

# NETROCKEY6 SMART 用户手册

V1.29

版权所有© 2008 北京飞天诚信科技有限公司

<http://www.ftsafes.com.cn/>

## 北京飞天诚信科技有限公司

### 软件开发协议

北京飞天诚信科技有限公司（以下简称飞天）的所有产品，包括但不限于：开发工具包，磁盘，光盘，硬件设备和文档，以及未来的所有订单，都受本协议的制约。如果您不愿接受这些条款，请在收到后的 7 天内将完整的开发工具包寄回飞天，预付邮资和保险。我们会把货款退还给您，但要扣除运费和适当的手续费。

#### 1. 许可使用

您可以将本软件合并、连接到您的计算机程序中，但其目的只是如使用手册中描述的那样保护您的程序。您可以以备份为目的复制合理数量的拷贝。

#### 2. 禁止使用

除在条款 1 中特别允许的之外，不得复制、反向工程、反汇编、反编译、修改、增加、改进软件、硬件和产品的其它部分。禁止对软件和产品任何部分进行反向工程，禁止推导软件的源代码。禁止使用产品中的磁盘或光盘来传播、存储非本产品的原始内容的任何信息或由飞天提供的产品的任何升级。禁止将软件放在公共服务器上传播。

#### 3. 有限担保

飞天保证在自产品发给您之日起的 12 个月内，在正常的使用情况下，硬件和软件存储介质没有重大的工艺和材料上的缺陷。

#### 4. 修理限度

当根据本协议提出索赔时，飞天唯一的责任就是根据实际情况，免费进行替换或维修。飞天对被替换下来的任何产品部件都享有所有权。

保修索赔单必须在担保期内写好，在发生故障 14 天内连同令人信服的证据交给飞天。当将产品返还给飞天或飞天的授权代理商时，须预付运费和保险。

除了在本协议中保证的担保之外，飞天不再提供特别的或隐含的担保，也不再对本协议中所描述的产品负其它责任，包括它们的质量、性能和对某一特定目的适应性。

#### 5. 责任限度

不管因为什么原因，不管是因合同中的规定还是由于刑事的原因，包括疏忽的原因，而使您及任何一方受到了损失，由我方产品所造成的损失或该产品是起诉的原因或与起诉有间接关系，飞天对您及任何一方所承担的全部责任不超出您购买该产品所支付的货款。在任何情况下，飞天对于由于您不履行责任所导致的损失，或对于数据、利润、储蓄或其它的后续的和偶然的损失，即使飞天被建议有这种损失的可能性，或您根据第 3 方的索赔而提出的任何索赔均不负责任。

#### 6. 协议终止

当您不能遵守本协议所规定的条款时，将终止您的许可和本协议。但条款 2、3、4、5 将继续有效。

## 网络锁 (NetROCKEY6 SMART) 说明书

NetROCKEY6 SMART用于网络上的多用户软件保护，它一方面继承了ROCKEY6单机版本的功能，把ROCKEY6 的原有的功能通过UDP/TCP、IPX、NetBios 协议扩展到网络上，另一方面它实现了一把加密锁的多用户软件的保护工作。NetROCKEY6 SMART 提供了许多新的特性如允许多个服务器同时运行、与Linux 互连、提供强有力的测试监视工具等。用户可以通过NetROCKEY6 SMART 限制网络上软件运行数目、总的运行次数和限制可以使用的时间期限。下面是对网络锁详细介绍。

### 一 基本概念

#### 1.1 网络锁服务端

网络锁服务端是一个运行于 Win9X, 2K, XP, Win2003 平台上的服务程序，提供 TCP/IP, IPX, NetBIOS 协议下的网络锁服务。网络上可以启动一个或多个服务程序。每个服务程序有自己的唯一标志，比如 IP 地址，MAC 地址，主机名称等。服务端需要一把或多把 NetROCKEY6 SMART 网络锁和相应的驱动程序。

#### 1.2 网络锁客户端

网络锁客户端是一个动态链接库，提供接口函数，使软件可以像访问 ROCKEY6 那样访问 NetROCKEY6 SMART。它不需要任何驱动程序和真实的网络锁，通过与服务端通讯来访问位于服务端的真实的网络锁。客户程序可以被任何支持动态链接库的语言调用，如 VC, VB, Delphi 等。

#### 1.3 配置文件

服务程序的配置文件为Svrcfg6.ini，客户程序的配置文件为Clifcfg6.ini。服务程序和客户程序从配置文件中获得网络配置信息和其他运行参数。开发人员可以用文本编辑器和开发包提供的工具编辑配置这些参数，如启动的协议类型、响应时间、服务器地址和其他所需信息。

#### 1.4 记录文件

网络锁的记录文件Svrlog6.txt 记录服务程序的运行状态。当您的服务程序出现问题时就可参考记录文件里面的信息。记录文件的路径和名称可在Svrcfg6.ini 中配置，默认在服务程序的所在路径。

#### 1.5 端口和组

网络协议如UDP/TCP 和IPX 必须先确定一个固定的端口(Port)才能进行通讯。端口号从 0 到65535。NETROCKEY6 SMART 网络锁服务程序默认端口为4837，如果这个端口被其他程序占用，服务程序会报告端口绑定错误，这时可以将所有服务端和客户端的配置文件中端口值修改成同一个其他有效端口值。NetBios协议没有端口的概念，但它有组的概念。组的名字最多由16个字母构成，所有的FTNetServer 服务器属于同一组，组的名字可以在配置文件中设置，但必须保证所有的客户端和服务端配置文件组名一致。

#### 1.6 网络地址

各种协议中每台机器都有一个唯一的标志称为地址。UDP/TCP 采用 xxx.xxx.xxx.xxx 的形式例如 192.168.0.1，IPX 采用 6 字节网卡硬地址 (例如 00-35-4f-20-00-32)，NetBios 采用不超过 16 个字母作为地址如 FTNetServer。

## 1.7 搜索方式

搜索时除了读取配置文件的协议和端口信息外，还可以指定搜索方式。搜索方式主要分为三种，即自动搜索、手动搜索和半自动搜索。自动搜索采用指定协议的广播方式确定网络上有多少个服务器在运行并返回所有的找到的网络锁。手动搜索必须在配置文件里指定待搜索服务器的地址列表，然后逐一搜索。自动搜索比较方便因为不需要预先指定服务器的地址，这样可以适用于任何网络，但它需要一些搜索时间。手动搜索不需要搜索时间，直接和指定的服务器通讯，但前提是必须知道服务器在哪里。半自动搜索采取折衷的方法，先采用手动搜索寻找指定地址的服务，如果发现网络锁就返回，如果指定地址的服务没有开启或没有插入可用的网络锁会进一步采取自动搜索，并返回自动搜索的结果。

## 1.8 登录模块和登录方式

客户端登录时需要指定模块号和登录方式。模块号从 0 到 255，每个模块都可以独立限制登录人数，总登录次数和登录期限。除此之外我们还提供两种登录方式即私有方式和共享方式。私有方式为默认设置，每个进程登录都算一次有效登录。共享方式是指每一台计算机共享一个用户，在同一台计算机上不管多少进程登录都算同一个用户。第一种方式可以限制使用软件的进程数目，第二种方式下可以限制使用软件的计算机数目。

## 1.9 超时设定

客户端发送数据后需要等待服务端的响应，等待时间长短取决于超时设定。用户可以在配置文件里设定，单位为秒，默认值为三秒。自动搜索时客户端的等待时间也取决于这个数值。如果网络比较缓慢或网络比较繁忙可以修改配置文件增加超时时间。

## 1.10 最大使用用户数目，最大登录次数和登录期限

NetROCKEY6 SMART 的每个模块都可以独立限制最大用户数目，最大登录次数和最终登录期限。最大用户数目可以限制使用这个模块的客户数目，范围从 1—65534。65535 (0xFFFF) 表示不限制用户数目。最大登录次数限制此模块总的使用次数，范围从 1—65534。每进行一次登录，此数目递减 1，至 0 时登录失败。65535 (0xFFFF) 表示不限制登录次数。登录期限是一个时间值，由年月日组成，表示可以登录此模块的最终期限。如果写入 2003 年 10 月 1 日，则超出此时间后登录失败。这三个值可以用工具 Nr6Lic 生成授权文件写入锁中。

## 1.11 不响应时服务端的保留时间

客户端每隔 1.5 分钟会自动向服务端发送空消息以表明自己的存在。这种发送是自动的和透明的，程序不能进行干预。服务端每隔一定时间要检查一下是否所有客户端都发过空消息，如果发现有的客户没有发送，表明这个客户端程序在没有正常关闭句柄的情况下退出，这时服务端就会自动删除这个客户。这保证了客户端在死机或忘记关闭句柄或网络出现硬故障时不会使用 2 个或以上的账号登录服务器。定期检查的间隔时间称为不响应时服务端的保留时间，可以在配置文件中设置。

## 1.12 单个进程中打开模块的限制

在单个进程中对单个的模块只能打开一次，这主要是出于安全性的考虑。你可以登录后将句柄存放在全局变量里给所有线程使用。

## 1.13 母锁

网络锁生产时，需要写入授权文件。授权文件里包括最大登录用户数目，登陆次数，登录期限等信息。母锁就是用来产生授权文件的。对一个公司来说，它需要一把就够了，可以用母锁格式化工具将 NetROCKEY6 SMART 变成一把母锁。格式化时，需要输入一个主密钥，这个密钥独立于超级密码，非常重要，请与母锁一起妥善保存。母锁格式化后，会根据主密钥产生一个子锁格式化密钥，用于格式化子锁。

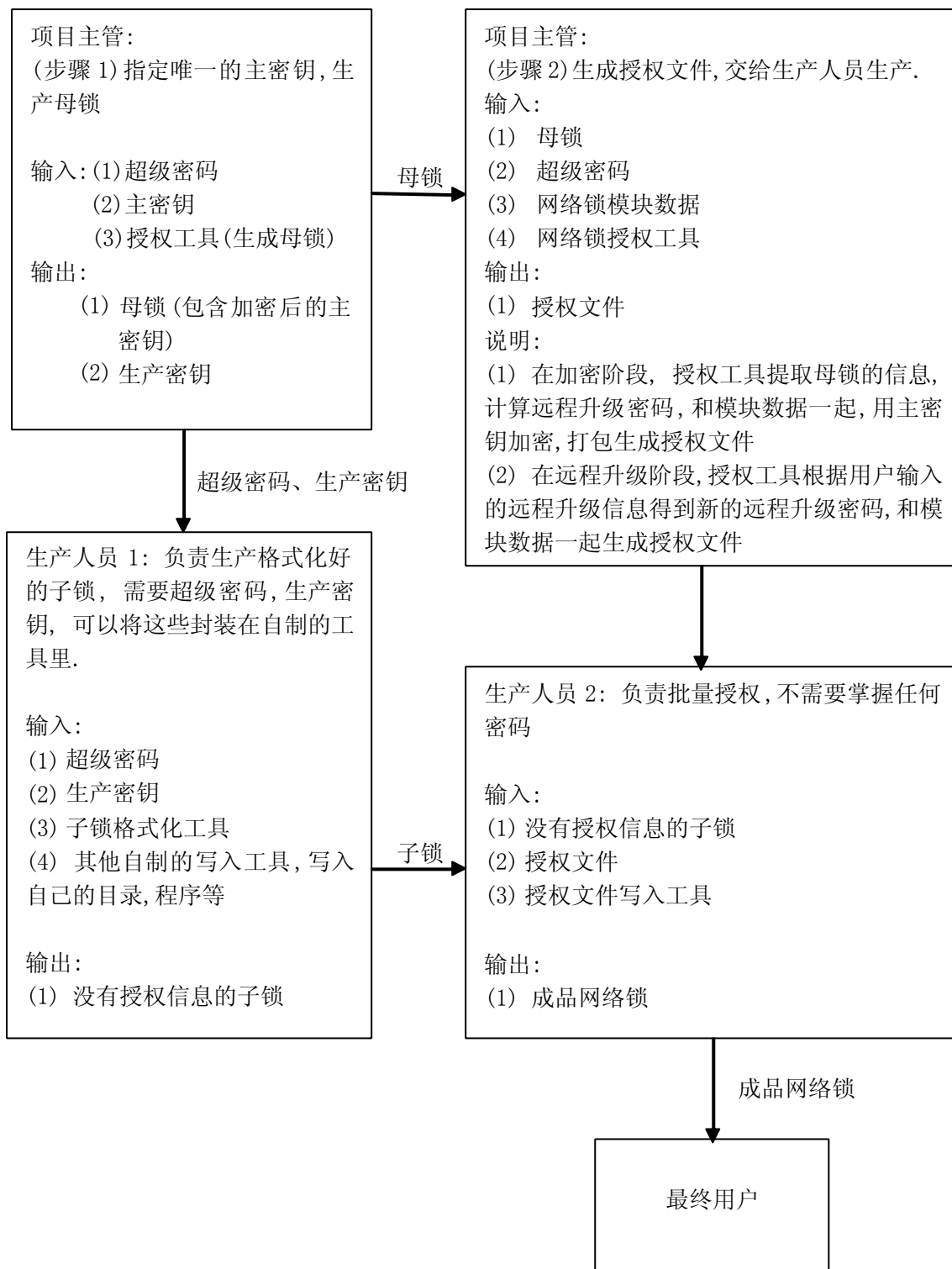
### 1.14 子锁

子锁是相对于母锁来说的，它需要子锁格式化工具来格式化。子锁需要批量的格式化，而且格式化时需要指定由母锁产生的生产密钥。经过格式化的子锁可以写入授权文件，经过这一个步骤后，就变成了一把彻底的网络锁。

## 二 网络锁生产流程

### 2.1 网络锁的生产流程

- (1) 管理人员首先要指定一个 8 字节长的主密钥。此密钥用于格式化母锁。然后使用 Nr6Lic 授权工具中的母锁格式化按钮格式化一把母锁。格式化完成后会生成一个子锁格式化用的密码，形式为一个字符串，称为生产密钥。将此密钥给生产人员 1。（参见下图）  
注意:妥善保管自己的主密钥和母锁，而生产密钥可以公开。
- (2) 管理人员使用授权工具和母锁生成授权文件，包括模块信息，登录次数，登录时间等，将此文件交给生产人员 2。
- (3) 生产人员 1 用子锁生产密钥，超级密码，子锁格式化工具批量格式化子锁，在格式化后，可以写入自己的程序，文件等。将处理好的子锁交给生产人员 2。
- (4) 生产人员 2 用授权文件写入工具批量处理子锁，做成成品网络锁。



## 2.2 远程升级流程

远程升级用于远程更新网络锁的登录限制信息, 如增加模块登录次数, 开启新的模块等。远程升级采用一次一密的方式, 升级密码可以公开, 但只能用于指定的一把锁, 而且只能使用一次。它可以即安全又简便升级网络锁的限制数据。具体的流程如下:

- (1) 最终用户决定要升级网络锁的限制信息后，比如决定要增加用户数目，使用次数，延长限制时间等，在最终用户端调用 Nr6Write.dll, 生成用于远程升级的初始密钥。此密钥包含锁的信息，是一个字符串。
- (2) 最终用户通过电话，email 等方式将初始密钥传到主管手中，主管用授权工具 Nr6Lic, 输入用户传来的密钥，追加或创建新的限制信息，生成一个授权文件。
- (3) 主管通过电话，email 等将授权文件或授权文件中的密钥串传给最终用户，在最终用户那里使用 Nr6Write.dll 更新加密锁信息。此授权文件只能用于产生初始密钥的锁，而且只能使用一次。

产品如果支持远程升级，需要做一个远程升级的界面，然后调用我们提供的 Nr6Write.dll, 完成如下功能：

- (1) 收集需要远程升级的加密锁硬件信息，生成初始密钥（通过 Nr6Write.dll 的 **GetUpdateInfo** 可以完成）
- (2) 将传来的授权文件写入加密锁（通过 Nr6Write.dll 的 **NrWriteModuleDirect**, **NrWriteModule** 可以完成）

具体的功能请参见 Nr6Write.dll 的介绍。

## 三 生产工具介绍

### 3.1 工具说明

NetROCKEY6 SMART 的生产分为 3 阶段，即生成授权文件阶段、格式化阶段和根据授权文件大批量生产阶段。格式化阶段可以将普通网络锁变为子锁。授权文件生成阶段需要利用母锁产生授权文件，这个文件包括模块信息，如模块号、最大用户数、最大登录次数和最终登录期限等。批量生产阶段需要将授权文件写入子锁，以变成最终的成品锁。不同的阶段可以由不同特权级的人负责，并被授予不同的工具。下面详细介绍这些工具。

### 3.2 授权文件生成工具（Nr6Lic.exe）

此工具位于 Tools/Nr6Licence, 负责将普通的网络锁变成母锁，和用母锁产生授权文件。主界面如下：



图 3-1 授权文件生成工具

在开始操作前,要先格式化一把或几把母锁,插入网络锁并点击格式化母锁按钮即可。格式化前需要确定一个8字节的主密钥,形式和超级密码相同。格式化后会产生母锁和子锁格式化密钥,请将母锁做好标记,妥善保管,一定要保证不能进行验证远程升级密码的操作,否则它会产生无效的授权文件。格式化的界面如下:



图 3-2 母锁格式化工具

然后就可以用主界面的其他按钮来生成授权文件了。操作前需要先插入母锁。按添加/修改按钮可以加入或修改列表中的模块信息。界面如下：

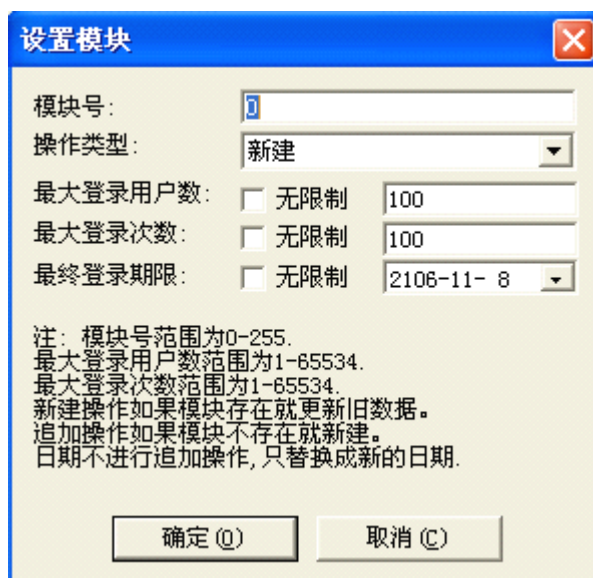


图 3-3 添加模块信息

模块号范围为 0—255，共 256 个。操作类型分为新建、追加和删除。新建操作会在锁中创建输入的信息块，供网络登录时使用。如果锁中的模块数据已存在，就会替换成新的数据。追加操作会将输入的数据追加到锁中的数据块中，如果锁中没有此模块的数据，就会新建一个数据块。需要注意的是时间不能追加，即在追加操作中会用新的时间替换旧的时间，这点和新建操作相同。除时间外，最大用户数和最大登录次数的追加遵循以下规则：

有限+有限=有限(如果和越界，则变为无限制)

有限+无限=无限

无限+有限=有限(用新的限制替换原来的无限制)

无限+无限=无限

选择删除操作会删除锁中的模块数据。

按下添加/修改按钮后数据会显示在列表里。这时加入新的模块，或修改已输入的模块（输入相同的模块号就可以修改），也可以通过删除模块按钮删除输入的模块。编辑好的数据可以存入磁盘中，以备后用。

如果为了给生产部批量生产，参数 1 可以选择从母锁获取，这时产生的授权文件只适用于刚格式化完成的网络锁，不能用于已经写入过授权文件的网络锁。但如果要远程升级，或更新已经写过一次授权文件的网络锁，可以选择手工输入，参见远程升级的介绍。设定完成后需要指定生成的文件名，按下生成密码按钮就可以生成授权文件。生成后的文件可以交给开发部批量处理，或邮寄给远程升级的客户。

密码文件中包括注释信息和生产密码，用 Notepad 等文本编辑工具打开后就可以看到。注释等可以删除，但可能会引起文件混淆。[Password] 节标题和 PWD 关键字绝不能删除或修改，否则会变为无效的密码文件。

从母锁获取远程升级信息生成的密码文件，只适用于刚用此母锁生成的生产密钥格式化完毕的网络锁，不适用已经写入过一次模块数据的网络锁。如果要更新已经写入过数据的网络锁，需要先提供这把网络锁的远程升级信息(Nr6Gen和Nr6Write.dll可以做到)，在Nr6Lic中选择手工输入。这样的密码文件只能使用一次，第二次写入就会失败，它采用一次一密的方式。只有再次获取远程升级信息，产成新的密码文件，才能再次更新。

### 3.3 子锁格式化工具(Nr6Fmt.exe)

这个工具可以将一般的网络锁锁格式化成子锁，然后将子锁移交给生产部批量生产。它需要超级密码和子锁格式化密码，并可以设定厂商、卷标信息。界面如下：



图 3-4 格式化工具

需要注意的是格式化密钥（生产密钥）一定要和母锁产成的生产密钥一样，否则会出现错误提示。

输入的信息可以保存成文件，以避免每次启动时重新输入。\*按钮可以隐藏输入的密码。可以一次插入多把锁，按下回车键即可。

### 3.4 批量生产工具 (Nr6Gen.exe)

此工具可以交给生产部批量生产网络锁。它需要插入格式化好的网络锁和指定由 Nr6Lic.exe 产生的密码文件名，界面如下：



图 3-5 批量生产工具

可以一次插入多把锁，按下回车键即可。

按下查看写入状态按钮可以查看任何一把子锁的写入状态。界面如下：

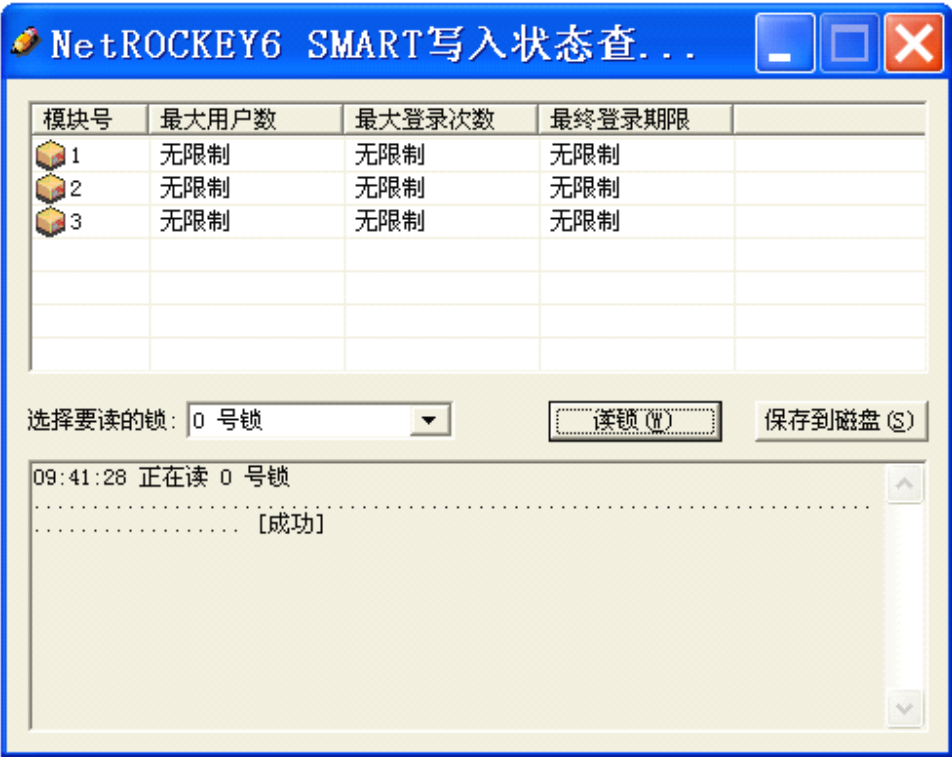


图 3-6 模块查看工具

选择要查看的锁并按下“读锁”按钮，程序就会读取所有模块并显示出来。还没有写入的模块和已被删除的模块将不会显示。也可以将读出的数据保存成 .mod 文件供 Nr6Lic.exe 使用。

### 3.5 简易生产过程

- (1) 用 Nr6Lic 格式化一把母锁，并记录子锁格式化密钥。
- (2) 用子锁格式化密钥和 Nr6Fmt，格式化几把子锁。
- (3) 插入母锁，运行 Nr6Lic，选择从母锁获取，输入测试用的模块信息和生成的密码文件名，生成授权文件。
- (4) 将母锁拔出保存好，插入一把或多把格式化好的网络锁，运行 Nr6Gen 程序，输入授权文件名，写入子锁，变成成品锁。
- (5) 将成品锁插入服务器，运行服务端程序和客户端测试程序，测试登录过程。

### 3.6 批量生产过程

批量生产涉及一系列负责人、生产者和最终用户：

A：管理者(1)，掌握超级密码和子锁格式化密钥，可以批量格式化子锁，写入其他运行程序。

B：管理者(2)，掌握超级密码和主密钥，可以生产母锁。

C：管理者(3)，只掌握超级密码，不掌握主密钥，需要 B 提供的母锁，可以决定写入锁中的模块信息，生成密码文件。

D：生产者，不掌握任何密码，可以使用 C 交付的密码文件和 A 交付的格式化过的网络锁批量生产。

E：最终用户，不掌握任何密码。需要 D 提供的网络锁，可以进行远程升级。

生产过程：

- (1) B 保管主密钥，并用 Nr6Lic 中的母锁格式化工具生产少量母锁，交给 C，并将子锁格式化密钥交给 A
- (2) A 用格式化密钥和子锁格式化工具批量产生子锁，并交给 D 处理。
- (3) C 得到母锁后，确定要写入的模块数据，并生成授权文件交给 D
- (4) D 得到 A, C 提供的锁和密码文件后，可以大批量生产，并交付 E 使用。
- (5) 如果 D 或 E 需要远程更新模块数据的内容，需要先向 C 提供远程升级信息，D 可以通过 Nr6Gen 获得一串字符密码，E 需要利用 Nr6Write.dll 来获得。字符密码邮给 C 后，C 可以根据此字符密码生成新的授权文件，并邮给 D 或 E。D 可以用 Nr6Gen 更新，E 可以利用 Nr6Write.dll 远程更新。

### 3.7 Nr6Write.Dll 说明

Nr6Write.dll 负责远程升级。它提供 3 个函数调用：

#### (1) int WINAPI NrWriteModule(int iIndex, LPCTSTR csPWFilename);

<b>功能</b>	通过密码文件远程更新模块信息。	
<b>参数</b>	iIndex	要写的锁的索引（句柄）
	csPWFilename	Nr6Lic 生成的授权文件的路径和名称
<b>返回</b>	返回 0 表示成功，其他为错误信息，参见 DIC32.h 和 Nr6Write.h 中得定义。	
<b>说明</b>	此函数被执行前需要 Find 和 Open，执行完毕后会自动关闭锁	

#### (2) int WINAPI NrWriteModuleDirect(int iIndex, LPCTSTR csPassword);

<b>功能</b>	通过密码字符串远程更新模块信息	
<b>参数</b>	iIndex	要写的锁的索引

	csPWFilename	Nr6Lic 生成的授权文件中的密码，可以在授权文件得 PWD 中取出。如密码文件中 PWD=ASD32-9AC，这里直接将 ” ASD32-9AC” 输入即可
<b>返回</b>	返回 0 表示成功，其他为错误信息，参见 DIC32.h 和 Nr6Write.h 中的定义。	
<b>说明</b>	此函数被执行前需要 Find 和 Open，执行完毕后会自动关闭锁	

### (3) int WINAPI GetUpdateInfo(int iIndex, char\* pRetBuf, int iBufLen);

<b>功能</b>	得到远程升级信息，用于远程升级	
<b>参数</b>	iIndex	锁的索引
	pRetBuf	返回的密码字符串缓冲
	iBufLen	缓冲区的最大长度
<b>返回</b>	返回 0 表示成功，其他为错误信息，参见 DIC32.h 和 Nr6Write.h 中的定义。密码会返回到 pRetBuf 中	
<b>说明</b>	此函数被执行前需要 Find 和 Open，执行完毕后会自动关闭锁	

这三个函数执行完毕后会自动将锁关闭。

## 四 开发包说明

NetROCKEY6 SMART的开发工具包含如下目录：

<API32>      32位网络锁API接口库  
 <ClientLib> 客户端动态链接库和配置文件  
 <Server>      服务端服务程序和配置文件  
 <Tools>      网络锁开发的辅助工具  
 <Samples>    开发样例  
 <Include>    开发包头文件  
 <Doc>        开发文档

下面我们先介绍一下每个部分的主要功能：

### 4.1 配置文件和配置工具

配置工具位于<Tools\Nr6Config> 子目录下，文件列表如下：

Nr6Config.exe      INI 文件编辑器用于服务端和客户端的配置文件编辑。  
 svrCfg6.ini        服务程序的配置文件。  
 CliCfg6.ini        客户端程序配置文件。

配置文件包括服务端svrCfg6.ini和客户端CliCfg6.ini 可以设定网络锁运行参数。服务端程序和客户端程序运行时运行目录下必须由各自的配置文件，否则按默认配置运行。配置文件中的所有字符均为大小写敏感，下面是客户端配置文件的模板：

```

[Header]
Sign = Nr6ClientHeader
; 客户端配置文件头
[Common]
Timeout =5
  
```

; 超时设定用户所有协议等待响应的时间单位为秒  
*SearchFlag =0*  
; 搜索标志0 表示自动搜索1 表示手动搜索2 表示半自动搜  
; 索即先手动搜索如果没有发现就进一步采取自动搜索  
; 如果采用手动搜索必须指定待搜索的服务器列表见下面每  
; 个协议项的*SearchList*  
*ProtoFlag=0*  
; 是否自动选择最快的协议: 1是, 0否。(选择此项开始时将会耗费一定的时间用于检测)  
*[TCPUDP]*  
*bUsedTCP =1*  
*bUsedUDP =1*  
; 是否使用TCP/UDP 通讯1 表示使用0 表示不使用  
*TCPPort =4837*  
; TCP 端口必须使用和服务端TCP 相同端口  
*UDPPort =4837*  
; UDP 端口必须使用和服务端UDP 相同端口  
*SearchList =192. 168. 0. 16, 192. 168. 0. 1, swordhui*  
; 手动搜索的搜索序列服务器的IP 或主机名地址用, 隔开  
*[IPX]*  
*bUsed =0*  
; 是否使用IPX 通讯1 表示使用0 表示不使用  
*IPXPort =4837*  
; IPX 端口必须使用和服务端IPX 相同端口  
*SearchList =00-A0-0C-13-0E-D2, 00-00-B4-B2-ED-7B*  
; 手动搜索的搜索序列为服务器网卡地址用, 隔开  
; 网卡地址可以用*nbtstat -a* 机器名得到, 也可以用我们的配置  
; 程序将主机名转换成网卡地址.  
*[NetBios]*  
*bUsed =0*  
; 是否使用NetBios 通讯1 表示使用0 表示不使用  
*RegGrpName =FTNetServer*  
; 服务器所在组的名称必须和服务端设置相同  
*SearchList=Book, swordhui*  
; 手动搜索的搜索序列为服务名字用, 隔开  
; 服务名字可以为服务器配置中的注册名也可以用主机名

服务端配置文件模板:

*[Header]*  
*Sign=Nr6SvrHeader*  
; 服务器配置文件标志  
*[common]*  
*Timeout=2*  
; 超时设定用于所有协议设定等待响应时间, 单位秒  
*IdleTime=3*  
; 客户端每隔1.5 分钟向服务器自动发送生存消息如果网线断  
; 开或客户忘记关闭句柄就退出程序此消息终止发送服务端

; 如果超过 *IdleTime* 仍得不到客户的自动生存消息就杀掉这个  
 ; 客户用于网线断开或客户端死机或客户端退出程序前忘关句柄  
*LogFile=svrlog6.txt*  
 ; log 文件名纪录服务器输出信息可以包含路径信息  
 [TCPUDP]  
*bUsed=0*  
 ; 是否启动TCP/UDP 服务1 表示启动0 表示不启动  
*TCPPort=4837*  
 ; TCP 端口, 如果被占用可以使用新的端口, 但要保证和所有客户端一致  
*UDPPort=4837*  
 ; UDP 端口, 如果被占用可以使用新的端口, 但要保证和所有客户端一致  
 [IPX]  
*bUsed=0*  
 ; 是否启动IPX 服务1 表示启动0 表示不启动  
*IPXPort=4837*  
 ; IPX 端口, 如果被占用可以使用新的端口, 但要保证和所有客户端一致  
 [NetBios]  
*bUsed=0*  
 ; 是否启动NetBios服务, 1表示启动, 0表示不启动  
*RegName=FTNetServer*  
 ; 是否注册服务器名称, 第一个默认值为FTNetServer001  
 ; 也可以自己指定名称, 如果发现重名就在后面加002, 003...  
*RegGrpName=FTNetServer*  
 ; 服务器所在组的名称。可以替换为自己公司的名称,  
 ; 但必须保证所有服务器的配置文件和所有客户端的配置文件着一项相同。

配置文件编辑器是一个图形界面程序, 可以用来编辑SvrCfg6.ini和CliCfg6.ini文件。  
 如果在当前路径下找不到svrCfg6.ini和CliCfg6.ini文件 图4-1 所示的窗口就会出现, 选择所需的配置文件后默认的配置文件就会被创建在当前路径下。

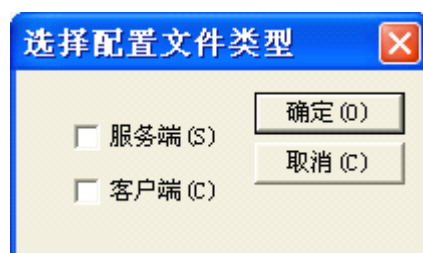


图 4-1 选择配置文件类型

配置工具可以编辑当前路径下的svrCfg6.ini和CliCfg6.ini, 界面如下:

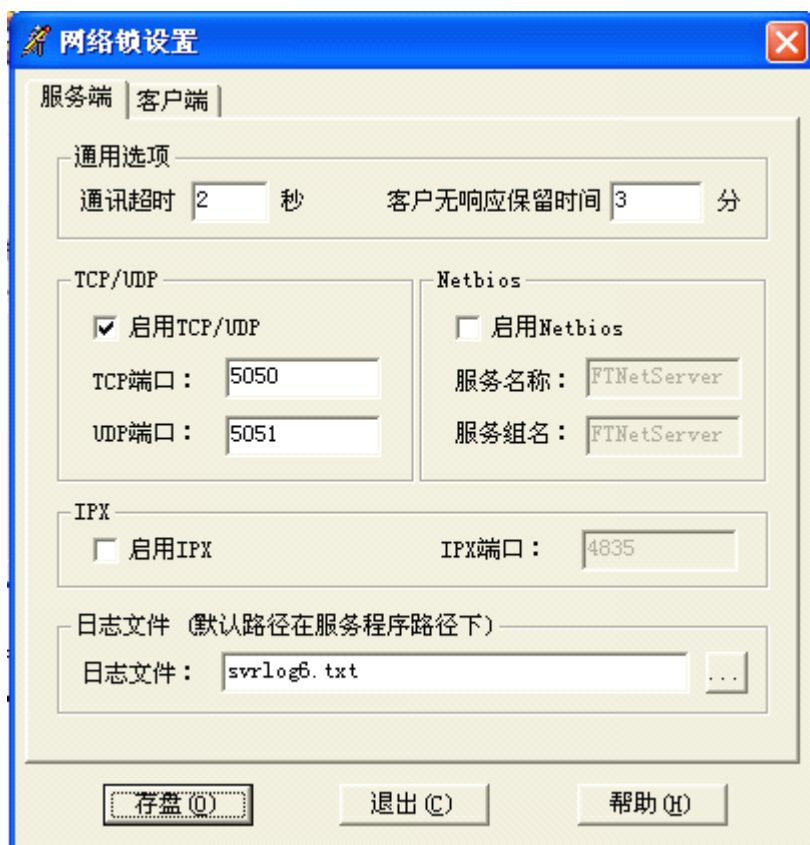


图 4-2 客户端配置

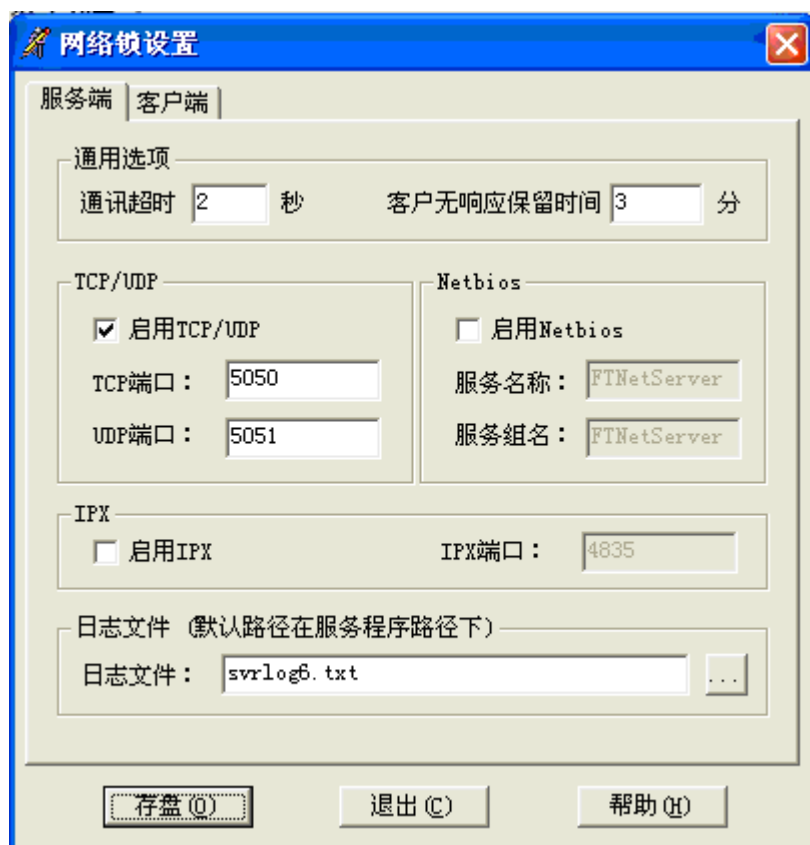


图 4-3 服务端配置

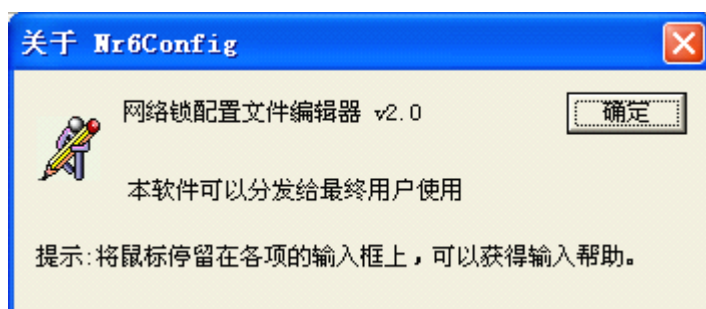


图 4-4 帮助信息

注：当前目录下如果只有svrCfg6.ini或CliCfg6.ini其中一个文件时，主窗口只有一个标签，这时你只能编辑当前目录下的文件。点击窗口左上角的图标会出现系统菜单，在导出命令里可以导出当前路径下没有的ini文件。

## 4.2 服务端程序

服务端程序位于<Server> 目录下。运行此程序的机器称为网络锁服务器，但在运行之前需要先安装ROCKEY6驱动程序，并插入已经写入授权文件的成品网络锁。

第一次运行后它会自动将自己注册成服务程序，每次开机自动启动，直到你卸载服务程序为止。启动后它会搜索当前路径下的svrCfg6.ini配置文件并按照配置参数启动，没有找到时就使用默认配置。服务器运行的信息写入到SvrCfg6.ini指定的log文件里，启动服务后会在系统任务区显示一个小图标如下图所示：



图 4-5 服务图标

在图标上双击鼠标左键会出现菜单如下图所示：



图 4-6 服务控制界面

在这里可以控制服务程序运行。点击终止会终止当前服务，点击启动会重新开始执行，点击卸载会卸载服务程序，下次启动时不会自动运行。右键菜单里有退出命令可以退出服务程序。

注意：

- (1) 服务程序需要NetROCKEY6 SMART 驱动程序和插入用Nr6Write写好模块的NetROCKEY6 SMART 锁。客户端不需要驱动程序和ROCKEY6 SMART锁。
- (2) 服务启动后不要拔插锁, 如果需要拔插请先退出服务，然后再拔插锁。否则客户端会出

现错误。

### 4.3 网络锁监视器

网络锁监视器位于<Tools\Nr6Mon> 目录下。监视器(Nr6Mon.exe)可以独立运行于网络的任何位置,不需要驱动程序和任何辅助D11。它用于监视网络上所有服务器和登录状况,还可以控制本机服务的启动关闭或强行踢出某个登录客户。如果Nr6Mon.exe 的执行目录下有svrCfg6.ini或CliCfg6.ini,它会自动读取这些文件中的端口信息进行网络连接。监视器启动后会自动使用指定的协议搜索寻找网络上启动的服务,可以通过设置菜单或工具条上的协议按钮来指定使用的协议。按钮的按下状态表示使用此协议,弹起状态表示不使用此协议。搜索的过程如下图所示:

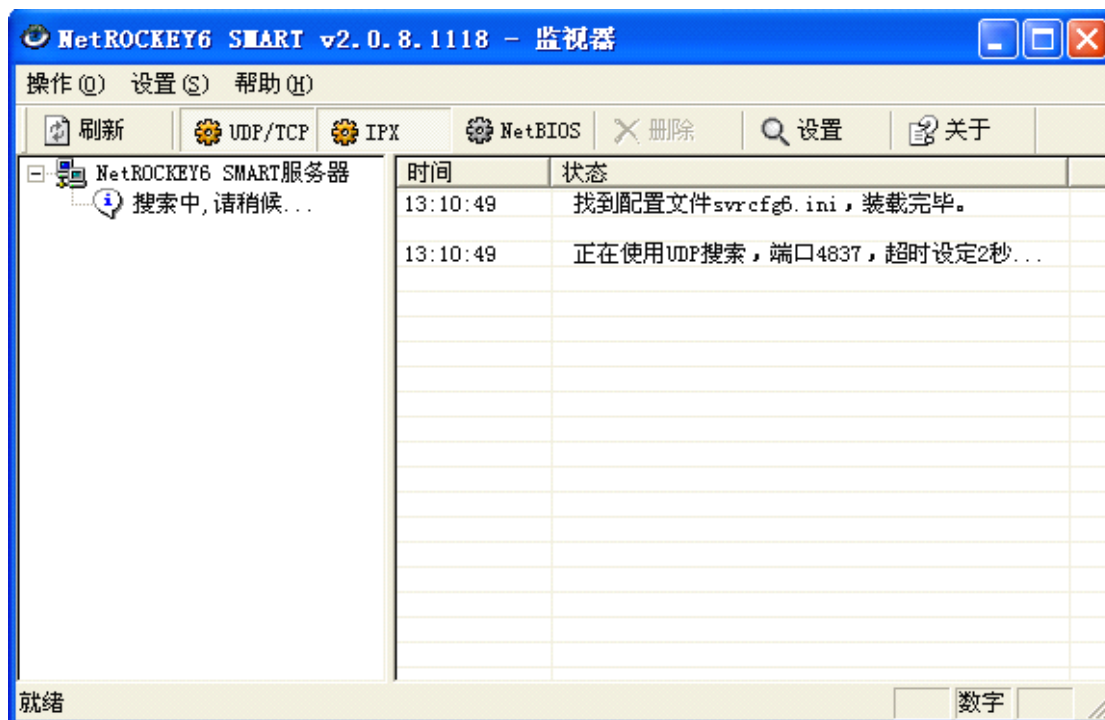


图 4-7 监视器搜索界面

如果某个协议出现错误会出现错误码提示，这时可以查阅说明书“错误代码”一节获得失败的原因。搜索完毕后会显示搜索结果如下：



图 4-8 监视器搜索结果

左边树状控件会显示出网络上所有的服务器和网络锁硬件ID。右边有服务器的详细信息包括服务器平台和开启协议等，如果发现本机启动了服务程序会在机器名后面出现”本机”的字样，这时可以通过监视器操纵本机服务，包括开启关闭某个协议或杀掉某个客户端等。这些操作可以在操作菜单或工具条中找到。

在ROCKEY6 SMART的硬件ID上点击鼠标左键会出现登录用户的信息，右边显示客户所在机器名称、平台、使用协议、登录模块、登录时间和句柄（注：此句柄为服务端句柄，和客户端句柄不一样，为在服务端的唯一标志）等各种信息。如果是登录本机服务的客户，选中后按Del键或选择工具条删除键可以强制删除。按F5键或选择工具条的刷新按钮会刷新当前显示的内容，除了手动刷新外还可以设置自动刷新，点击工具条的设置按钮会出现如下对话框：

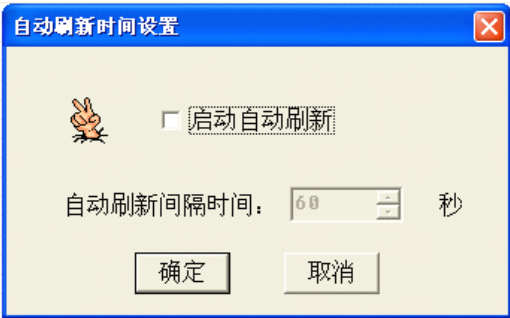


图 4-9 监视器自动刷新设置

选择启动自动刷新并输入以秒为单位的刷新闻隔时间即可。刷新按钮或自动刷新只刷新当前显示的信息，可以点击界面左侧的树状结构以显示和刷新不同的信息。例如：选中树状结构中的根，刷新网络上所有服务器信息，选中某个服务器刷新这个服务器的信息。选中某个ROCKEY6，则刷新所有用户信息。

如果本机启动了服务可以用操作菜单来开启或关闭指定协议的服务，选中登录本机的客户后将鼠标移到删除按钮它会变亮，这时按下删除按钮或Del键会强制删除这些客户。

4.4 客户端测试工具

客户端测试工具位于<Tools\Nr6Test> 目录下，主要用于测试网络锁的所有功能。Nr6Test.exe 可以读取当前路径下的客户端配置文件进行网络锁测试。测试网络锁时它需要Nr6Cli.dll 和 CliCfg6.ini 两个文件放到当前路径下。它不需要ROCKEY6的驱动程序，可以在网络的任何机器上运行。

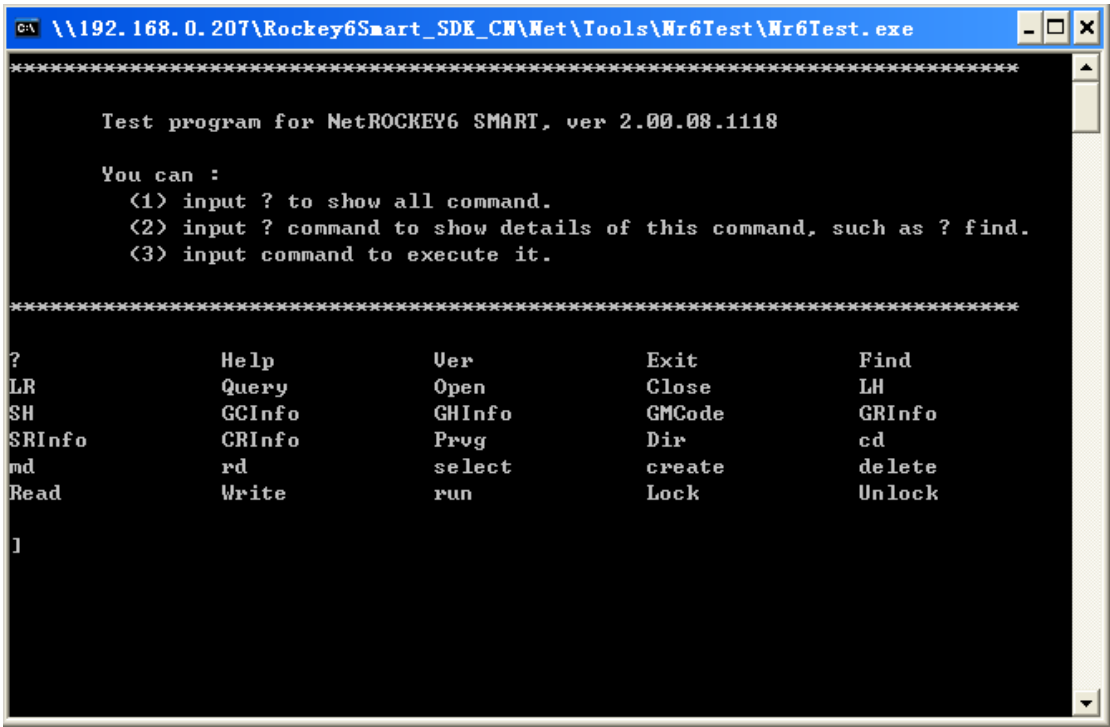


图 4-10 网络锁测试工具

五 API 调用说明

5.1 int DICNet\_Find(WORD Zone,WORD Agent,WORD User1,WORD User2);

类别	内容
功能	在网络上查找指定的加密锁设备，或查找所有的加密锁设备。
输入	Zone 为区域编码, Agent 为代理编码, User1, User2 为用户编码。如果指定这四个值会返回相应的锁，如果全设为 0，会返回网络上所有的锁。
返回值	如果返回值为 0，表示没有找到相应的加密锁设备，可能是服务没有开启或服务端没有配置好。如果返回值大于 0，返回的是网络上可登录设备。这时可以用 DICNet_Query 来获得锁的相关信息。也可以用 DICNet_Open 来登录。
使用说明	此命令执行时会从客户端配置文件 Clicfg6.ini 读取搜索方式信息。如果采用自动搜索方式，会用广播的方式在网络上找所有的服务器。这样它的搜索时间受超时设定的影响。如果超时设定的时间比较长，搜索等待的时间也比较长。如果采用手动的搜索方式，即指定搜索列表，可以迅速返回指定服务器的网络锁讯息，不受超时设定的影响。

5.2 int DICNet\_Query(int iIndex, int iQueryType, void \*pRet);

类别	内容	
功能	查询 Find 找到的锁的信息。	
输入	iIndex	为找到设备的索引，实际上就是 1 个数字，0，1，2，3，4，5，6，7 的序号，这个序号应该小于 Find 的返回值。
	iQueryType	支持三种方式，NR_QUERY_SVRNAME 表示查询服务器的机器名称（即在网上邻居里看到的名称），这时 pRet 需要送入一个不小于 16 字节的字符串地址。NR_QUERY_HID 为查询加密锁的硬件 ID（即硬件序列号，为锁的唯一标志），这时 pRet 需要送入一个 DWORD 的地址。NR_QUERY_CARDINFO 为查询锁的卷标、厂商信息，pRet 为一个 DICST_CardInfo 结构的地址。此命令和 GET_CARD_INFO 的返回值完全相同，请参见下面命令中的 GET_CARD_INFO 说明。
	pRet	返回的信息，与 iQueryType 相关，参见 iQueryType 说明。
返回	返回的是一个错误编码，具体含义请看 6.1。	

### 5.3 int DICNet\_Open(DWORD iIndex,DWORD iModule,DWORD\* pHandle);

类别	内容	
功能	登录指定的网络锁。	
输入	iIndex	为找到设备的索引，实际上就是 1 个数字，0，1，2，3，4，5，6，7 的序号，这个序号应该小于 Find 的返回值。
	iModule	低字节为要登录的模块号，范围 0-255，参见 Nr6Lic 工具的说明部分。高字节为登陆方式，1 为共享登陆，0 为私有登陆（见 1.8）
	pHandle	为一个 DWORD 的地址，登录成功后返回一个句柄，用来做后继操作。
返回	返回的是一个错误编码，具体含义请看 6.1。成功后句柄会在 pHandle 中返回。	
使用说明	这个命令可以登录指定的网络锁。登录受最大登录用户，最大登录次数，最终登录期限的限制。如果不满足模块的限制条件，登录会失败。限制条件可以用 Nr6Lic 设置。如果该模块限制了最大登录次数，每次登录，不管成功还是失败，都会使该模块的最大登录次数减 1。	

### 5.4 int DICNet\_Close(DWORD dwHandle);

类别	内容	
功能	关闭指定的网络锁	
输入	锁的句柄，DIC_NetOpen 中 pHandle 的返回值。	
返回	返回的是一个错误编码，具体含义请看 6.1。	
使用说明	关闭网络连接，释放增加的用户数。	

### 5.5 int DICNet\_Command(DWORD dwHandle,int cmd,void\* data);

类别	内容	
功能	完成对指定网络锁的操作。	
输入	hic	锁的句柄，DIC_NetOpen 中 pHandle 的返回值。
	cmd	具体的操作，用不同的常数代表不同的操作，具体规定见 [cmd 说明]

	data	跟命令有关的输入/输出数据，具体见[特殊的 data 数据结构]的说明
返回	返回的是一个错误编码，具体含义请看 6.1。	
使用说明	这是一个多功能命令，所有跟设备有关的操作都由此命令来完成。	

[cmd 说明]

每一个 cmd 常数已经在 Nr6Cli.h 文件中定义为一个宏，代表不同的操作。特别说明的是这些宏有可能和不同集成开发环境中预定义的宏冲突，而有些预编译是不会报告这种冲突的。如果书写格式正确但不能够完成相应的功能的情况，请以 Dic32.h 中相应的常数为参数或把相应的宏改变名称后再使用。

宏	Data 数据结构	执行的操作
GET_CARD_INFO	DICST_CardInfo	取得 IC 卡的卷标和厂商信息
GET_HARDWARE_INFO	DICST_HardInfo	取得硬件的生产时间，序列号，销售时间，和 COS 版本
GET_MANAGER_CODE	DICST_ManagerCode	取得管理编码，包括区域编码，代理商编码和两个用户编码
GET_CARD_PRIVILEGE	一个字节	取得卡的当前密码验证状态。有三个返回值:0(密码没有验证)，4(超级密码验证通过)，8(远程升级密码验证通过)
GET_REMOTE_INFO	DICST_RemoteInfo	取得远程升级信息
SET_REMOTE_INFO	DICST_RemoteInfo	设置远程升级信息
CHECK_REMOTE_INFO	DICST_RemoteInfo	验证远程升级信息，此命令需要预先执行 NET_LOCK 锁定句柄。验证成功后会进入特权状态进行操作。
GET_CURRENT_DIR	DICST_Dir	取得当前目录信息
SET_CURRENT_DIR	DICST_Dir	设置当前目录信息
GET_CURRENT_FILE	DICST_File	取得当前文件信息
SET_CURRENT_FILE	DICST_File	设置当前文件信息
GET_PARENT_DIR	DICST_Dir	取得上一级目录
SET_PARENT_DIR	DICST_Dir	设置上一级目录
LIST_DIR	输入为一个字节的文件索引，从 0 开始。输出为 DICST_Dir 或 DICST_File 结构。	根据索引返回文件系统中的文件信息。返回的结构中的属性值来确定是目录还是文件。
READ_FILE	DICST_Before_Read_Data 和 DICST_After_Read_Data	读取当前文件内容
WRITE_FILE	DICST_Write_Data	写入文件内容
CREATE_DIR	DICST_Dir	创建目录
CREATE_FILE	DICST_File	创建文件
REMOVE_DIR	无	删除当前选定的目录
REMOVE_FILE	无	删除选定的文件
RANDOM	输入为一个字节的个数，输出为 BYTE 数组，返回相应个数的随机数。(最大不超过 16 个)	取得随机数

RUN	DICST_Before_Read_Data 和 DICST_After_Run_Data	运行可执行文件
DESENC	DICST_Des_Data 和 DICST_After_EncDec_Data	DES 加密
DESDEC	DICST_Des_Data 和 DICST_After_EncDec_Data	DES 解密
RSAENC	DICST_Rsa_Data 和 DICST_After_EncDec_Data	RSA 加密
RSADec	DICST_Rsa_Data 和 DICST_After_EncDec_Data	RSA 解密
GETFREESPACE	返回值为一个 DWORD	得到文件系统剩余空间
STPCOUNTER	无	单步递减使用记数
NET_LOCK	无	锁定网络锁，进行连续操作。一般用于验证远程升级密码。操作成功后网络锁只能由当前句柄使用，其他句柄一概访问失败，包括登录操作。此命令如果特别需求，尽量少使用。
NET_UNLOCK	无	解除锁定。
NET_GETPATH	一个 WORD 数组，不少于 256 个 WORD 值。	得到当前的路径信息。数组中下标0表示总个数，下标1开始为目录 ID，得到当前的路径。

[特殊的 data 数据结构]

所有对数据结构的操作都可以用 DIC\_Get 和 DIC\_Set 中的模式操作来完成，一般情况下没有必要关心这些数据结构。

```
=====
DICST_CardInfo  /*由开发商设置*/
```

```
typedef struct{
    char volume[16];      //IC 卡的卷标
    char atr[15];         //厂商信息
} DICST_CardInfo;
```

```
=====
DICST_HardInfo  /*这个结构的信息由生产厂商设置，开发商无权改动*/
```

```
typedef struct{
    DWORD FactoryTime;    //生产时间
    DWORD HardSerial;     //序列号
    DWORD ShipTime;       //销售时间
    DWORD COSVersion;     //COS 版本
} DICST_HardInfo;
```

```
=====
DICST_ManagerCode
```

```
typedef struct{
    WORD Zone;            //区域编码
    WORD Agent;           //代理商编码
    WORD User1;           //用户编码 1
    WORD User2;           //用户编码 2
}
```

```

} DICST_ManagerCode;
=====
DICST_RemoteInfo    /*由开发设置和解释*/
typedef struct{
    DWORD RemoteTag;        //远程升级标志
    BYTE RemotePass[8];     //远程升级密码
} DICST_RemoteInfo;
=====
DICST_Dir
typedef struct {
    WORD        dirid;        // 目录 ID
    BYTE        dircla;       // 目录类别
    BYTE        diratrpri;    // 目录属性 & 目录安全级
    WORD        dirsize;      // 没用
    char        dirname[16];  // 目录名
} DICST_Dir;
=====
DICST_File
typedef struct {
    WORD        fileid;       // 文件 ID
    BYTE        filecla;      // 文件类别
    BYTE        fileatrpri;   // 文件属性 & 文件安全级
    WORD        filesize;     // 文件大小
    char        filename[17]; // 文件名
} DICST_File;
diratrpri(目录属性)和 fileatrpri(文件属性)的定义

```

标志	类型
FILEATTR_NORMAL	一般的数据文件
FILEATTR_EXEC	表示可执行
FILEATTR_DIR	表示为目录
FILEATTR_QUIGORE	表示如果当前安全级别已经高于可执行程序自身的级别， 则不执行本程序
FILEATTR_INTERNAL	表示为内部应用程序自己使用的文件，不能从外部进行操作。 如果一个可执行文件被标志为内部文件，它就具有了 隐含的属性，在列目录时，只有通过了超级密码验证才能 看到这些隐含的可执行文件。

```

=====
DICST_Upgrade_RemotePass    /*在远程升级管理中使用*/
typedef struct
{
    DWORD RemoteTag;    //远程升级标志
    DWORD HardSerial;   //硬件序列号
    BYTE RemotePass[8]; //远程升级密码
} DICST_Upgrade_RemotePass;
=====

```

DICST\_Before\_Read\_Data

```
typedef struct
{
    WORD        offset;
    WORD        size;
} DICST_Before_Read_Data;
```

=====

DICST\_After\_Read\_Data

```
typedef struct
{
    WORD        readedsize;
    char        buffer[1];    // 大小为 readsize
} DICST_After_Read_Data;
```

=====

DICST\_Write\_Data

```
typedef struct
{
    WORD        offset;    //文件的偏移量
    WORD        size;      //文件的大小
    char        buffer[1];    // 大小为 size
} DICST_Write_Data;
```

=====

DICST\_System\_Info /\*格式化之后要设置卡的卷标和厂商信息\*/

```
typedef struct
{
    char        volume[16];    //卷标
    char        atr[15];      //厂商信息
} DICST_System_Info;
```

=====

DICST\_Before\_Run\_Data

```
typedef struct
{
    WORD RunID;    //文件的 ID
    WORD ParaSize;    //参数的大小
    BYTE Para[1];    // 大小为 ParaSize
} DICST_Before_Run_Data;
```

=====

DICST\_After\_Run\_Data

```
typedef struct
{
    WORD ResultSize;    //返回参数的字节数
    BYTE Result[1];    // 大小为 ResultSize
} DICST_After_Run_Data;
```

=====

DICST\_Des\_Data

```
typedef struct
{
    //对加密来说    //对解密来说
```

```

    WORD dataLength;           //加密数据的长度 //解密数据的长度
    WORD KeyFileid;           //DES密钥文件的文件ID
    BYTE KeyIndex;            //本次加密使用的密钥在DES密钥文件中的标识,
                              //注意一个DES密钥文件包含多个DES加解密使用的密
    钥

                              //各次解密使用的密钥用KeyIndex来标识
    char data[1];             //加密数据           //解密数据
} DICST_Des_Data;
=====
DICST_Rsa_Data;
typedef struct
{
    //对加密来说           //对解密来说
    WORD KeyFileid;         //公钥文件的文件ID //私钥文件的文件ID
    WORD dataLength;         //加密数据的长度 //解密数据的长度
    char data[1];           //加密数据           //解密数据
} DICST_Rsa_Data;
=====
DICST_After_EncDec_Data
typedef struct
{
    //对加密来说           //对解密来说
    WORD dataLength;         //加密后返回数据的长度 //解密后返回数据的长度
    char data[1];           //加密后返回数据       //解密后返回数据
} DICST_After_EncDec_Data;

```

## 5.6 DWORD SetIniPathName(LPCTSTR iniName);

类别	内容
功能	设置客户端配置文件路径。设定后 Find 会根据新的配置信息搜索网络。默认为当前路径下的 Clicfg6.ini 文件。
输入	IniName 是文件名，包含路径信息。
返回	返回的是一个错误编码，具体含义请看 6.1。

## 5.7 DWORD NrGetLastError();

类别	内容
功能	如果发生网络错误时，得到详细的与协议有关的错误代码。
输入	无
返回	返回的是一个错误编码，具体含义请看 6.3。

## 5.8 int DIC\_Get(void\* xdata, int p1, int p2, char\* buffer);

类别	内容
功能	从 API 的返回数据中提取出单项数据，适用于没有结构定义的编程语言。对于支持结构定义的语言，比如 C, C++, Pascal 等，可以直接使用上面定义的结构。
输入	xdata    提取的目标缓冲区
	p1       模式/偏移，最高位为 1 表示这是模式操作，否则是用户自定义的偏移值。详细说明参见附录 H5.
	p2       返回方式/尺寸，最高 2 位表示返回的方式，剩余的位表示操作的尺寸。详细说明参见附录 H5.

	Buffer	提取出数据的字符型缓冲区
返回	返回的是一个错误编码，具体含义请看 6.1。	

### 5.9 int DIC\_Set(void\* xdata, int p1, int p2, int p3, char\* buffer);

类别	内容	
功能	设置 API 输入数据中的单项数据，适用于没有结构定义的编程语言。对于支持结构定义的语言，比如 C, C++, Pascal 等，可以直接使用上面定义的结构。	
输入	xdata	存储的目标缓冲区
	p1	模式/偏移，最高位为 1 表示这是模式操作，否则是用户自定义的偏移值。详细说明参见附录 H6.
	p2	传入方式/尺寸，最高 2 位表示传入的方式，剩余的位表示操作的大小。详细说明参见附录 H6.
	P3	通常是用户要传入的值。详细说明参见附录 H6.
	Buffer	准备存储的字符型缓冲区。详细说明参见附录 H6.
返回	返回的是一个错误编码，具体含义请看 6.1。	

## 六 错误代码

NetRockey6 SMART 错误码分为三类：常规错误码，网络错误码和网络扩展错误码。常规错误码为 API 的返回值，主要指是函数调用的错误。当网络出现故障时会返回网络错误码，这时如果想获得更详细的协议错误信息，可以用 NrGetLastError 函数获得扩展错误码。

### 6.1 常规错误码

代码	意义
0x00000000	成功，没有错误
0x80100001	内部连接检查失败
0x80100002	操作被用户中止
0x80100003	不正确的操作句柄
0x80100004	不正确的参数
0x80100005	注册的启动信息丢失或无效
0x80100006	没有足够的内存用于完成命令
0x80100007	内部超时
0x80100008	用户给出的缓冲区太小，不足以放下全部的返回数据
0x80100009	未知的读卡器
0x8010000A	用户指定的时间超时
0x8010000B	卡正在被其它连接占用
0x8010000C	在读卡器里面没有卡
0x8010000D	未知的卡类型
0x8010000E	读卡器无法完成退出卡操作
0x8010000F	当前的卡不支持用户指定的通讯协议
0x80100010	卡还没有准备好接收命令

0x80100011	某些变量的值不合适
0x80100012	操作被系统中止，可能是重新登录或关机
0x80100013	内部通讯错误
0x80100014	内部未知错误
0x80100015	无效的厂商信息
0x80100016	用户尝试结束某个不存在的处理
0x80100017	指定的读卡器当前无法使用
0x80100018	操作被中止，允许服务程序退出
0x80100019	PCI 的接收缓冲区太小
0x8010001A	读卡器的驱动无法支持当前的读卡器
0x8010001B	读卡器的驱动程序无法建立唯一的名字，已经有相同名字的读卡器存在
0x8010001C	卡无法被当前的读卡器支持
0x8010001D	智能卡服务没有开启
0x8010001E	智能卡服务已经被中止
0x8010001F	某个意外的智能卡错误产生
0x80100020	无法获知智能卡的提供者信息
0x80100021	无法获知智能卡的生产者信息
0x80100022	当前的智能卡无法支持用户要求的功能
0x80100023	指定的目录不存在
0x80100024	指定的文件不存在
0x80100025	指定的目录不再是有效的目录
0x80100026	指定的文件不再是有效的文件，没有选择文件
0x80100027	此文件拒绝访问
0x80100028	卡的空间已满，无法再写入信息
0x80100029	设置文件指针错误
0x8010002A	PIN 码错误
0x8010002B	一个无法识别的错误码从智能卡服务返回
0x8010002C	请求的证书不存在
0x8010002D	请求的证书不允许获得
0x8010002E	找不到任何一个读卡器
0x8010002F	智能卡通讯过程中发生数据丢失，请再次尝试
0x80100030	请求的密钥文件不存在
0x80100065	由于厂商信息配置冲突，读卡器无法跟卡通讯
0x80100066	卡对复位没有响应
0x80100067	卡没有电
0x80100068	卡被复位了，因此共享的信息无效了
0x80100069	卡已经被移出了
0x8010006A	因为安全规则，访问被拒绝了
0x8010006B	PIN 码没有被验证，访问被拒绝
0x8010006C	已经到达最大 PIN 码验证次数，访问被拒绝
0x8010006D	已经到达最后的智能卡文件，没有更多的文件可以访问了
0x8010006E	操作被用户中止
0x8010006F	智能卡 PIN 没有设置

0xA0100001	文件已经存在
0xA0100002	卡内存储器操作出错
0xA0100003	用户给出了无效的 CLA
0xA0100004	用户给出了无效的 INS
0xA0100005	虚拟机地址超界/异常
0xA0100006	除 0 错
0xA0100007	卡没有被插入到正确的位置
0xA0100008	卡当前处于某种未知的状态
0xA0100009	卡还没有被打开
0xA010000A	未知的命令
0xA010000B	将设定超级密码的重新设置次数是 0
0xA010000C	打开了太多的设备
虚拟设备专用错误码	
0xA0101001	创建虚拟卡文件失败
0xA0101002	打开虚拟卡文件失败
网络锁专用代码	
0xA0100101	Dic32u.dll发生异常
0xA0100102	网络错误
0xA0100103	没有找到Dic32u.dll
0xA0100104	参数错误
0xA0100105	模块定义文件没有找到
0xA0100109	服务端CacheSize 出现错误。
0xA010010A	服务端的ROCKEY6设备损坏或被拔走
0xA010010B	服务端不支持此API
0xA010010C	相同的模块号在相同的进程中被打开
0xA010010D	网络传输错误，用NrGetLastError
0xA010010E	无效的句柄，可能是此句柄已经关闭
0xA010010F	网络锁硬件损坏
0xA0100110	客户端D11被修改，拒绝提供服务
0xA0100111	服务端程序被修改，不再提供服务
0xA0100112	用户当前路径被其他用户删除，
0xA0100113	网络上找不到锁
0xA0100114	网络上找不到更多的锁
0xA0100115	已经被其他句柄锁定。
0xA0100116	此命令需要先锁定
0xA0100117	登录时数据校验失败
0xA0102001	输入的 TOKENINFO 校验出错
0xA0102002	服务变动，输入的 ROOTKEY 与卡内不一致

0xA0102003	同一个 TOKENINFO 重复登录
0xA0102004	模块号不匹配
0xA0102005	模块号超出合法范围
0xA0102006	指定模块数据找不到
0xA0102007	指定模块数据已经删除
0xA0102008	卡内模块数据校验值错
0xA0102009	已达最大用户数
0xA010200A	超过允许登录期限
0xA010200B	允许登录次数已减为 0
0xA010200C	USERINFO 校验值不正确

## 6.2 网络错误码

宏	代码	含义
ERR_INIT_SOCKET	2001	初始化Winsock出错
ERR_NOSUCHPROTO	2002	没有与制定协议对应的服务
ERR_UDPSOCKETCREATE	2003	UDP创建套接字失败
ERR_UDPSETBROADCAST	2004	UDP设置广播失败
ERR_UDPBINDFAILED	2005	UDP绑定失败
ERR_SVRCALLBACKNULL	2006	服务端回调函数为空
ERR_TCPSOCKETCREATE	2007	TCP创建套接字失败
ERR_TCPBINDFAILED	2008	TCP绑定失败
ERR_TCPLISTENFAILED	2009	TCP侦听失败
ERR_NOSUCHSEACH	2010	不支持的搜索方式
ERR_UDPSEND	2012	UDP发送数据失败
ERR_UDPTIMEOUT	2013	UDP等待数据超时
ERR_UDPrecv	2014	UDP接收数据失败
ERR_TCPCONNECT	2015	TCP连接服务器失败
ERR_TCPSENDTIMEOUT	2016	TCP等待发送超时
ERR_TCPSEND	2017	TCP发送数据失败
ERR_TCPRECVTIMEOUT	2018	TCP等待接收超时
ERR_TCPRECV	2019	TCP接受数据失败
ERR_IPXSOCKETCREATE	2020	IPX创建套接字失败
ERR_IPXSETBROADCAST	2021	IPX设置广播失败
ERR_IPXSEND	2022	IPX发送数据失败
ERR_IPXRECV	2023	IPX接收数据失败
ERR_IPXBIND	2024	IPX绑定端口失败
ERR_NBSRESET	2025	NBS初始化信道失败
ERR_NBSADDNAME	2026	NBS加入名字失败
ERR_NBSEND	2027	NBS发送失败
ERR_NBSRECV	2028	NBS接收失败

### 6.3 扩展错误码

当 API 返回网络错误时就可以用 `NrGetLastError` 函数得到详细的错误原因。从网络错误码可以断定是哪种协议出现了错误，下面将 `NrGetLastError` 的有可能的返回值列出来供参考：

UDP/TCP 和 IPX 的扩展错误码参考

宏	代码
WSABASEERR	10000
WSAEINTR	(WSABASEERR+4)
WSAEBADF	(WSABASEERR+9)
WSAEACCES	(WSABASEERR+13)
WSAEFAULT	(WSABASEERR+14)
WSAEINVAL	(WSABASEERR+22)
WSAEMFILE	(WSABASEERR+24)
WSAEWOULDBLOCK	(WSABASEERR+35)
WSAEINPROGRESS	(WSABASEERR+36)
WSAEALREADY	(WSABASEERR+37)
WSAENOTSOCK	(WSABASEERR+38)
WSAEDESTADDRREQ	(WSABASEERR+39)
WSAEMSGSIZE	(WSABASEERR+40)
WSAEPROTOTYPE	(WSABASEERR+41)
WSAENOPROTOOPT	(WSABASEERR+42)
WSAEPROTONOSUPPORT	(WSABASEERR+43)
WSAESOCKTNOSUPPORT	(WSABASEERR+44)
WSAEOPNOTSUPP	(WSABASEERR+45)
WSAEPFNOSUPPORT	(WSABASEERR+46)
WSAEAFNOSUPPORT	(WSABASEERR+47)
WSAEADDRINUSE	(WSABASEERR+48)
WSAEADDRNOTAVAIL	(WSABASEERR+49)
WSAENETDOWN	(WSABASEERR+50)
WSAENETUNREACH	(WSABASEERR+51)
WSAENETRESET	(WSABASEERR+52)
WSAECONNABORTED	(WSABASEERR+53)
WSAECONNRESET	(WSABASEERR+54)
WSAENOBUFS	(WSABASEERR+55)
WSAEISCONN	(WSABASEERR+56)
WSAENOTCONN	(WSABASEERR+57)
WSAESHUTDOWN	(WSABASEERR+58)
WSAETOOMANYREFS	(WSABASEERR+59)
WSAETIMEDOUT	(WSABASEERR+60)
WSAECONNREFUSED	(WSABASEERR+61)
WSAELOOP	(WSABASEERR+62)
WSAENAMETOOLONG	(WSABASEERR+63)
WSAEHOSTDOWN	(WSABASEERR+64)

WSAEHOSTUNREACH	(WSABASEERR+65)
WSAENOTEMPTY	(WSABASEERR+66)
WSAEPROCLIM	(WSABASEERR+67)
WSAEUSERS	(WSABASEERR+68)
WSAEDQUOT	(WSABASEERR+69)
WSAESTALE	(WSABASEERR+70)
WSAEREMOTE	(WSABASEERR+71)
WSASYSNOTREADY	(WSABASEERR+91)
WSAVERNOTSUPPORTED	(WSABASEERR+92)
WSANOTINITIALISED	(WSABASEERR+93)
WSAEDISCON	(WSABASEERR+101)
WSAHOST_NOT_FOUND	(WSABASEERR+1001)
WSATRY_AGAIN	(WSABASEERR+1002)
WSANO_RECOVERY	(WSABASEERR+1003)
WSANO_DATA	(WSABASEERR+1004)

## NetBios 扩展错误码：参考

代码	含义
0x00	good return, also returned when ASYNCH request accepted
0x01	illegal buffer length
0x03	illegal command
0x05	command time out
0x06	message incomplete, issue another command
0x07	illegal buffer address
0x08	session number out of range
0x09	No resource available
0x0a	session closed
0x0b	command cancelled
0x0d	duplicate name
0x0e	name table full
0x0f	No deletions, name has active sessions
0x11	local session table full
0x12	remote session table full
0x13	illegal name number
0x14	no call name
0x15	cannot put * in NCB_NAME
0x16	name in use on remote adapter
0x17	name deleted
0x18	session ended abnormally
0x19	name conflict detected
0x21	interface busy, IRET before retrying
0x22	too many commands outstanding, retry later
0x23	ncb_lana_num field invalid
0x24	command completed while cancel occurring

0x26	command not valid to cancel
0x30	name defined by another local process
0x34	environment undefined. RESET required
0x35	required OS resources exhausted
0x36	max number of applications exceeded
0x37	no saps available for netbios
0x38	requested resources are not available
0x39	invalid ncb address or length > segment
0x3B	invalid NCB DDID
0x3C	lock of user area failed
0x3f	NETBIOS not loaded
0x40	system error
0xff	asynchronous command is not yet finished

## 七 快速测试网络锁的功能

(1) 备齐所有的组件和决定使用的网络协议。组件可以在开发包里找到，包括光盘、说明书、网络锁。协议可以选择 TCP/UDP, IPX, NetBios, 以下以 TCP/UDP 为例。

(2) 检查网络是否连通，包括相应协议是否已经安装和物理网络是否连通。TCP/UDP 协议可以使用 Ping 工具测试。

(3) 根据上面的生产流程生产一把成品网络锁，允许模块 0 最大用户数为 3。

(4) 在服务端安装 ROCKEY6 的驱动程序，并插入成品锁。驱动程序可以在光盘上找到。

(5) 在服务端启动 NetROCKEY6 SMART 服务程序。运行之前可以修改服务程序目录下的 svrCfg6.ini 改变运行参数，或不修改采取默认设置。默认设置会启动 UDP/TCP 服务。服务程序启动后可以用 NetRockey6 SMART 监视器在任何一台联网机器上察看服务器状态。

(6) 在客户端运行测试程序 NrTest，登录模块 0。由于模块 0 写入 3，NrTest 只允许登录 3 个。用监视器可以查看客户和服务状态。

## 八 常见问题解答

1. 为什么服务端程序或客户端程序不按照 ini 文件中指定的配置运行？

答：这是因为服务端程序或客户端程序找不到 ini 文件的位置而采用默认配置。服务端在 Nr6svr.exe 的当前路径搜索 svrCfg6.ini，客户端可以用函数 SetIniPathName 函数指定。如果不显式指定路径，他们就会在默认的各自当前运行路径搜索，找不到后就会使用默认配置。缺省时客户端在调用 Nr6Cli.dll 的可执行程序当前路径下搜索。用参数或函数指定 ini 路径可以有效的避免搜索不到的问题。

2. 为什么监视器找不到已经启动的服务？

答：监视器要找到网络上的服务需要三点 (1) 监视器运行的机器上要安装至少一个服务程序提供的服务的协议，比如服务器提供 IPX 和 UDP/TCP 服务，监视器的宿主主机上至少要安装 IPX 或 UDP/TCP 其中一个协议。(2) 协议的端口地址必须和服务端相同。监视器运行后会自动在当前路径搜索配置文件 CliCfg6.ini 或 svrcfg6.ini，并读取其中的端口信息。如果找不到这个文件会按默认端口配置。如果默认的或读取得端口数据和服务端不同就会不能通信，这时将服务器的配置文件拷贝到监视器运行目录即可。(3) 检查服务端有没有安装防火墙。如果安装防火墙会屏蔽客户端连接请求，配置或删除掉防火墙即可。

3. 为什么客户端找不到服务器上的锁?

答: 客户端找不到网络锁除了上面 2 中的三点外还要检查服务器上有没有安装驱动和有没有插入网络锁。如果服务器上没有插锁或插入的锁不是网络锁客户端不会得到返回信息。

4. 用户的机器上安装了两套用ROCKEY6网络锁的软件该如何处理?

答: 这种情况虽然少见但也有可能发生这时有两种方法解决(1)将服务器设置在不同的机器上这是最好的解决方法互相没有冲突(2) 如果必须将服务器设置在同一台机器上只需要启动一个服务即可。

5. 网络锁服务器对系统有什么要求?

答: 网络锁的TCP/UDP服务, IPX可以稳定的运行于Win9X, Win2000, WinXP, Win2003系统下。NetBIOS服务可以稳定的运行于Win2000, WinXP, Win2003系统下, 如果可能请尽量使用NT系列的系统做服务平台。