition and decoding.
2. Source codes of the demo program (that calls library) that illustrate its usage, as follows:
  - **QRPro\_Types.h** - header file that describes library interface
  - **Demo.cpp** - source code of sample application that uses QRC library
  - **LoadBMP.c, LoadBMP.h** - the functions for loading "BMP" files
3. Executable file:
  - **demo\_pro.out** - demo application built using Demo.cpp source code.
4. **Read Me** – file (if required)
5. This User's Guide

### 2.2 Dynamic Library Package – “QRC\_PRO\_Linux\_64\_so”

Decoding SDK package contains:

# QR Code Decoding SDK

## (Professional Edition, Linux 64 bit)

---

1. Dynamic Library **libcQRCode\_Pro.so**, designed to perform symbol search, recognition and decoding.
2. Source codes of the demo program (that calls library) that illustrate its usage, as follows:
  - **QRPro\_Types.h** - header file that describes library interface
  - **Demo.cpp** - source code of sample application that uses QRC library
  - **LoadBMP.c, LoadBMP.h** - the functions for loading "BMP" files
3. Executable file:
  - **demo\_pro\_so.out** - demo application built using Demo.cpp source code.
4. **Read Me** – file (if required)
5. This User's Guide

### 3. Deployment

#### 3.1 Static Library Package – “QRC\_PRO\_Linux\_64\_o”

- Unpack all files into the same directory.
- Set rights for the "executable" to the file "**demo\_pro.out**".
- Run "**demo\_pro.out**".
- Source codes of **.c** and **.h** files show how the library functions should be called. Build Linux demo application using the command:

```
g++ Demo.cpp icQRCode_Pro.o -m64 -lm -o mydemo.out
```

- After completing compilation that would generate instant executable program "**mydemo.out**" you can run it the same way as it's described above for "demo\_pro.out".

#### 3.2 Dynamic Library Package – “QRC\_PRO\_Linux\_64\_so”

- Unpack all files into the same directory.
- Copy **libcQRCode\_Pro.so** to **.../usr/lib**.
- Set rights for the "executable" to the file "**demo\_pro\_so.out**".
- Run "**demo\_pro\_so.out**".
- Source codes of **.c** and **.h** files show how the library functions should be called. Build Linux demo application using the command:

# QR Code Decoding SDK

## (Professional Edition, Linux 64 bit)

---

**g++ Demo.cpp -rdynamic -ldl -m64 -lm -o mydemo.out**

- After completing compilation that would generate instant executable program "**mydemo.out**" you can run it the same way as it's described above for "demo\_pro\_so.out".

## 4. Licensing / Evaluation

Stand-alone license is locked to the computer, on which it was activated. License can be activated at any time during 30-days trial/evaluation period.

Trial license is fully functional but limited to 30 days after the first use of library.

Licensing can be done either from the customer's app or by executing **TwoDtgLicense** app, provided by 2DTG as part of the SDK package.

License may be transferred to another computer after it's deactivated from the first one. Deactivation can be done in "on-line mode" only, so internet connection is required.

### 4.1 Licensing from TwoDtgLicense app

#### 4.1.1 Executing TwoDtgLicense

**libqt5widgets.so.5** is required for the licensing application to function. It has been reported that some versions of Linux do not have this application installed. If you receive an error reporting no such file found you just need to install the missing item using the following command:

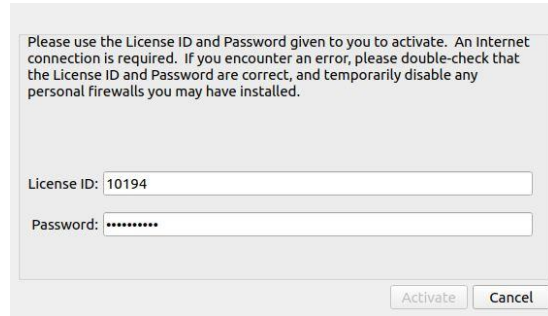
```
sudo apt-get install libqt5widgets5
```

#### 4.1.2 Online Activation

Once starting the GUI you will have a couple of activation options, either Online or Manually.

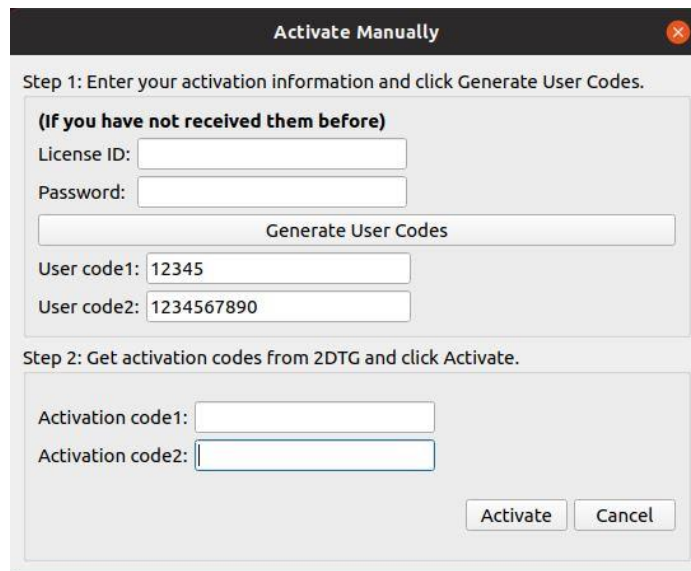
Select the "Activate Online" button and you will then be prompted to enter your License ID and Password:

# QR Code Decoding SDK (Professional Edition, Linux 64 bit)



## 4.1.3 Manual Activation

If you are working on a device that does not have internet access you also have the option for activating your license manually. Simply select the “Activate Manually” button and you will then be prompted with the manual activation window below.



You will first need to enter your License ID and Password and click the Generate User Codes button. Once selected you will see User Code 1 and User Code 2 which you will need to provide to any 2DTG representative via telephone\* or email\*\* and they will provide you with the required Activation Code 1 and Activation Code 2. Once entered just click the Active button and that's it!

## 4.2 Licensing from User's Application

The description of how to activate (begin trial period) of the library from your application can be found in **twodtg\_license.h**.

# QR Code Decoding SDK

## (Professional Edition, Linux 64 bit)

---

### 4.2.1 Licensing system initialization.

Licensing system must be initialized before the first use of the Decoding library.

Call '**EvaluationDayCount**' function from your application to start 30-days trial period. This call requires root access. All subsequent calls of the decoding library do not require administrative privileges.

To check out if administrative privileges are required you can call '**RequiredRootAccess**' function.

### 4.2.2 On-line library activation on 2DTG website.

Call '**ActivationOnline**' function from your application and pass 'License ID' and 'License password' received during the purchase.

### 4.2.3 Manual activation (no access to the Internet from your PC).

- A. Call '**GetUserCodes**' function – it will generate userCode1 и userCode2 - OUTPUT parameters from your PC, required for Trigger Code dialog on 2DTG website. (If this function returns '0' for one or both parameters, please, contact 2DTG technical support).
- B. Log-in to your account on 2DTG website from a PC having access to the Internet, open your Order page and then 'Manual Unlock License' page:

▼ Enter License id and License password:

**License id: \***  
  
License id from email

**License password: \***  
  
License password from email

▼ Enter the codes from the user's Trigger Code dialog:

**Code 1: \***  
  
Code 1 from the user's Trigger Code dialog

**Code 2: \***  
  
Code 2 from the user's Trigger Code dialog

Unlock

Enter required data and click 'Unlock' – the system will return to you 'ActivationCode1' and 'ActivationCode2'.

## QR Code Decoding SDK

(Professional Edition, Linux 64 bit)

---

- C. Call '**ActivationManual**' function from your application and pass received 'ActivationCode1' and 'ActivationCode2'.