Chapter 1 Development Tools

1.1 Integrated Development Environment (IDE) Instructions

SpaceChain OS provides users with an environment and various tools to build and create applications for space exploration. This manual highlights the development tools and functions related to blockchain technology.

Special development tools are actually not required for the use of blockchain-based DAPP. SpaceChain OS upper script application can be developed directly using scripting tools such as Python and Lua. However, if users want to develop SpaceChain OS applications, or modify its system files and BSP standard, then they will need to use special development tools.

Currently, the exclusive development tool is being developed with our partner Acoinfo. Developed based on RealEvo-IDE, the tool integrates functions that lets users design, develop, simulate, deploy and perform testing.

SpaceChain and Acoinfo are working together to optimise and modify the tool in terms of its applications and drivers in the aerospace field.

The tool is suitable for chip hardware platforms in the aerospace field. In the future, this exclusive tool will play an important role in the application of aerospace and blockchain technology. Other parts of the OS will also be developed using RealEvo-IDE.

SpaceChain OS will be developed and maintained independently, serving the aerospace and blockchain industries. As the operator of the infrastructure, SpaceChain OS operates on aerospace devices and the corresponding ground receiving and controlling equipment.

As this is an open-source OS, we hope the community will get actively involved in maintaining and submitting codes.

In this manual, we will use the tools from the previous version to demonstrate development. Please note that this article will get updated at any time. Do frequently visit the SpaceChain website for the latest version.

1.2 RealEvo-IDE installation

Access to IDE: Log into the official SpaceChain's website to get usage license. However, do note that the application portal for usage license will only be launched at a later date. In the meantime, you can contact info@spacechain.com for permission to obtain the license.

After getting a set of SylixOS development kit, users can read the file in the CD to get more information about the SylixOS integrated development kit.

Here's how to install RealEvo-IDE using the SylixOS development kit CD:

- Open SylixOS IDE.
- Double click to open InstallWizard.exe and get the installation tool set, notes and SylixOS website information of SylixOS development kit.

SylixOS IDE includes the installation of "RealEvo-IDE", "QtCreator", and "RealEvo-QtSylixOS". It is recommended that you close the antivirus program during installation.

- Perform the installation according to the sequence in Figure 1.1. RealEvo-IDE in the integrated development environment can be used normally after registration. Refer to *RealEvo Software Register Procedures* in the CD for registration processes and relevant notes.

IDE is installed once the above procedures have been completed.

Figure 1.1 Tool Set of IDE

1.3 RealEvo-IDE register

1.3.1 License description of RealEvo series software

RealEvo series software consists of the following independent software:

- RealEvo-IDE: To be registered independently.
- RealEvo-Simulator: No registration required.
- RealEvo-QtSylixOS: No registration required.

1.3.2 Online registration procedures

- Open RealEvo-Register tool. Select Start Menu → All Programs → ACOINFO → RealEvo → RealEvo-Register. The RealEvo-Register running interface is shown in Figure 错误！未找到引用源。. Key in the serial number (in the disc or sent through mail) in the box.

- Click “Generate Register Code” button (Figure 1.2) and select the disk and network adapter to be bound in the pop-up dialog box.

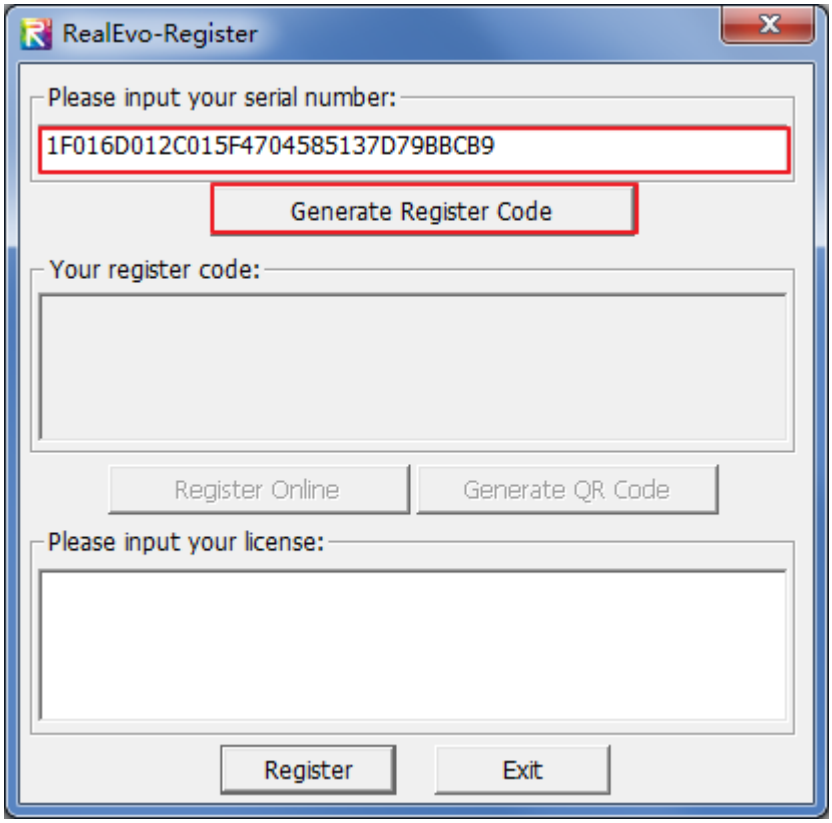

Figure 1.2 Input SN

- After the disk and the network adapter are selected, click "OK" to generate a registration code.

(Figure 1.3).

Note: No change is allowed once the disk and the network adapter are bound. The serial number can only be registered with the bound disk and the network adapter. Once the disk or network adapter is replaced, the software will be unavailable. Therefore, select the physical disk and physical network adapter that are available on the computer for a long term. The network adapter can't be disabled during software registration and use.

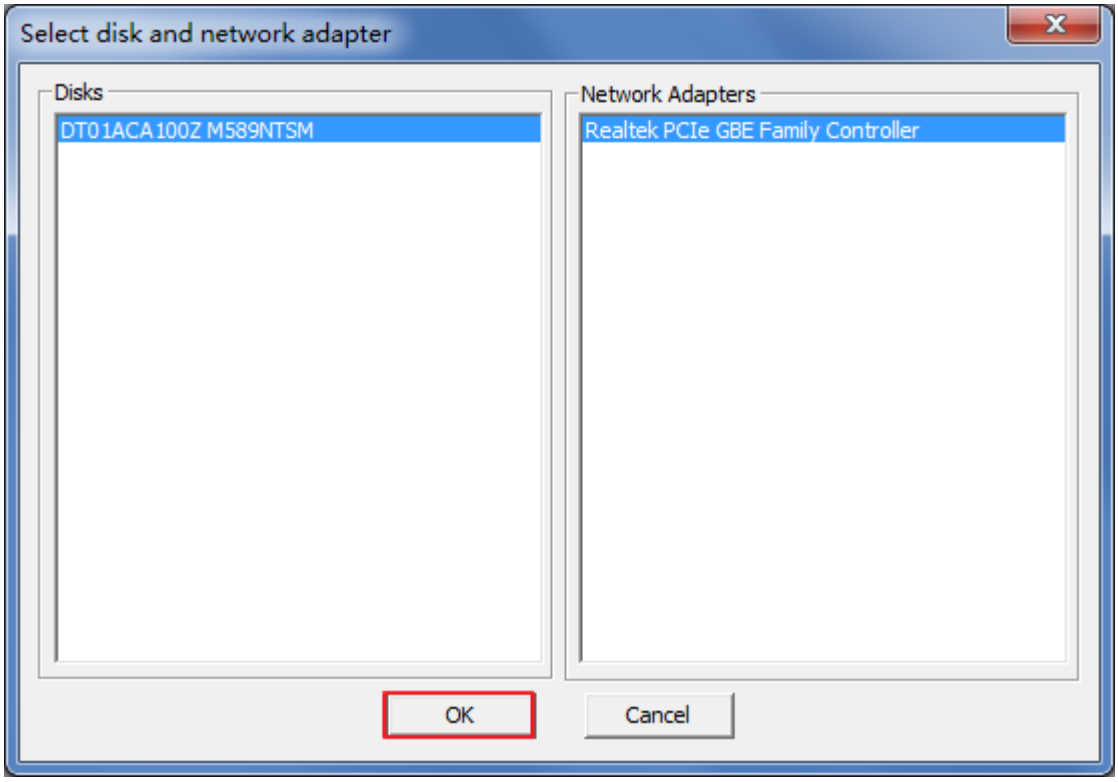

Figure 1.3 Select Disk and Network Adapter

- If the computer is already connected to the Internet, click "Register Online" to get license (Figures 1.4).

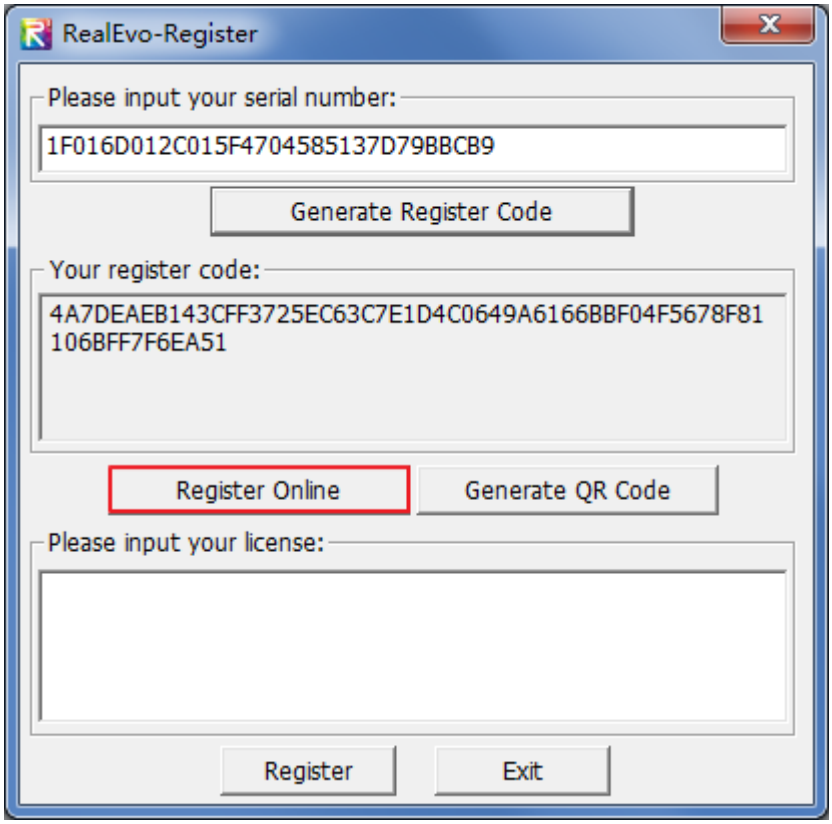

Figure 1.4 Generate Register Code

- Copy the license generated in Figure Figure 1.5, and paste it to the input box of license below RealEvo-Register tool, as shown in Figure 1.6.

Note: when the license of the software expires, a new license can be registered by starting RealEvo-Register.exe tool.

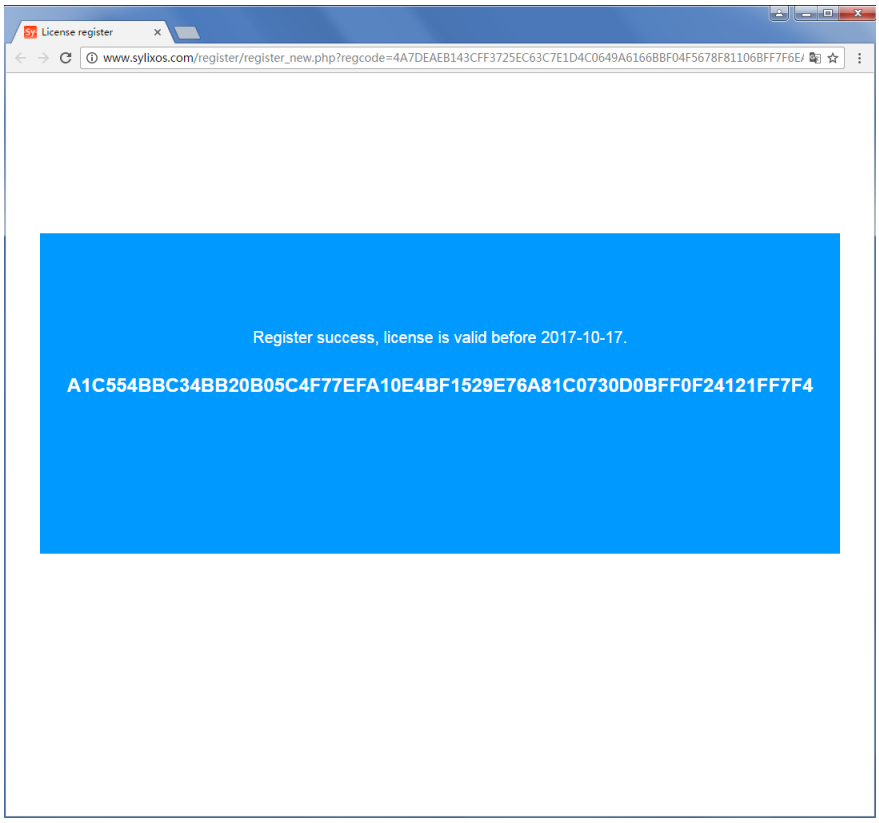

Figure 1.5 Generate License

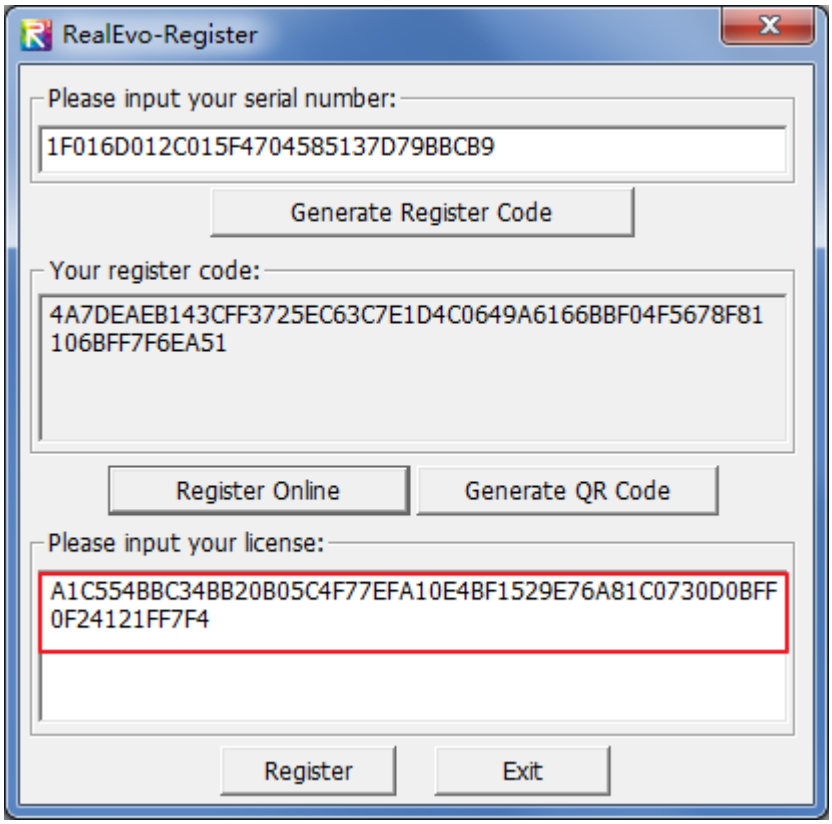

Figure 1.6 Input License

- Click “Register” (Figure 1.6) to complete the registration (Figure 1.7).

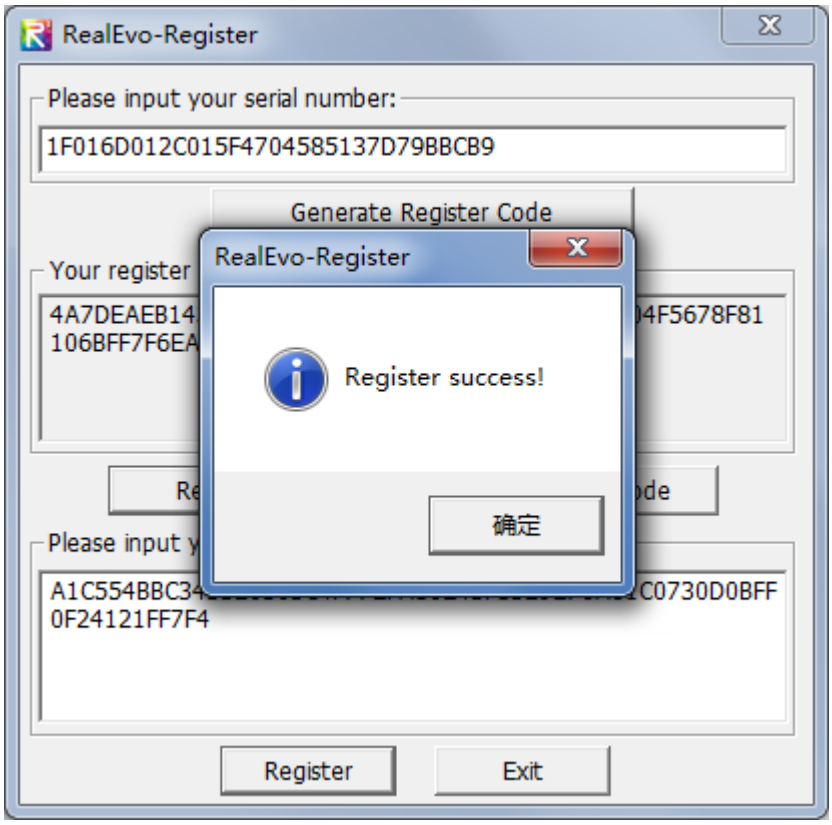

Figure 1.7 Registration completed

Chapter 2 Import and Deployment of Blockchain Project

2.1 Hardware preparation

This manual uses MYIR Z-turn development board as an example. The core CPU of the development board is ARM A9 dual-core. After SpaceChain OS on new Z-turn development board is started for the first time, partition and formatting operations need to be executed (Figure 2.1).

```
[unknown@sylixos:/]# umount /media/sdcard0
[unknown@sylixos:/]# fdisk -f /dev/blk/sdcard-0
block device /dev/blk/sdcard-0 total size: 3812 (MB)
please input how many partition(s) you want to make (1 ~ 4) : 2
please input how many bytes align (4K 8K ...) : 4096
please input the partition 0 size percentage(%) 0 means all left space : 20
is this partition active(y/n) : n
please input the file system type
1: FAT   2: TPSFS   3: LINUX   4: RESERVED
1
please input the partition 1 size percentage(%) 0 means all left space : 0
is this partition active(y/n) : y
please input the file system type
1: FAT   2: TPSFS   3: LINUX   4: RESERVED
2
making partition...
block device  : /dev/blk/sdcard-0
block type    : SD/MMC
block serial  : 00002F3F
block firmware: 2017.10, v0.0
block product : 00000 SDHC memory card
block media   : SDHC(v2.0)

partition >>

PART ACT  SIZE(KB)  OFFSET(KB)             TYPE
---- --- ---------- ---------- ------------------------------------
   0         780492       1024 Win95 FAT32 Partition
   1  *     3121972     781516 SylixOS True Power Safe Partition

total partition 2
[unknown@sylixos:/]# remount /dev/blk/sdcard-0
Block device /dev/blk/sdcard-0 part 0 mount to /media/sdcard0 use vfat file system.
Magic number error, mount failed
Block device /dev/blk/sdcard-0 part 1 mount to /media/sdcard1 use tpsfs file system.
[u/]# mkfs /media/sdcard0
now format media, please wait...
disk format ok.
[unknown@sylixos:/]# mkfs /media/sdcard1
now format media, please wait...
disk format ok.
```

Figure 0.1 Partition and Formatting Operations

Note: After completion of formatting, upload the boot file BOOT.bin to /boot directory and execute sync command. The file system can be mounted normally after reboot.

2.2 Create Base Project

- Select “File → New → Project”, and a window will pop up listing the wizards supported by SpaceChain OS (Figure 2.2).

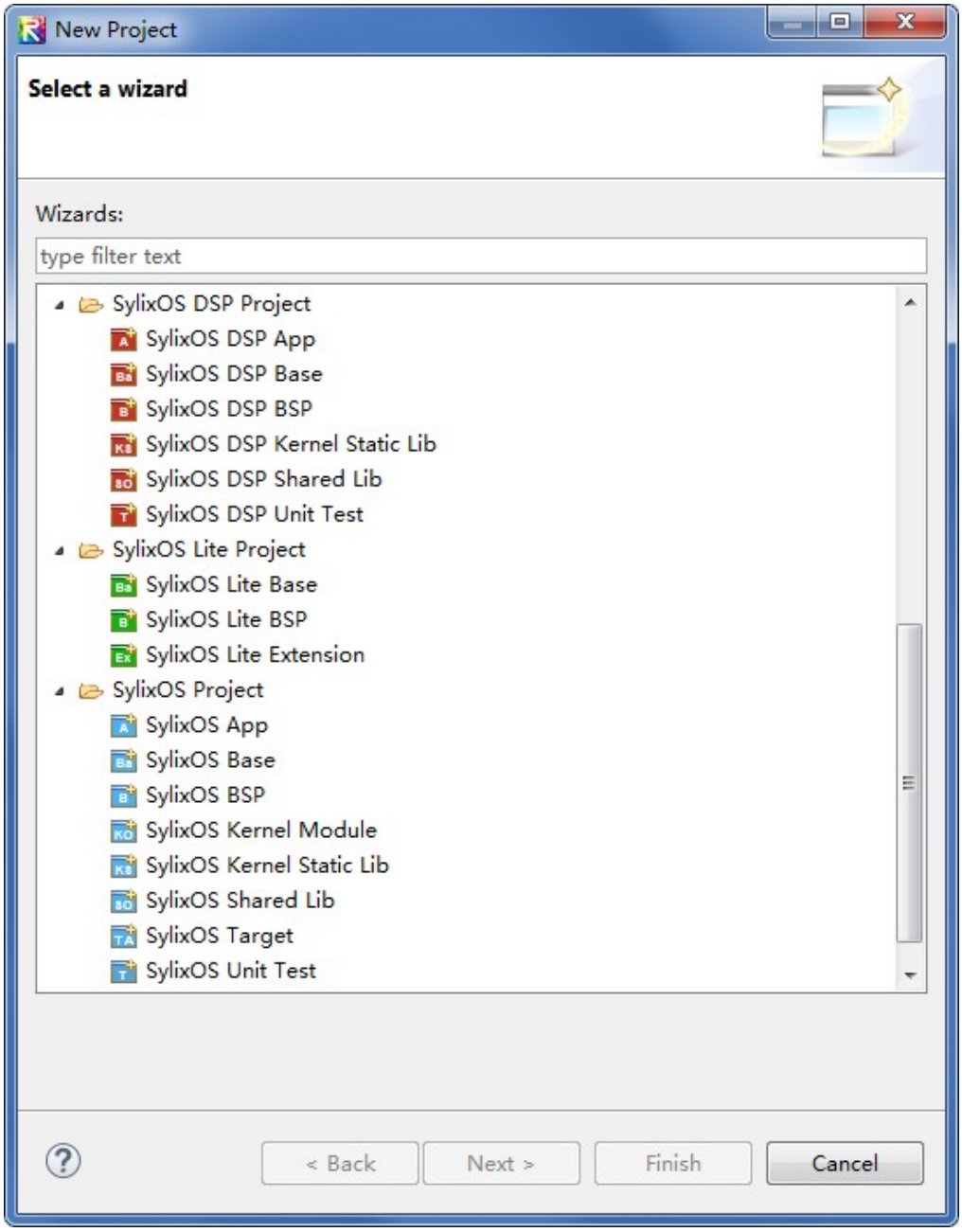

Figure 0.2 Create a Project

- Select “SylixOS Base” (Figure) and click “Next”. It will lead you to the configuration page where you’ll need to input the project name in the box (Figure 2.3).

Note: Do not include spaces in project name, uncheck “Use default location” and create Project in the location except for default Workspace

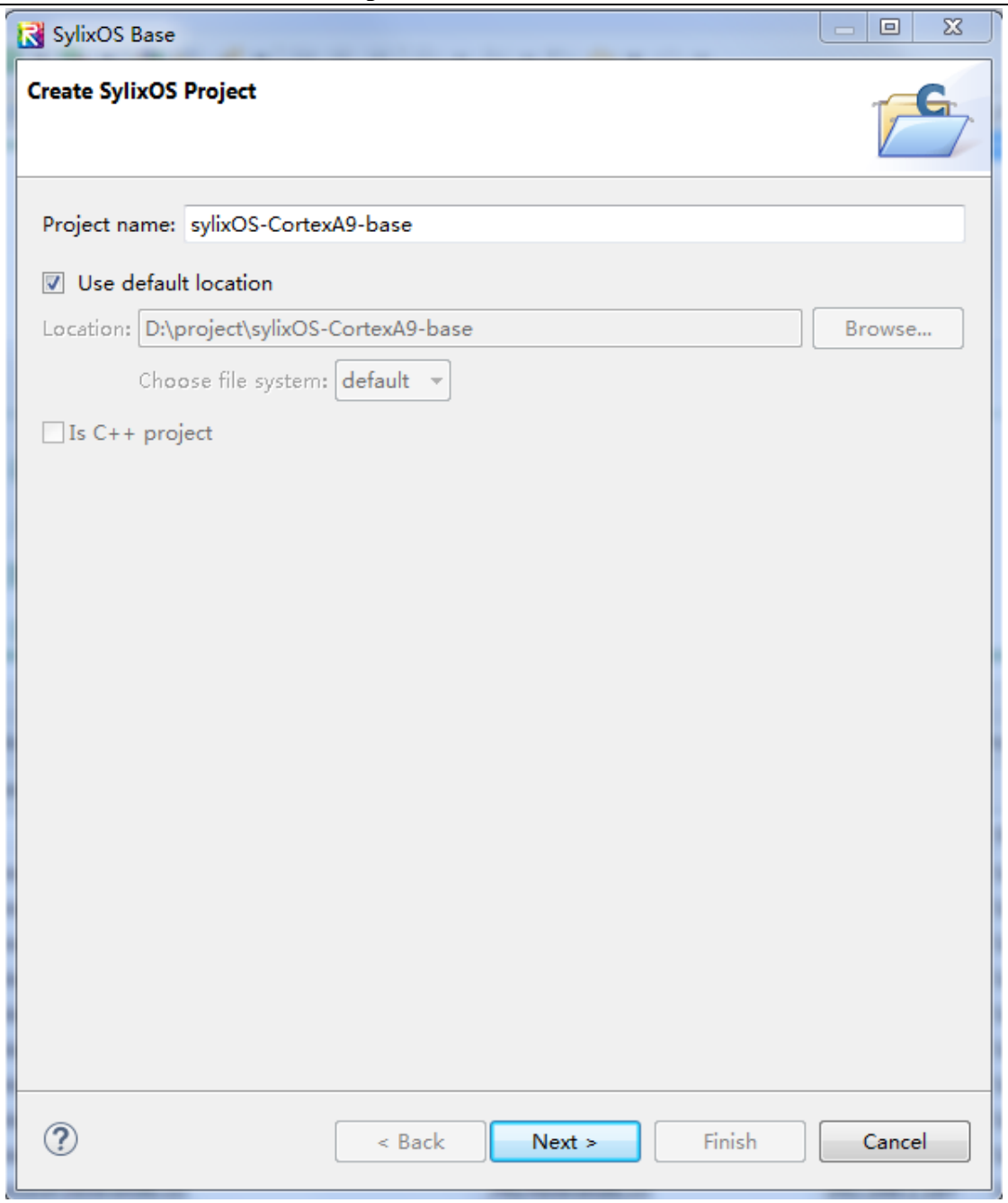

Figure 0.3 Create SylixOS Base Project

- Click “Next”. It will lead you to the Base Setting page of tool chain where you’ll need to set the base compiler options, including the Toolchain (select my64-sylixos-toolchain), Debug Level, CPU Type and FPU (floating-point) Type settings (Figure 错误！未找到引用源。).

Configuration item analysis:

- *Toolchain: toolchain (select mips64-sylixos-toolchain to compile 64-bit system, and select mips-sylixos-toolchain to compile 32-bit system)*
- *Debug Level: debug level, SylixOS provides both Debug and Release configurations*
- *CPU Type: processor model*

❖ *FPU Type: floating point processor*

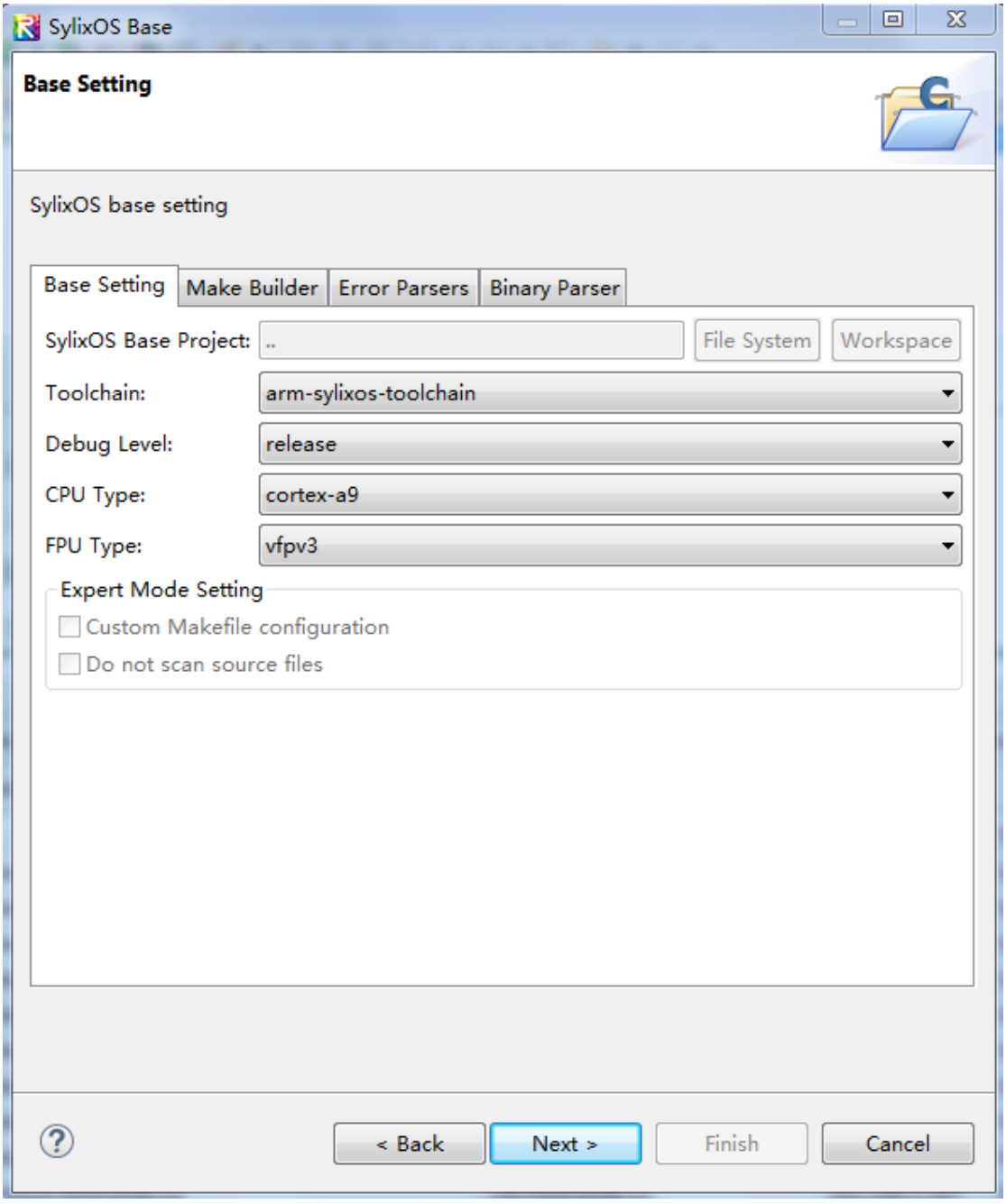

Figure 0.4 Base Setting of Base Project

● Click “Next” to enter the Component Select page. SylixOS contains a large number of optional components (Figure 2.5). Select required components accordingly. Click “Finish” to complete the project creation.

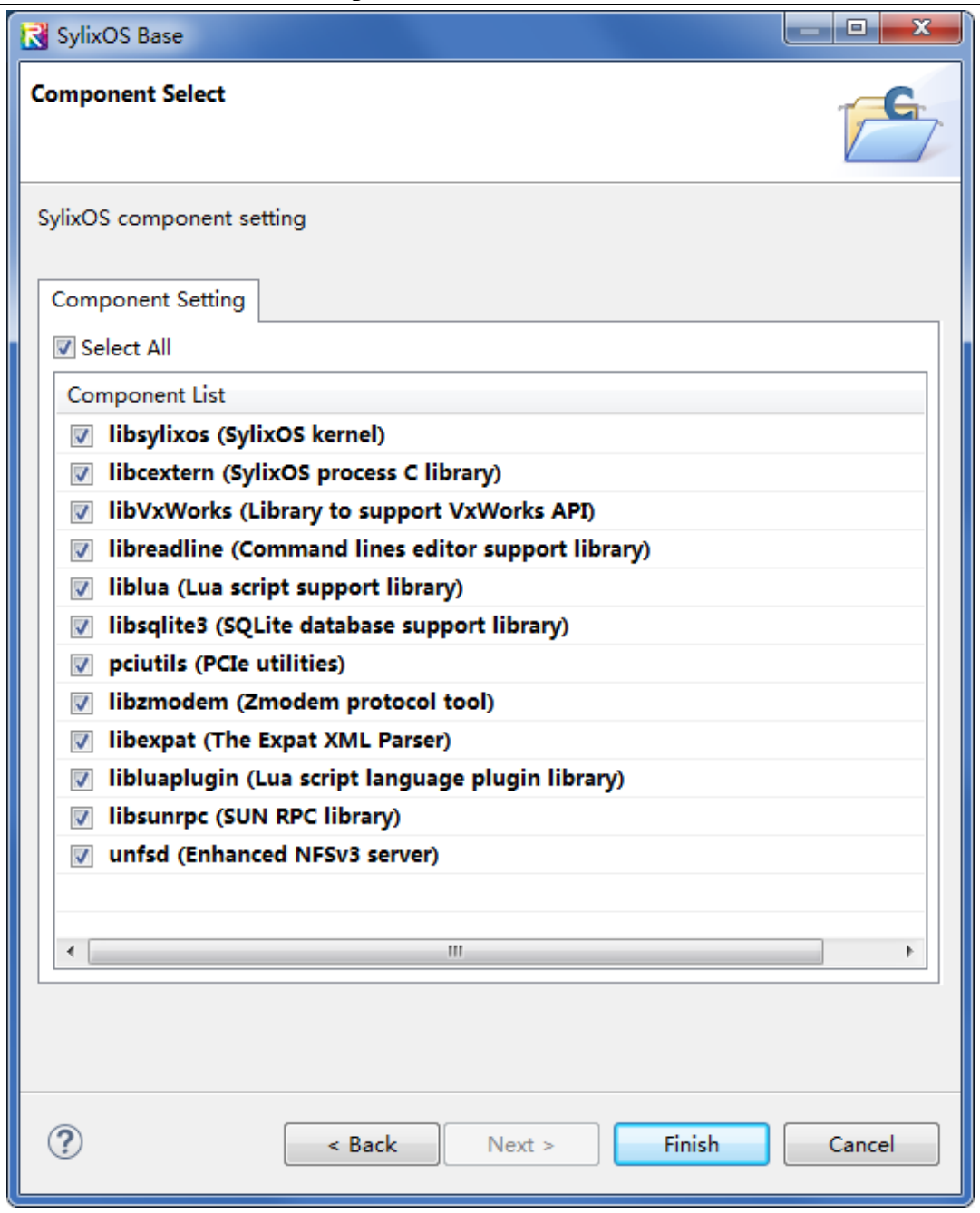

Figure 0.5 SylixOS Base Component Selection

- Click the created Base project and right click “Build Project” to build SylixOS Base project.

Note: For detailed creation information of SylixOS Base project, refer to *RealEvo-IDE Instruction Manual*.

2.3 Project import

After Blockchain BASE project is created, the application project qtum and library file project need to be imported through RealEvo-IDE, and the library files to be imported include:

1. Libboost
2. Libdb
3. Libenvent
4. libopenssl

- This file takes the libopenssl as an example to demonstrate the project import. Select “File → Import” and open the dialog box of Import (Figure Figure 0.6.6).

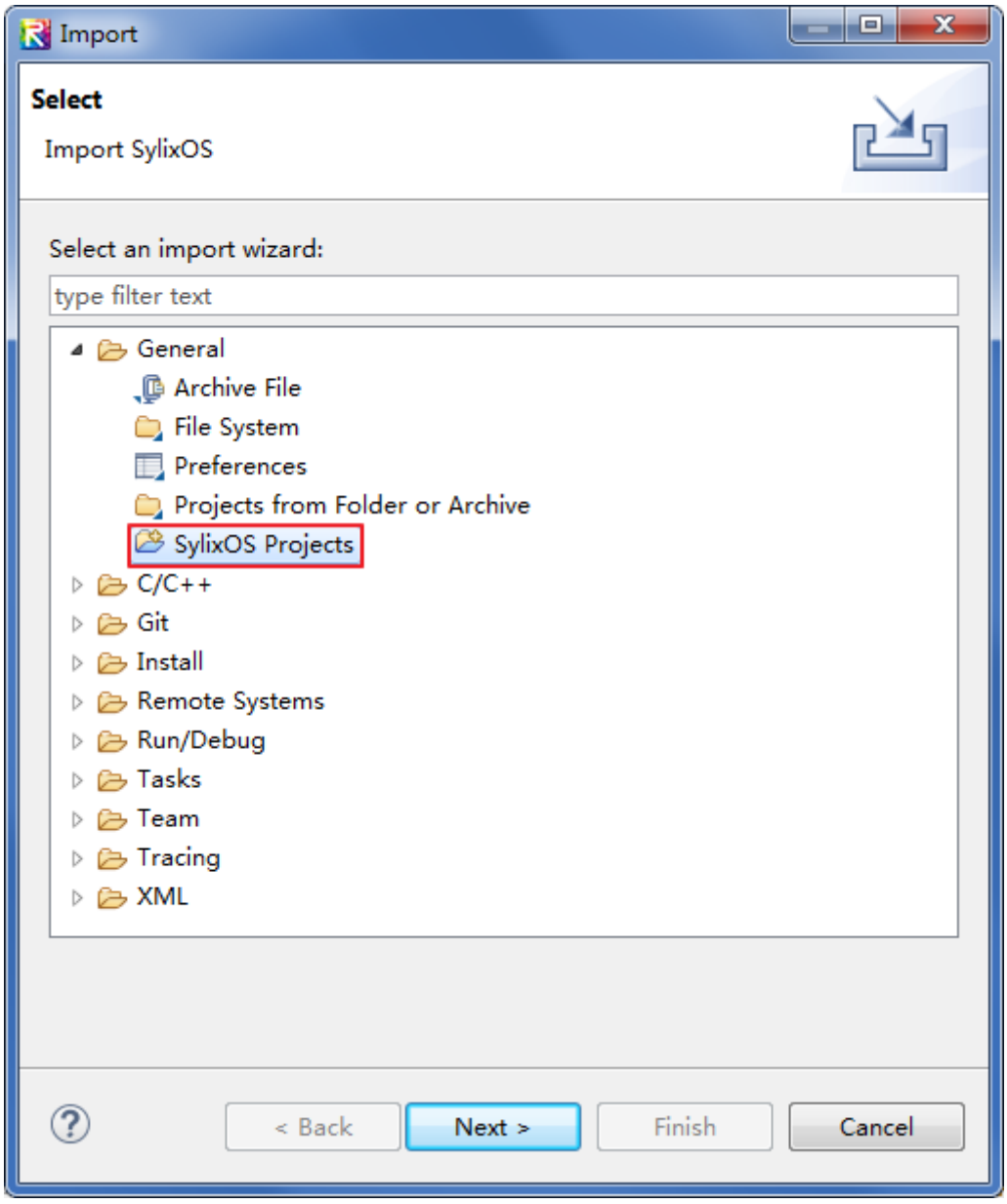

Figure 0.6 Import Project

- Check “SylixOS Projects” and click “Next” to enter the project selection interface (Figure 2.7).

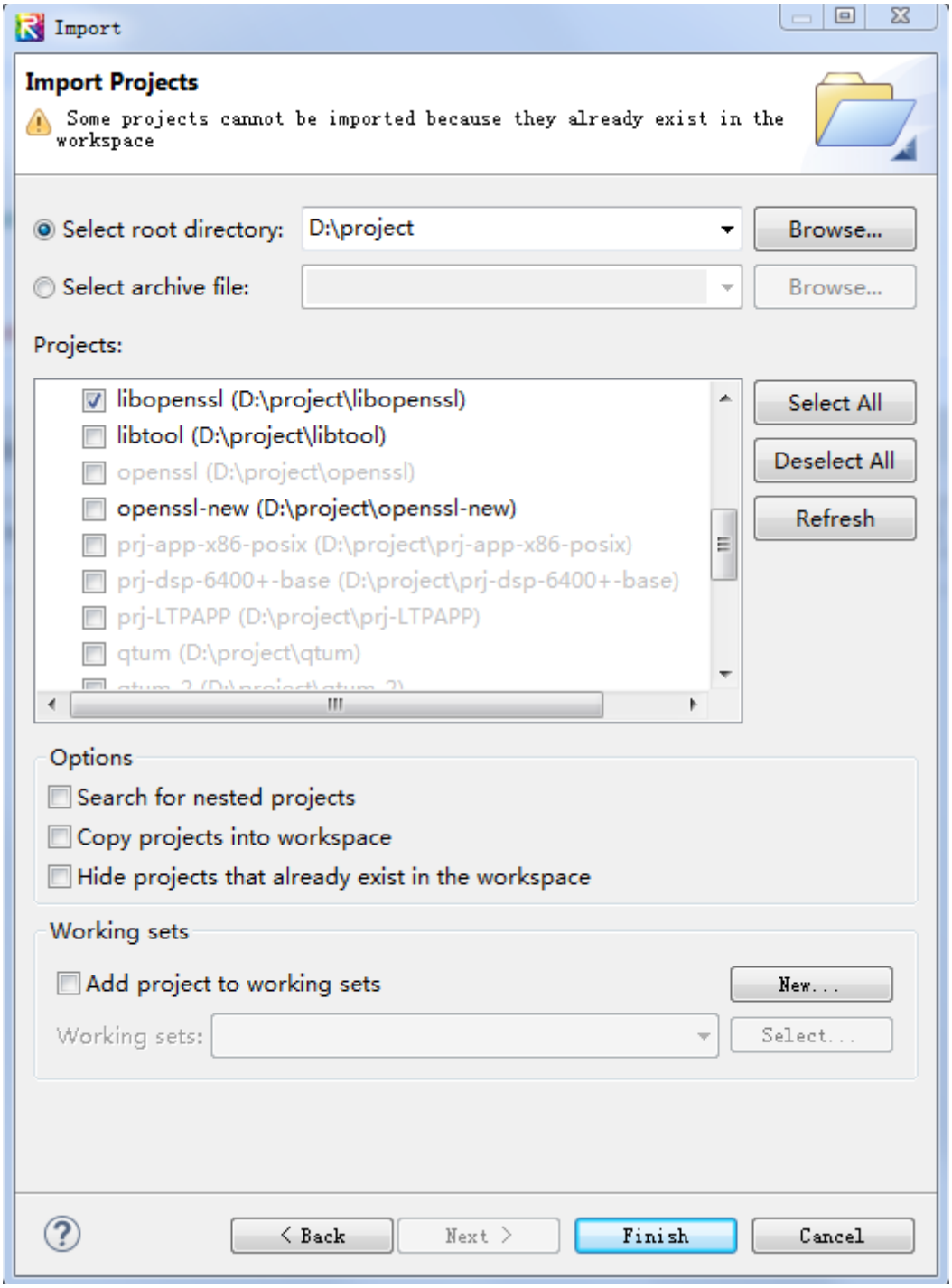

Figure 0.7 Select Import Projects

- Click “Browse”, select the directory of the project in the pop-up box, list the project names under the directory that can be imported in Projects list, select the project to be imported and click “Finish” to complete the project import.

- When the project import is successful, users can see the newly imported project libopenssl in Workspace project list (Figure 2.8).

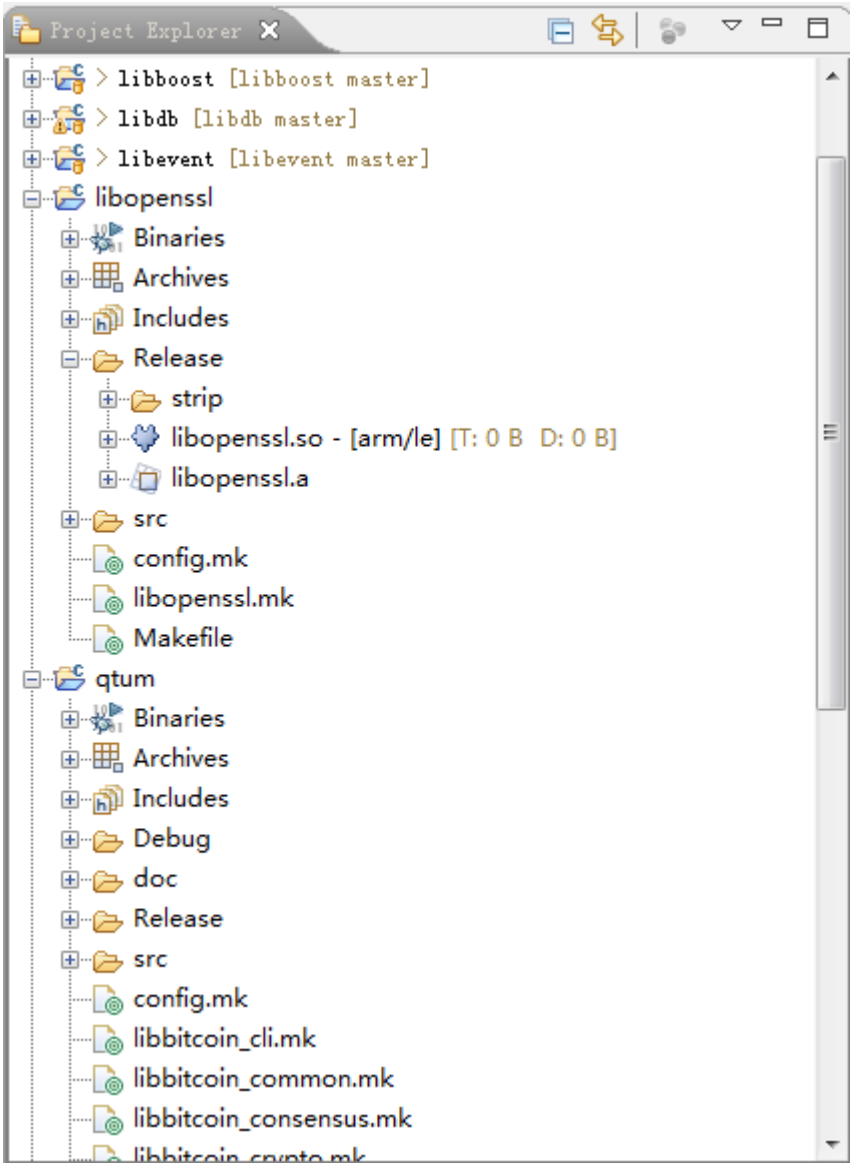

Figure 0.8 Completion of Project Import

2.4 Project deployment

The following library files are required for this project:

1. Libopenssl
2. Libboost
3. Libdb
4. Libevent

After the deployment of the library file is completed, the application project program qtum can be deployed;

Use the one-click deployment feature of RealEdo-IDE for program deployment (in libopensl), right-click the libopensl project, and select “Properties” Open Project Properties Page, select “SylixOS Project → Device Setting” Tabs to view and change deployment configurations. When creating SylixOS App project, RealEvo-IDE will add the projects in the current configuration export folder with the same names of the project name into the file list.

As shown in Figure 2.9, users can select any existing device deployment files that have been added in Workspace.

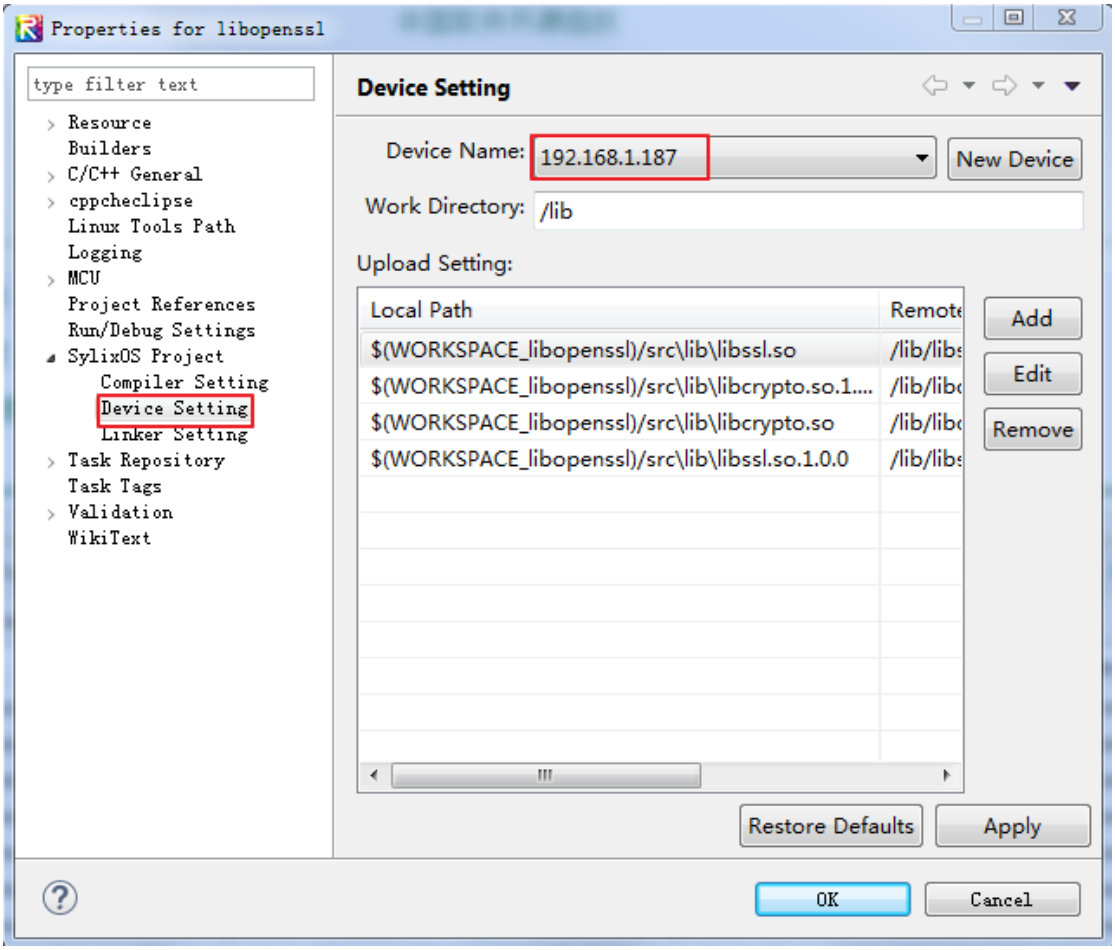

Figure 0.9 Deployment Setting of SylixOS libopenssl Project

After the setting is completed, right click the project, and select “SylixOS → Upload” to deploy the project.

Note: if the deployment does not succeed, check whether the firewall is in the off status.

Log in to the target board system, switch to “/apps/qtum” directory, and run qtum command, as shown in Figure 2.10; for details, refer to Chapter III.

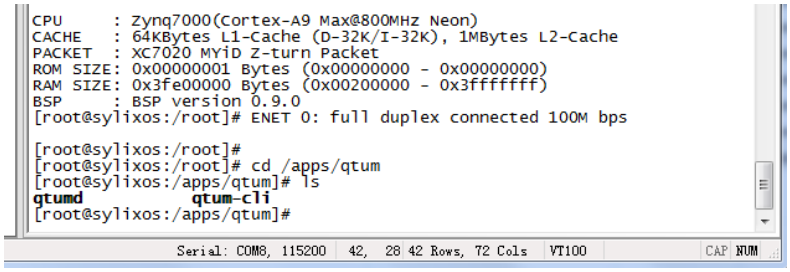

Figure 0.10 Enter qtum Environment and Execute qtum Command

2.5 Project deletion

Right click the project to be deleted, and select “Delete” to delete the project, as shown in Figure 2.11.

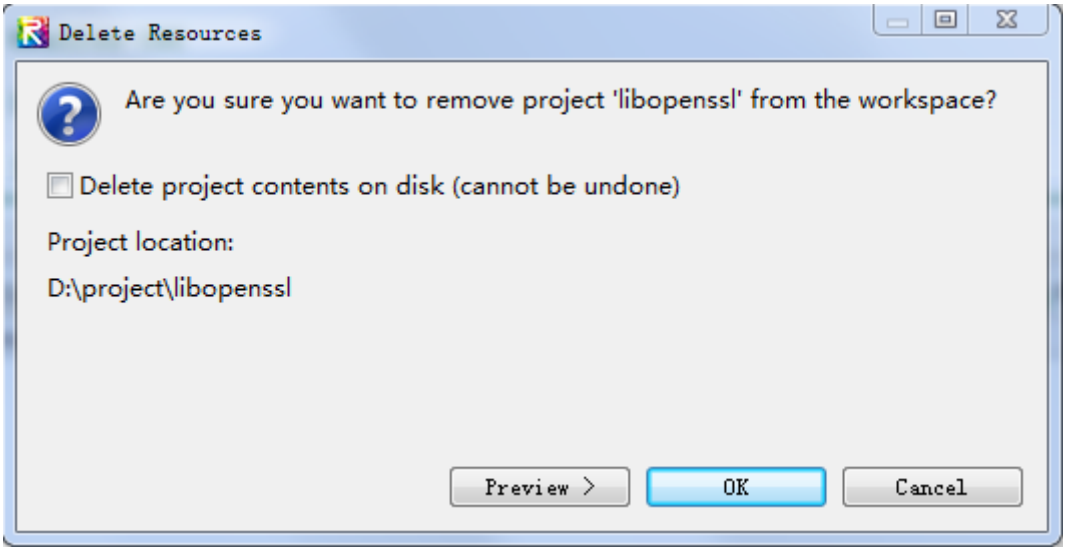

Figure 0.11 Delete Project

Chapter 3 Basic Usage of Blockchain Wallet

3.1 Usage of common commands

Qtum software has a lot of commands. This manual will take a part of common commands under -testnet pattern as examples to demonstrate its usage methods and results.

3.1.1 Get the node information

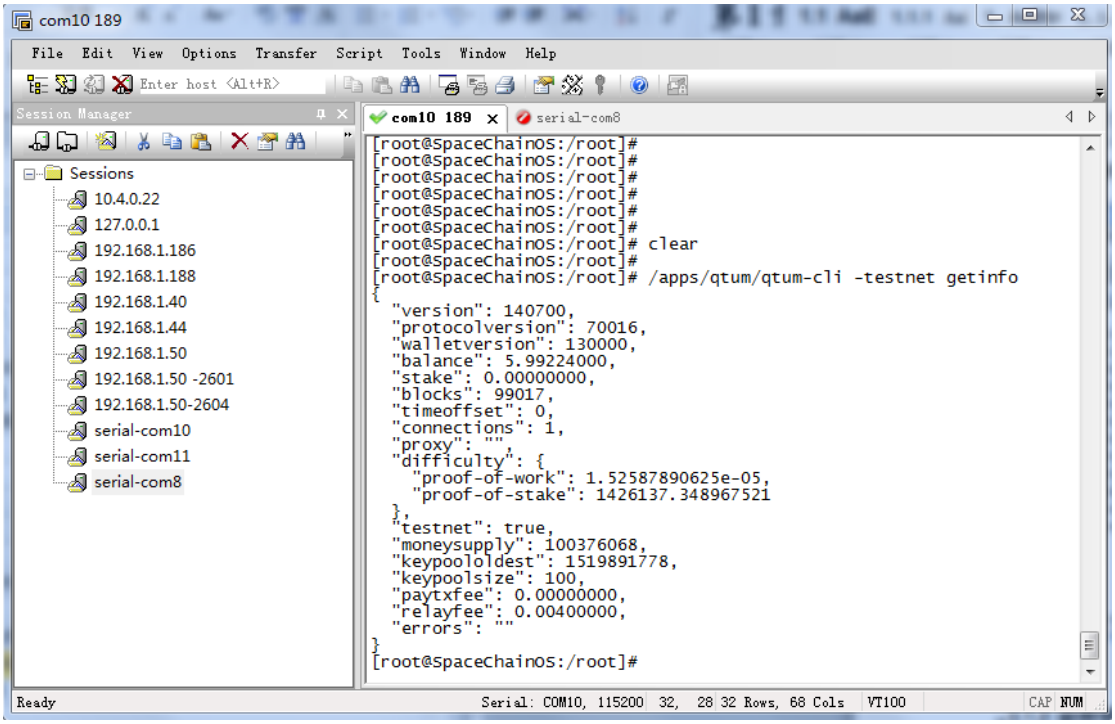

Figure 0.1 Common Demonstration for Getting the Node Information

3.1.2 Get the neighbor node information

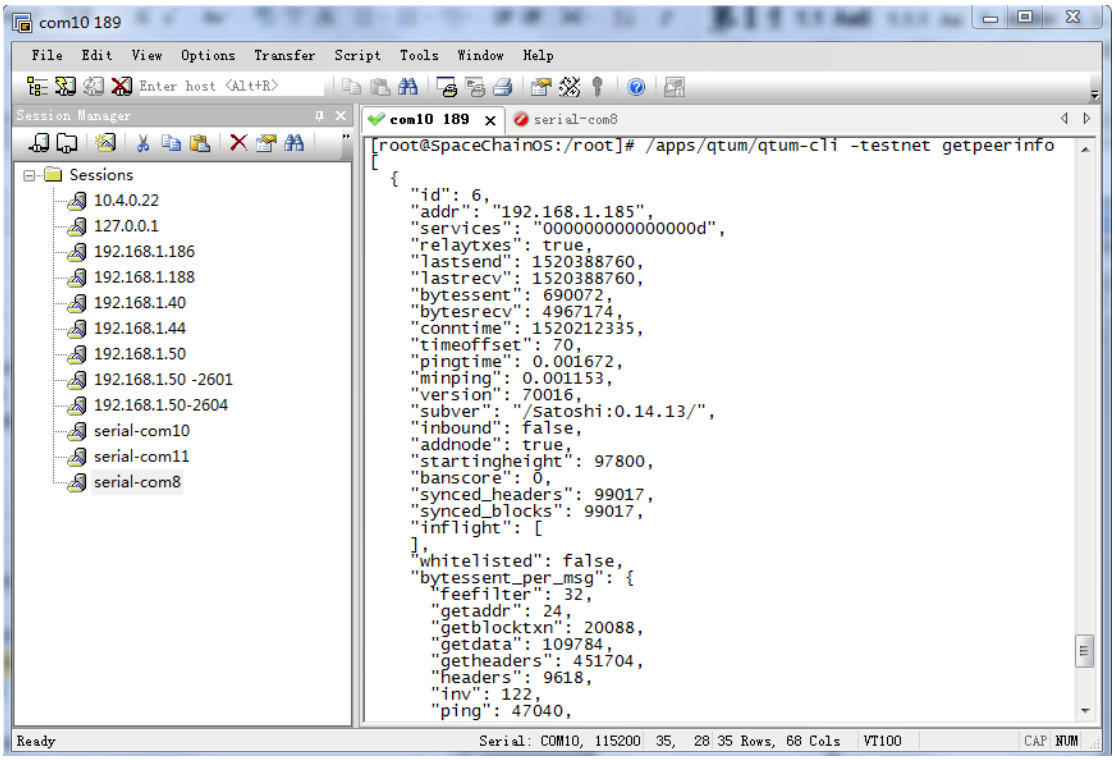

Figure 0.2 Common Demonstration for Getting the Neighbor Node Information

3.1.3 Transfer

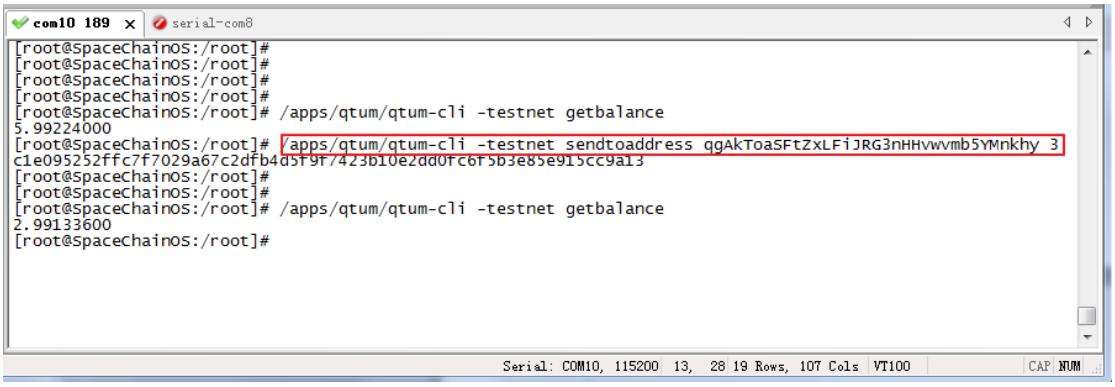

Figure 0.3 Transfer Command Demonstration

3.1.4 Check account balance after transfer:

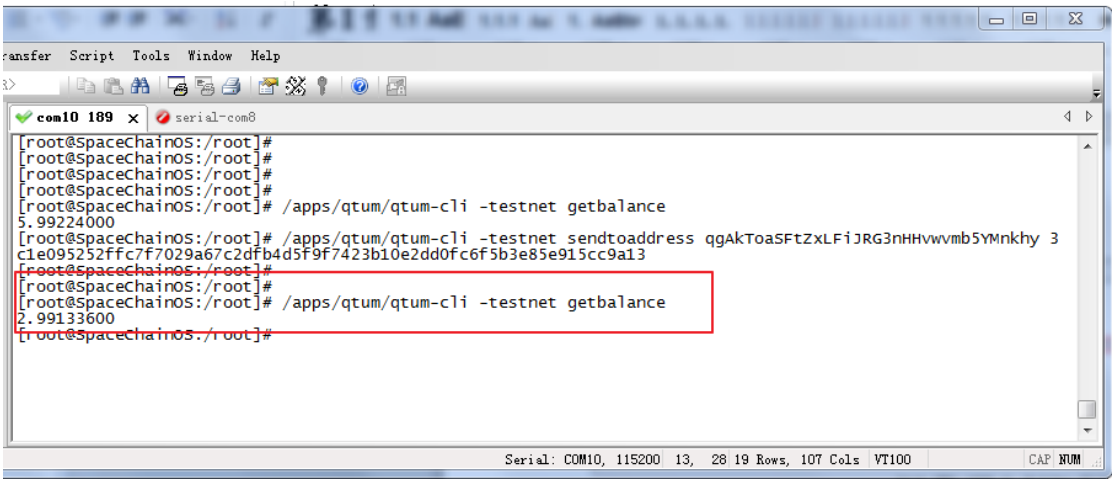

Figure 0.4 Balance Command Demonstration after Transfer

3.1.5 Query transaction record

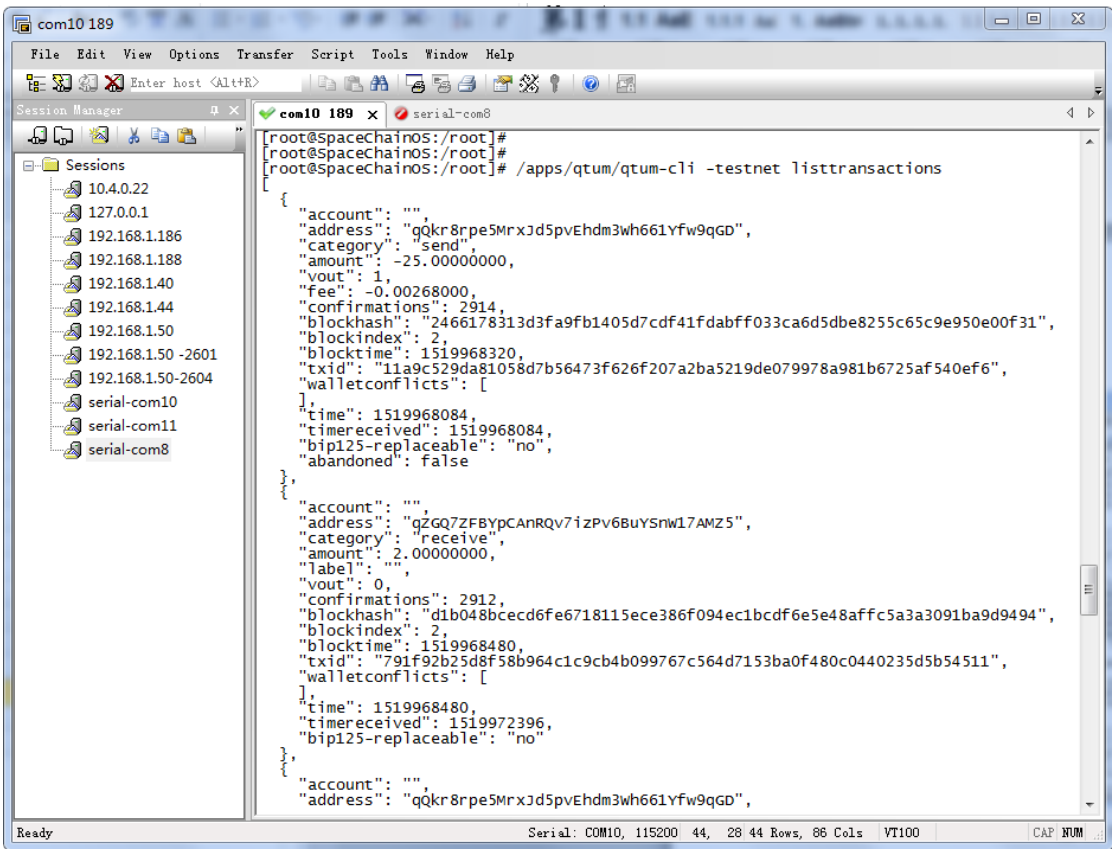

Figure 0.5 Command Demonstration of Transaction Query

3.1.6 Get the beneficiary address of this node

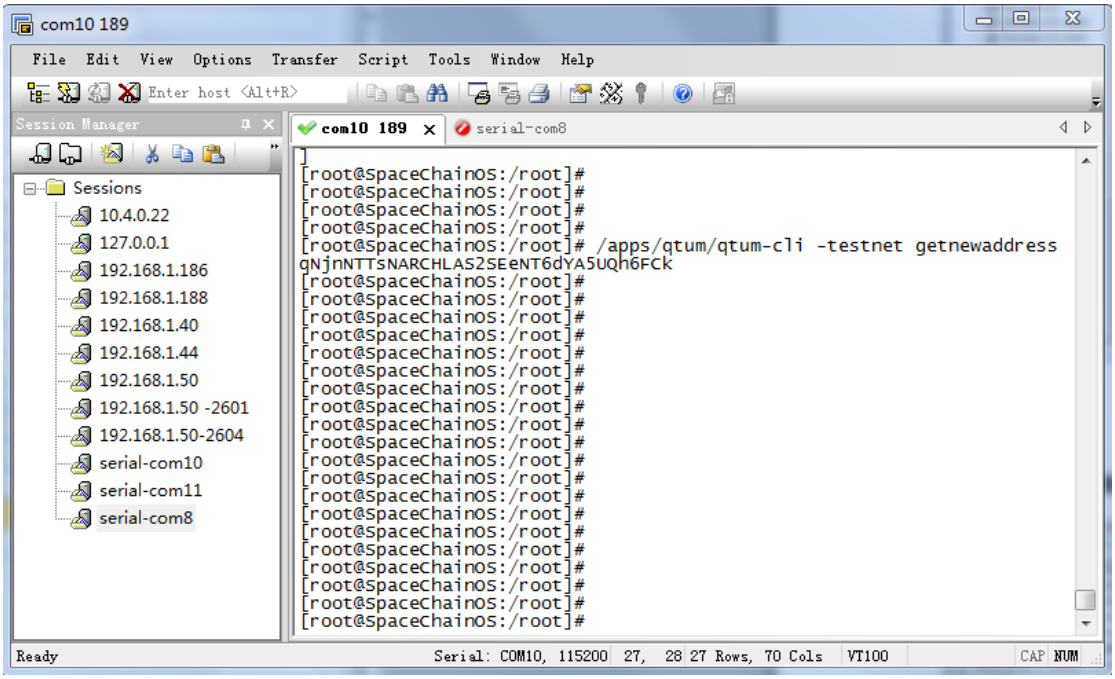

Figure 0.6 Command Demonstration for Get the Beneficiary Address of This Node