NPort

Programmable Communication Gateway
Programmer's Guide

First Edition, April 2003

Moxa Technologies Co., Ltd.
Tel: +866-2-8919-1230
Fax: +886-2-8919-1231
www.moxa.com
support@moxa.com.tw

Programmer's Guide for NPort PCG

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

Trademarks

MOXA is a registered trademark of
Moxa Technologies Co., Ltd.
All other trademarks or registered marks in this manual belong to their respective manufacturers.

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document "as is," without warranty of any kind, either expressed or implied, including, but not limited to, the particular purpose. Moxa reserves the right to make improvements and/or changes to this manual or the product(s) and/or program(s) described herein at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa Technologies assumes no responsibility for its use, or for any infringements on the rights of fourth parties which may result from its use.

This manual could include unintentional technical or typographical errors. Changes are periodically made to the information herein, with the changes incorporated into new editions of the publication.

MOXA Internet Services

Customer satisfaction is our number one concern. To ensure that customers receive the full benefit of our products, Moxa Internet Services has been set up to provide technical support, driver updates, product information, and user's manual updates.

The following services are provided:

E-mail for technical support

    address: support@moxa.com.tw

World Wide Web site for product information

    address: http://www.moxa.com
                           or
                    http://www.moxa.com.tw

Table of Contents

1
Introduction

NPort Programmable Communication Gateway (PCG) is a stand-alone, programmable communication device with RS-232/422/485 and Ethernet interface for industrial automation. PCG provides a Moxa embedded OS (Moxa OS) and TCP/IP communication stack, and Software Development Kit (SDK) that allows system integrators to build up their own proprietary solution without using traditional PCs.

Moxa OS

NPort PCG has an embedded, small footprint, multi-tasking OS developed by Moxa Technologies Co. Ltd. It was developed originally for TCP/IP-based Terminal Server products starting in 1996. It is a powerful and reliable software platform with user-friendly SDK. Its major features are:

- 16-bit, Unix-like embedded operating system
- Small footprint (<300 KB with TCP/IP protocol stack)
- Preemptive scheduler
- Stream I/O
- Standard BSD Sockets for TCP/IP programming with multi-TCP session support

SDK (Software Development Kit)

To assist in the development of NPort PCG applications, Moxa provides a comprehensive and easy-to-use SDK, designed for use on Windows 95/98/ME/NT/2000/XP platforms. The main features are as follows.

- Borland™ Turbo C 2.01 compiler
- SDK libraries with more than 100 function calls
- SDK Manager and COM2AP utilities for software download and troubleshooting
- PComm Terminal Emulator
- More than 20 example programs
- Documentation
- Single-task user program
- 64 KB of user program space (tiny mode in Turbo C)

Other advanced features include:
- COM2FRM utility for advanced application deployment
- DSCI library for utility development

User Application Space

The user application space available on an NPort PCG is a full 64 KB. This is because the serial I/O, Ethernet, and TCP/IP protocol stack buffers occupy part of the kernel space. In general, this amount of space is sufficient for protocol conversions, and is sufficient for even more complex applications, such as Modbus conversion.

SDK Function Group Overview

The SDK Library contains six groups of function calls. The following table lists the six function call groups along with their primary function.

SDK Library Function Group Overview	
Function Call Group	**Description**
Serial I/O API	Serial communication function calls that follow the same style as PComm Library. This API includes communication parameters, character read/write, block read/write, I/O control, break generation, and more.
BSD Socket API	Standard BSD Socket API for TCP/IP programming. Supports TCP and UDP communication.
Simplified Socket API	Simplified TCP/IP programming. Supports TCP and UDP communication.
System Control API	Overall system control for NPort PCG including Watchdog timer enable/disable, system clock settings, system configuration, system restart, interrupt handling, and serial interface selection.
Flash Access API	Read, write, erase user storage space on flash. There are 32 KB for NPort 4511.
Debug API	For sending messages to the debug window in SDK Manager.

2
Installing SDK

This chapter covers:

➢ How to install SDK utilities and Library
➢ How to install the Turbo C compiler and environmental variables
➢ The location of Libraries and Example Files

System Requirements

You will need to have the following items available to use SDK's development tools:

➢ Windows 9x, NT, ME, 2000, or XP operating system
➢ At least 64 MB of RAM
➢ At least 15 MB of hard disk space

Installing SDK

To install Moxa SDK on a Windows system, insert the software CD-ROM included with the product into your computer's CD-ROM drive. The installation program should start automatically. Simply follow the onscreen instructions to complete the installation.

The installation procedure installs the following files:

Directory	Sub Directory\Files	Description
…\SDK	\Document\	• NPort 4511 hardware installation guide • Programmer's guide • API Reference
	\Example\	SDK example files
	\Firmware\np4511.rom	Kernel firmware for NPort 4511
	\Include\	This directory contains all declaration files
	sdkdbg.h	Debug symbol API declarations
	sdkflash.h	Flash access API declaration
	sdknet.h	Simplified Sockets API declaration
	sdksio.h	Serial I/O API declaration
	sdksock.h	BSD socket API declaration
	Sdksys.h	System API declaration
	\Library\	This directory contains the SDK Library files
	C0sdk.obj	Object code for all APIs.
	Moxa_sdk.lib	API library for NPort 4511 SDK.
	\Utility\	This directory contains SDK Manager, COM2AP, COM2FRM, MoxaTerm utilities along with correspond help file.
	\Dscilib\Library\	This directory contains library for utility development on Windows
	Dsci.dll	DLL file for DSCI (Device Server Control Interface) API
	Dsci.h	Declaration file for DSCI API
	Dsci.lib	Library file for DSCI API
	\Dscilib\Example\	Example program for DSCI
…\TC	tcc.exe	Turbo C 2.01 compiler
	tlink.exe	Turbo C 2.01 linker
	\Include\	Include file for Turbo C.
	\Lib\	Library file for Turbo C

Desktop Icons

The SDK installation places three program icons on your computer's desktop that serve as shortcuts to the following programs: NPort SDK Manager, COM2AP, and Terminal Emulator.

Setting Up Environment Variables

During the installation procedure, the SDK setup program automatically sets up your environment variables for the Turbo C compiler. You may also set up the environment variables manually from **Control panel → System**, or you can add a command line to autoexec.bat.

Assuming the installed directory for SDK is C:\PCGSDK, the command lines are as follows:

```
path=c:\pcgsdk\tc;%path%
set INCLUDE=c:\pcgsdk\tc\include
set LIB=c:\pcgsdk\tc\lib
```

Developing User Applications

In this chapter, we cover the following topics:

➢ Application Development Flow
➢ Software tools required for application development
➢ Software key protection
➢ Application deployment

In order to give you a better understanding of the development flow, we'll use the example program sdkser23.c to illustrate.

Application Development Flow

The following flowchart illustrates the standard process for developing your NPort PCG applications.

```
                        ┌──────────┐
                        │  PRG.C   │
                        └────┬─────┘
                             │ Compile
                             │ tcc.exe
   ┌──────────────────┐  ┌───▼──────┐  ┌──────────────┐
   │ Other OBJ from TC│  │ PRG..OBJ │  │ C0SDK.OBJ    │
   │(e.g., math,      │  │          │  │ MOXA_SDK.LIB │
   │ string, …)       │  └──────────┘  └──────────────┘
   └──────────────────┘
                             │ Link
                             │ tlink.exe    ┌─────────┐
                                            │ Private │
                        ┌──────────┐        │ Key     │
                        │ PRG.COM  │        └─────────┘
                        └──────────┘
         S/W protection
         COM2AP.EXE
                        ┌──────────┐
                        │ PRG.AP   │
                        └──────────┘
                             │ SDK Manager
                        ┌──────────────┐
                        │Download & Run│
                        │ PRG.AP       │
                        └──────────────┘
```
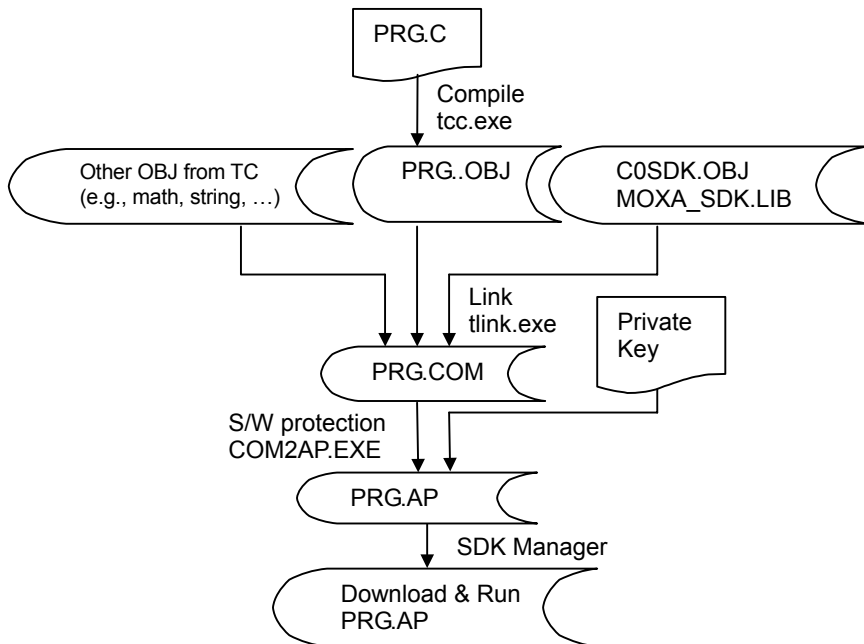
Development System

Before getting started on application development, you need to make sure that all of the required equipment is set up and ready to go. The basic configuration needed for application development is as follows:

➢ Windows95/98/ME/NT/2000/XP computer with SDK installed
➢ PCG with power adapter
➢ Ethernet hub/switch (or use a cross-over Ethernet cable to link the computer and PCG directly)
➢ Two Ethernet cables
➢ One RS-232 cable (DB9 Male to Female straight-through cable). This allows you to transmit and receive serial data along with RTS, CTS, DTR, DSR control signals. However, you will not be able to reach the DCD signal.

Another option is to use a mini adapter and DB9 Female to Female null modem cable. In this way, the DSR and DCD signals can be monitored because they are wired together. The mini adapter is included as a standard accessory in NPort 4511's Starter's Kit.

The basic wiring for establishing a development system is as follows.
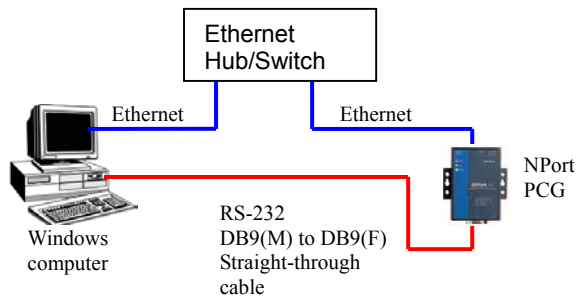
Source Code Editing

Since the Turbo C compiler and linker work under the traditional DOS environment, we recommend using DOS's **edit** program to edit your source code (simply type "edit" on the DOS command line and then hit Enter to start the editor). This is the approach we use in this manual.

You may also use "notepad" or any other text editor to write and edit source code. Turbo C's IDE environment is another good option. However, the environment variables need to be manually configured since NPort SDK works mainly from command line mode.

For example, to edit the file sdkser23.c, first open a DOS window, use the **cd** (change directory) command to open the folder c:\pcgsdk\sdk\example\serial, and then type **edit sdkser23.c**. You should now be able to view and edit the file sdkser23.c.

Compiling and Linking your Application

Under NPort PCG's SDK, compiling and linking is done from DOS command mode. Because of this, you will need to create a make file before compiling and linking a program.

Creating a Make File

For tcc.exe, the following parameters should be enabled or properly assigned.

        -I   – assign include files directory.
        -L   – assign library files directory.
        -O   – optimize jumps.
        -Z   – maximum register usage.
        -1   – 80186/286 instructions.
        -mt – tiny mode
        -c   – compile only
        -w   – enable all warnings

For tlink.exe, the following parameters should be enabled or properly assigned.

        -t   – create COM file.
        -s   – detailed map of segment.

For instance, the make file for C:\PCGSDK\SDK\EXAMPLE\SERIAL\SDKser23.c is as follows.

```
c:\pcgsdk\sdk\example\serial> type sdkser23.bat
tcc -I..\..\include -L..\..\library -O -Z -1 -mt -c -w SDKser23.c
tlink
/t/s  ..\..\library\C0sdk+SDKser23,SDKser23,SDKser23,..\..\library\m
oxa_sdk
```

Generating a COM file

You should be able to generate a COM file after having compiled and linked a program without receiving any error messages. For example, after you run the make file, sdkser23.bat, in c:\pcgsdk\sdk\example\serial, the file sdkser23.com is generated in the same directory.

Using COM2AP

After successfully generating a COM file, you will need to use COM2AP.EXE to convert the COM file into an AP file that has an embedded "Private Key." This must be done before downloading the AP file to the PCG..

Intellectual Property Protection

System integrators often provide their proprietary knowledge to solve particular problems for their customers. Once you start providing NPort PCG as a platform for other users, you will need to manage and protect your intellectual property that is embedded in NPort PCG. For this reason, NPort PCG provides a private key that you can embed into your application (AP). The target NPort PCG should have the same private key with the help of SDK Manager. The Moxa OS automatically checks to make sure that the private key in the AP and kernel match.

Embedding a Private Key in your Application

To embed a private key in the target AP (e.g., sdkser23.com) first double click on Com2ap desktop icon.
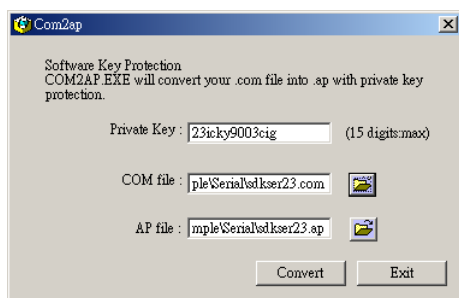
Next, take the following steps to complete the process of embedding a private key:

1. Type the private key in the "Private Key" field. E.g., you could choose "1234" as the private key.
2. Click on the folder icon to the right of **COM file** and then navigate to the COM file you would like to convert (e.g., sdkser23.com).
3. The name of the target AP file (sdkser23.ap, for this example) will appear in the **Protected User Application** text input box. If desired, you can change the file name, or click on the folder icon and navigate to a different folder in which you would like to store the resulting AP file.
4. Prior to downloading and running the AP file, use SDK Manager to set up the same private key for the PCG. Refer to next section for more detail.

Embedding a Private Key from the Command Line

To give program developers a convenient alternative for producing an AP, COM2AP can also be activated from the DOS command line. Enter the command in the following format:

```
COM2AP –Kxxxxxxx –Syyyyyyyy.yyy –Dzzzzzzzz
```

Argument description

-K   Private key. Select a key with at most 15 characters or numbers (e.g., you could choose private key = 23icky9003cig).

-S   Source file. The source COM file.

-D   AP file. Note that you should not type the file extension name since it is predefined as "AP".

For example, to generate the AP file from the command line, add the following line after tlink in the batch file.

```
com2ap –K23icky9003cig –Ssdkser23.com –Dsdkser23
```

Using SDK Manager

After the application has been successfully compiled and linked, it time to use SDK Manager. SDK Manager is a utility that provides the following functions.

 ➢ Search and locate NPort PCGs.
 ➢ Change the IP, Netmask, Gateway, default serial comm parameters, and operation modes for NPort PCG.
 ➢ AP download
 ➢ Debug window

Searching for NPort PCGs

The Search Menu provides two different methods to search the network for a PCG. **Broadcast Search** is used to locate all PCGs connected to the same LAN as the host, and **Search by IP** is used to locate a specific PCG, particularly if it is located outside the LAN and can only be accessed by going through a router. The Search Menu also provides the **Locate Server** function that can be used to identify a particular PCG unit.

Broadcast Search

To access this function, select **Broadcast Search** under the **Search** menu, or click on the **Broadcast Search** toolbar icon.
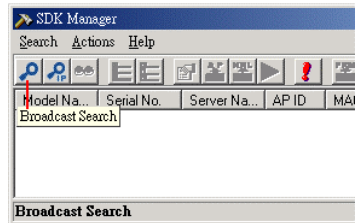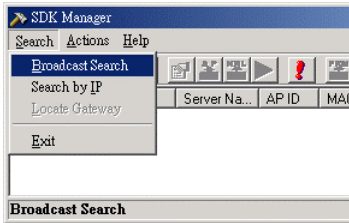
The **Broadcast** window will display the progress of the search.
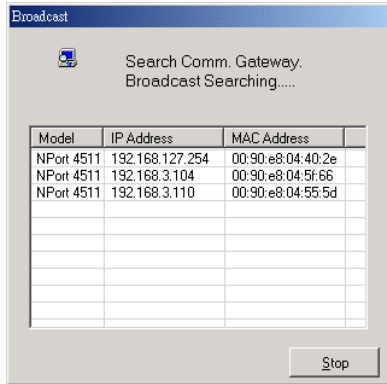
Once the search is complete, the **Model Name**, **Serial No.**, **Server Name**, **AP ID**, **MAC Address**, **IP Address**, **Subnet Mask**, and **OP Mode** of each PCG that was located will be displayed in the SDK Manager window.
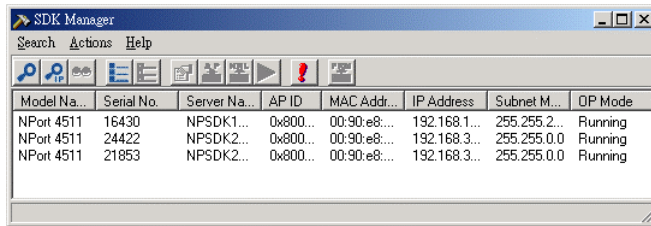

Search by IP

To access this function, select **Search by IP** under the **Search** menu, or click on the **Search by IP** toolbar icon.
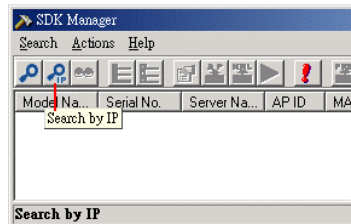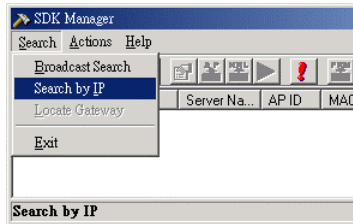
When the **Search by IP Address** window opens, type the IP address in the text

input box, and then click **OK**.
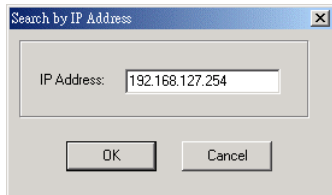
The **Search by IP** window will display the progress of the search.
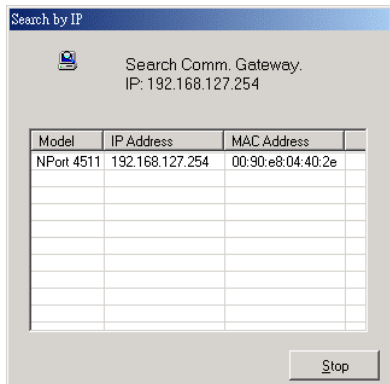
Device not found error message.

NOTE *If you receive the **Specified device not found** error message, as shown above, and the PCG is located on the same LAN as the host, try using Broadcast Search, or change the host computer's IP address and/or Netmask so that the computer and PCG are on the same subnet.*

If the search is successful, the **Model Name**, **Serial No.**, **Server Name**, **AP ID**, **MAC Address**, **IP Address**, **Subnet Mask**, and **OP Mode** of the PCG will be displayed in the SDK Manager window.
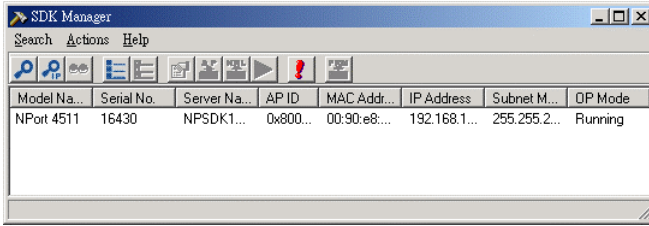
Locate Gateway

The **Locate Gateway** function is used to find the physical location of a PCG unit when there are multiple PCGs on the same network. To use this function, first click on the device you would like to locate to highlight the device's information, and then select **Locate Gateway** under the **Search** menu, or click on the **Locate Gateway** toolbar icon.
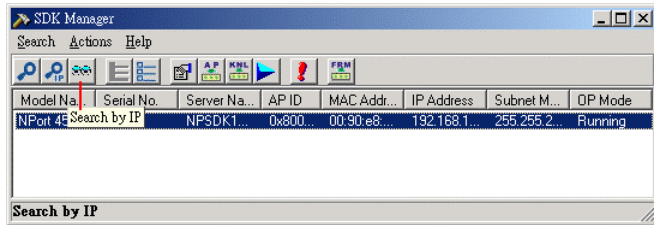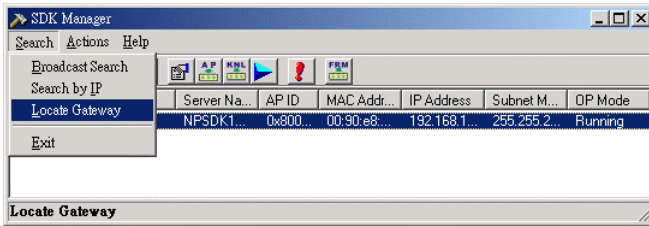
If the Locate Result is OK, then the Ready light on the located PCG unit will blink steadily, allowing you to identify the PCG and IP Address.
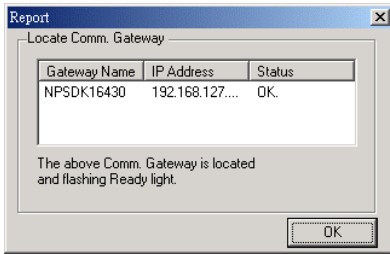
Click on **OK** to cause the PCG unit's Ready light to stop blinking.

Selecting/Deselecting PCGs

The **Select All** and **Deselect All** functions are provided to make it easier to configure large numbers of PCGs.

NOTE *SDK Manager includes a **multi-selection** capability. Hold down the Ctrl key to select multiple PCGs that are not listed in order, or hold down the Shift key to select all PCGs listed between the first and last PCG that you click on.*

Select All

The **Select All** function is used to select all PCGs listed in the SDK Manager window. To use this function, select **Select All** under the **Actions** menu, or click on the **Select All** toolbar icon.
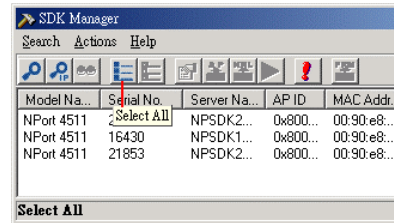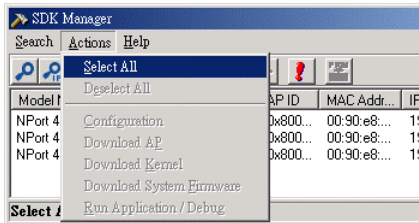
This will cause all NPort 4511s listed in NPort SDK Manager window to become highlighted.
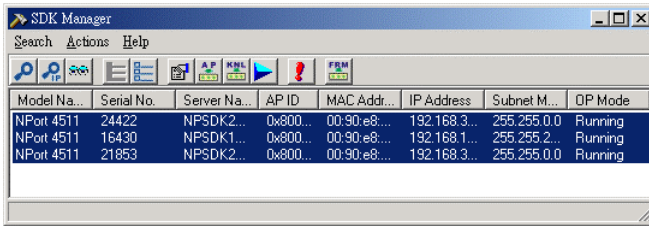
Deselect All

The **Deselect All** function is used to deselect all PCGs listed in the SDK Manager window. To use this function, select **Deselect All** under the **Actions** menu, or click on the **Deselect All** toolbar icon.
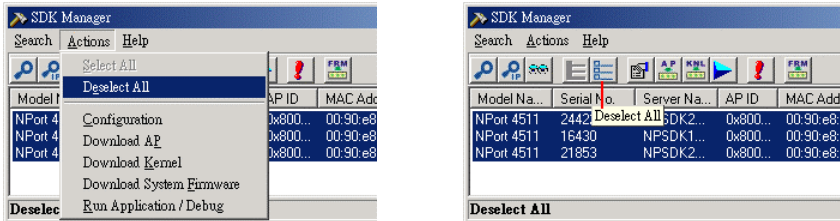
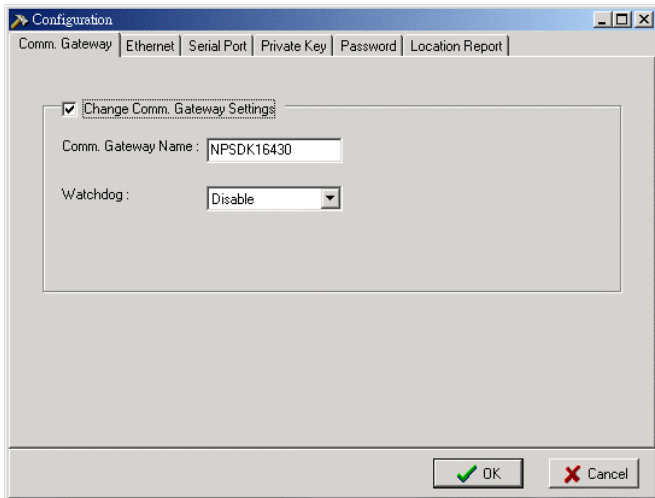This will cause all NPort 4511s listed in the SDK Manager window to become unhighlighted.
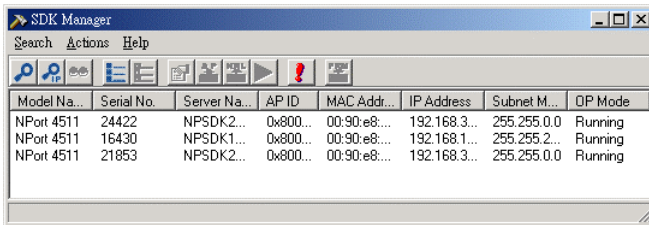
Configuring an NPort PCG

To use this function, select **Configuration** under the **Actions** menu, or click on the **Configuration** toolbar icon. The **Configuration** window opens with the **Comm. Gateway** tab selected. Each of the six Configuration window tabs is discussed in the following subsections.

Comm. Gateway Tab
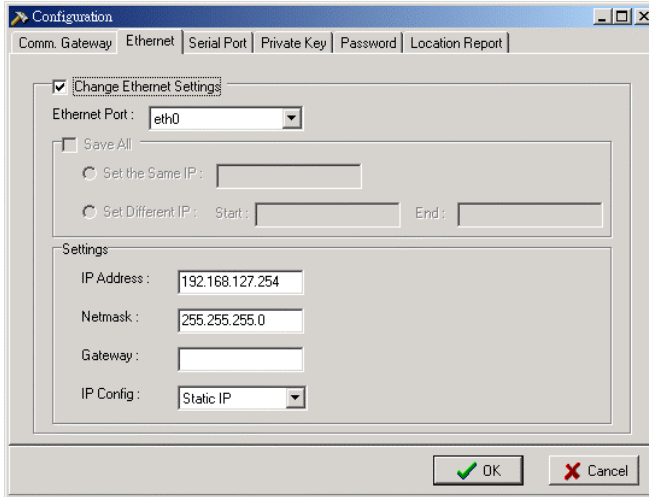
To make changes, first click in the **Change Comm. Gateway Settings** box, and then modify **Comm. Gateway Name** and select **Disable** or **Enable** for the **Watchdog** option.

Server Settings		
Setting	**Options**	**Comments**
Comm. Gateway Name	*Alphanumeric*	Determined by user.
Watchdog	Disable	Disable internal watchdog timer. We suggest that you disable the watchdog timer during the application development stage for easier troubleshooting.
	Enable	Enable internal watchdog timer. The NPort PCG uses the CPU's internal watchdog timer. The kernel is responsible for resetting the timer. You may also set the watchdog timer on or off for applications. During regular operation, we recommend that you turn the watchdog timer on for maximum system reliability. By default, the watchdog timer is set to 1.34 sec.
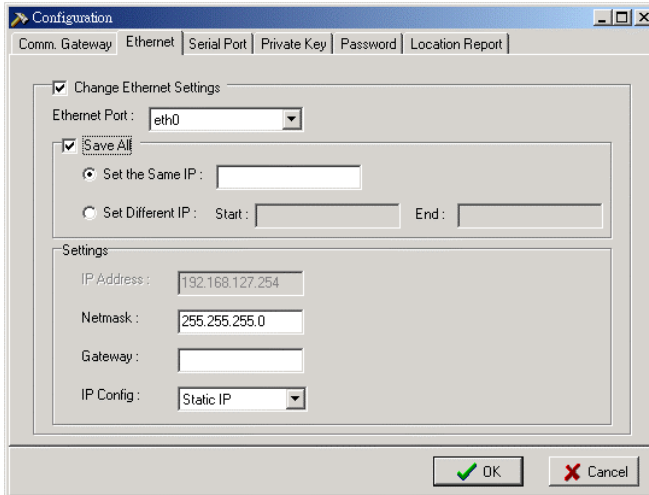
Ethernet Tab

To make changes, first click in the **Change Ethernet Settings** box, and then select which Ethernet Port to modify from the **Ethernet Port** drop-down list.

← When only 1 PCG is selected.

← When multiple PCGs are selected.

Ethernet Settings		
Setting	**Options**	**Comments**
Ethernet Port	eth0	NPort 4511 has only one Ethernet port.
Set the Same IP	*xxx.xxx.xxx.xxx*	*Available with multi-selection* When multiple target PCGs are selected, they will all be configured with the same IP address.
Set Different IPs	*xxx.xxx.xxx.xxx*	*Available with multi-selection* Define a range of IPs by inputting Start and End values. These IP addresses will be assigned in sequence to selected PCGs, in order of their appearance in the Manager window.
IP Address	*xxx.xxx.xxx.xxx*	User defined IP address
Netmask	*xxx.xxx.xxx.xxx*	User defined Netmask—defines which subnet the port is on.
Gateway	*xxx.xxx.xxx.xxx*	User defined Gateway—defines the LAN side IP address of the router.
IP Config	Static IP, DHCP, BOOTP, DHCP+BOOTP	User defined IP configuration method.

Serial Port Tab

This section of the Configuration window allow you to modify the serial communication parameters. The modified parameters will be saved in the PCG's flash ROM.
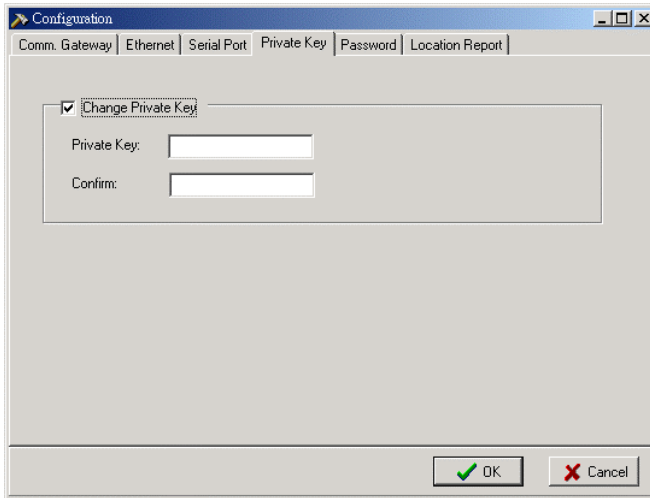
To make changes, first click in the **Change Serial Settings** box, and then click on the port in the display area in the bottom part of the window to make changes to the configuration of the serial ports.
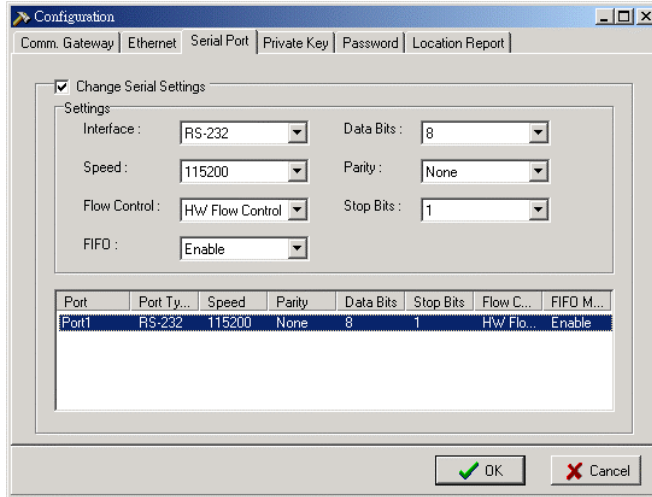
Serial Port Settings	
Setting	**Options**
Port Type	RS-232, RS-422, RS-485(2w)
Speed	50 to 230400 bps
Flow Control	None, HW Flow Control, SW Flow Control
FIFO Mode	Enable, Disable
Data Bits	5, 6, 7, 8
Parity	None, Even, Odd, Space, Mark
Stop Bits	1, 1.5, 2

Private Key Tab

As mentioned above, a private key is required for both the AP and the PCG. To embed the private key in the PCG, first click in the **Change Private Key** box, and then enter the Private Key in the **Private Key** input box, and then reenter the Private Key in the **Confirm** box.
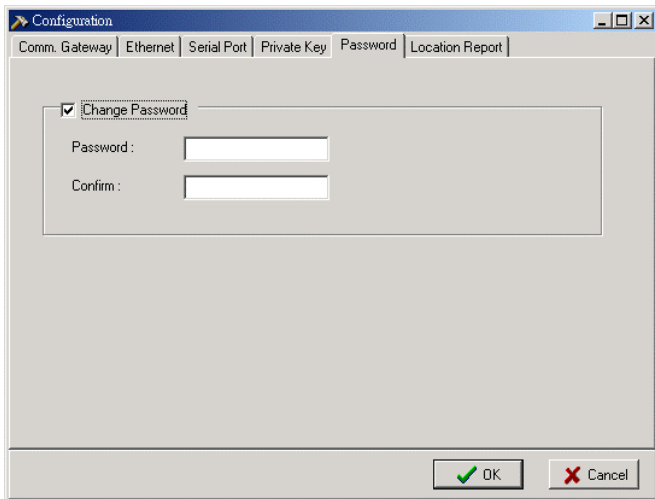

Private Key Settings		
Setting	**Options**	**Comments**
Private Key Input	*Alphanumeric*	Maximum number of alphanumeric characters allowed is 15.
Confirm Private Key	*Alphanumeric*	Retype the Private Key entered in the first box.

Password Tab

The PCG's password is saved in the PCG's flash ROM. The main purpose of the password is to prevent unauthorized access from a host computer. To make changes, first click in the **Change Password** box, and then enter the password in the **Password** input box, and then reenter the password in the **Confirm** box. To erase the password, just leave this field blank and then click **OK**.
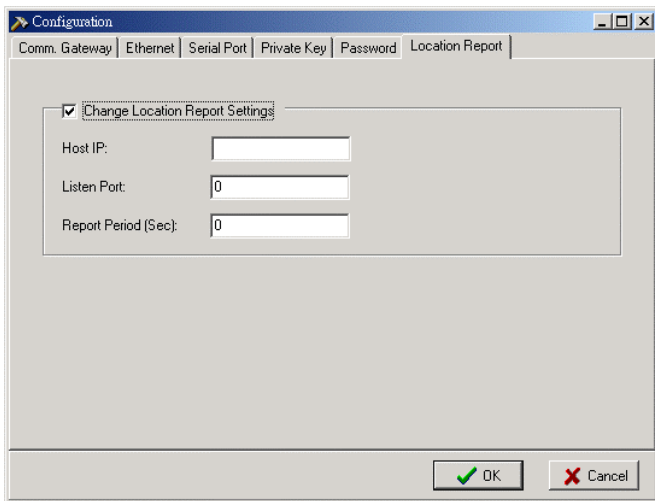

Password Settings		
Setting	**Options**	**Comments**
Password	*Alphanumeric*	Maximum number of characters allowed is 16.
Confirm	*Alphanumeric*	Retype the Password entered in the first box.

Location Report Tab

In a dynamic IP environment, the PCG's IP address changes from time to time, making it hard for the host computer to locate the PCG. To solve this problem, NPort PCG automatically reports it's location to a remote host computer. The Location Report section of the Configuration window allows you to set up the IP address corresponding to the TCP port for a host computer. The **Report Period (Sec)** setting determines how frequently the location report is issued. You can use the DSCI library to develop software that learns the location of a remote PCG. Refer to the DSCI Library help file, **..\PCGSDK\SDK\Dscilib\Library\svrctl.chm**, for more details.

To make changes, first click in the **Change Location Report Settings** box, and then enter **Host IP**, **Listen Port**, and **Report Period (Sec)**.

Location Report Settings		
Setting	**Options**	**Comments**
Host IP	*xxx.xxx.xxx.xxx*	IP of a remote host computer. Leave this field blank to disable this function.
Listen Port	0 to 1024	Host computer's TCP listen port.
Report Period (Sec)	0 to 60 sec	Set to 0 to disable this function.

Application Download

Up to this point, we have only discussed how to configure the PCG's network and other parameters. The next thing to do is download the prepared application (AP) to the PCG for testing.

Download AP

To download the AP, first select **Download AP** from the **Actions** menu, or click on the Download AP toolbar icon.
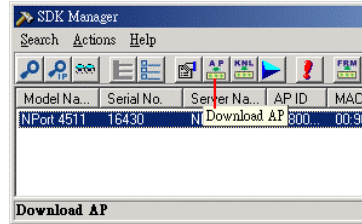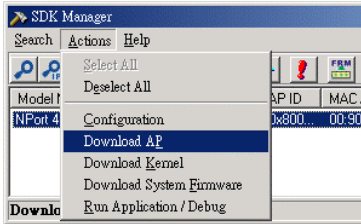
Use the **Open** button to navigate to the folder that contains the AP file, or just type the AP filename directly in the Filename input box. Click **OK** to start downloading the AP. The following window appears when the AP has finished downloading.
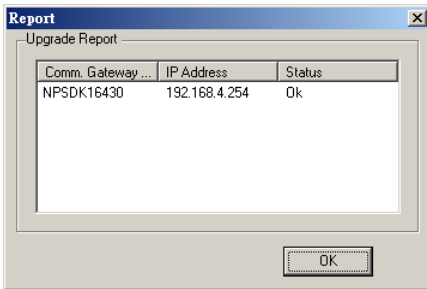
Download Kernel

The kernel for NPort PCG is preloaded before shipment. When you receive an updated kernel from Moxa, you can use the Download Kernel function to update the kernel yourself. To download the kernel to the PCG, first select **Download Kernel** from the **Actions menu**, or click on the **Download Kernel** toolbar icon.

Use the **Open** button to navigate to the folder that contains the kernel file, or just type the kernel filename directly in the Filename input box. Click **OK** to start downloading the kernel. The following window appears when kernel has finished downloading.

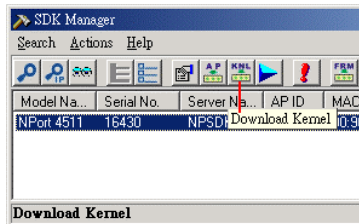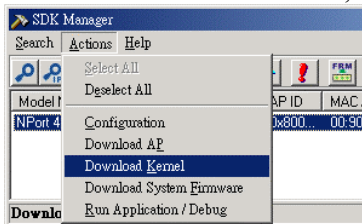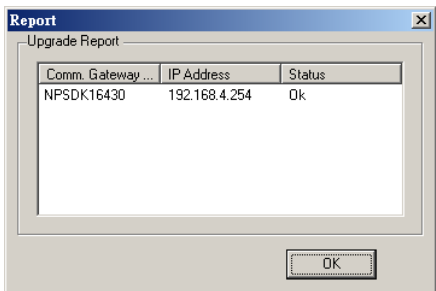# Run Application / Debug

You should now be ready to run your application, but first we need to discuss the two PCG operation modes.

Choose from PCG's Running Mode and Developing Mode

NPort 4511 provides two operation modes selected by DIP switch. Turn **SW1** to the **ON** position for Developing Mode, and to the **OFF** position for Running Mode.

Developing Mode

When set for **Developing Mode**, the PCG will not start running the application automatically after the system is rebooted. To start the application, select **Run Application / Debug** from the **Actions** menu, or click on the **Run Application / Debug** toolbar icon. We suggest setting the operation mode to Developing Mode when going through the debugging process.

Running Mode

When set for **Running Mode**, the PCG automatically executes the application after the system boots up. However, you can stop the application from within SDK Manager. This mode is suitable for regular shipment.

Start Application

To start an application loaded in the PCG, first select **Run Application / Debug** from the **Actions** menu, or click on the **Run Application / Debug** toolbar icon.
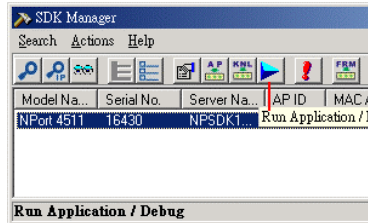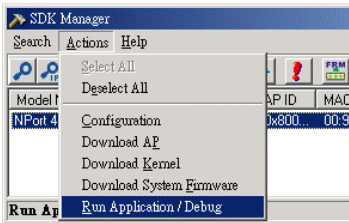
The Debugger window will appear as shown in the following figure. From the SDK Debugger window, click on the Start Application button to start running your PCG application. There are three message areas in the Debugger window as shown below.
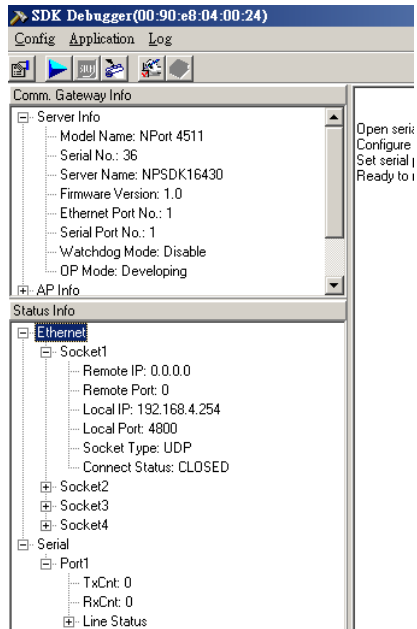
Comm. Gateway Info Area

The PCG's basic configuration settings are shown in the **Comm. Gateway Info.** area. The information includes the following:

- **Server Info.**—includes Model Name, Serial No., Server Name, Firmware Version, number of Serial Ports, number of Ethernet Ports, Watchdog Mode, and OP Mode
- **AP Info.**—includes latest AP Name and file location, Version, Date, and Size
- **Ethernet Info.**—includes MAC address, IP address, Netmask, and Gateway
- **Serial Info.**—includes serial communication parameters.

Comm. Gateway Info
- Server Info
  - Model Name: NPort 4511
  - Serial No.: 36
  - Server Name: NPSDK16430
  - Firmware Version: 1.0
  - Ethernet Port No.: 1
  - Serial Port No.: 1
  - Watchdog Mode: Disable
  - OP Mode: Developing
- AP Info
  - AP Name: C:\PCGSDK\SDK\Example\Serial\SDKSER23
  - Version: 0.0.0.0
  - Date: 4-1-2003
  - Size: 3870
- Ethernet Info
  - LAN1
    - MAC: 00:90:e8:04:00:24
    - IP: 192.168.4.254
    - Netmask: 255.255.0.0
    - Gateway:
- Serial Info
  - Port1
    - Serial Port Type: RS-232
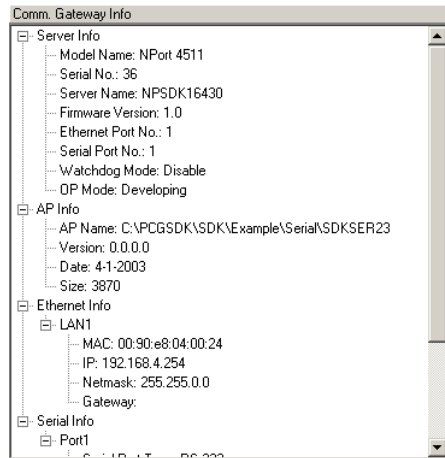
Status Info Area

The **Status Info** area shows PCG Gateway's online operation status. There are two main items.

- **Ethernet**—NPort PCGs support up to ten user programmable TCP sessions. To monitor the usage of these TCP sessions from SDK Debugger, you can see Remote IP, Remote Port, Local IP, Local Port, Socket Type, and Connect Status.
- **Serial**—you can see the total Tx and Rx counts starting from when the SDK Debugger was activated. In addition, you can see line status, including RTS, DTR, CTS, DSR, and DCD.

Status Info
- Ethernet
  - Socket1
    - Remote IP: 0.0.0.0
    - Remote Port: 0
    - Local IP: 192.168.4.254
    - Local Port: 4800
    - Socket Type: UDP
    - Connect Status: CLOSED
  - Socket2
  - Socket3
  - Socket4
- Serial
  - Port1
    - TxCnt: 0
    - RxCnt: 0
    - Line Status
      - RTS: Off
      - DTR: On
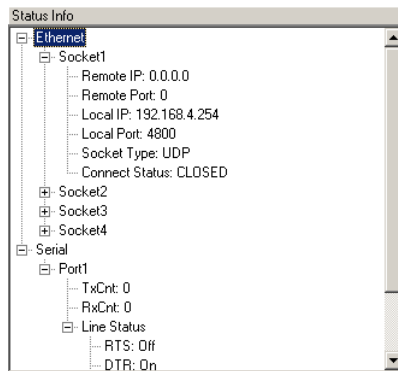
Debug

The main debug approach for NPort PCG is to put the debug API in your source code as a debug symbol. The debug message will be sent to SDK's debug window via the Ethernet console. The "single step" and "break point" debug methods are currently not supported.
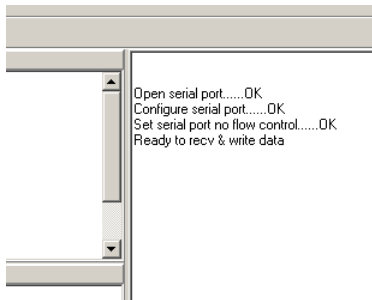
Example Programs

NPort PCG's SDK is a comprehensive utility designed to match a programmer's basic intution. To let you quickly build up your first application, we've prepared an abundance of examples. All of the example programs are in the folder ..\SDK\EXAMPLE.
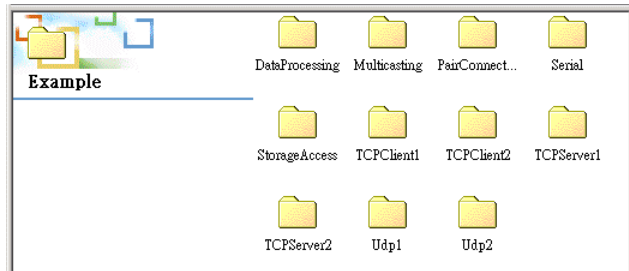
4
Application Deployment

After your application has been tested and is ready to go, it's time to deliver the NPort PCG with the software installed to your customer. There are several useful tips and tools that let you organize the production efficiently.

Changing Default Settings for Multiple PCGs

The default settings for NPort PCG may be different from the settings you want to use when shipping the product to your customers. To make the job easier, SDK Manager can be used to change multiple PCG settings, all at the same time. The main procedure is as follows:

1. First, connect all PCGs and the host computer to the same Ethernet hub or switch.
2. Start SDK Manager, and use Broadcast Search to search all PCGs. All of the PCGs connected to the hub or switch should appear in the SDK Manager window.
3. Enter configuration menu, and change each field under each tab to the desired settings.

Creating System Firmware—COM2FRM

In general, the AP file is downloaded to the PCG separate from the kernel. However, NPort PCG's SDK provides you with another advanced tool, COM2FRM, that is used to merge the kernel and AP file into a system firmware file. In this way, product delivery is made easier, and furthermore, the quality of the software is assured.

The COM2FRM program can be found in the .\SDK\UTILITY\COM2FRM directory. For easy access, simply create a shortcut on your desktop to the COM2FRM program. After starting the COM2FRM program, the following window will appear.

To generate a system firmware file, you will need to input the Private Key, original COM file, and Kernel file. The kernel file can be found in .\SDK\FIRMWARE. After typing in the above information, type in a file name for the generated system firmware file. To download the system firmware, open SDK Manager and locate the target PCG. Select **Download System Firmware** under the Actions menu, or click on the **Download System Firmware** toolbar icon.
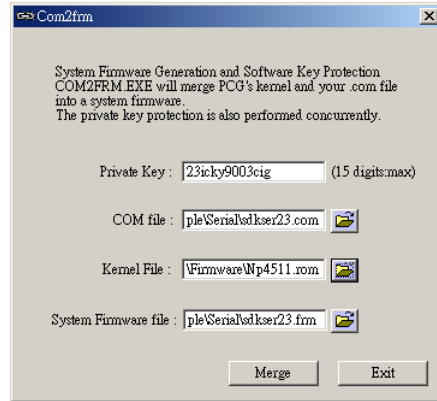
Command Line Usage

To speed up program development speed, COM2FRM can also be activated from the DOS command line, as shown below. Simply type:

```
COM2FRM -Kxxx -Syyyyyyyy.yyy -Fyyyyyyyy.yyy -Dzzzzzzzz
```

Argument description

- -K   Private key. Select a key with at most 15 characters or numbers (e.g., you could choose private key = 23icky9003cig).
- -S   Source file. The source COM file.
- -F   The file name for the kernel.
- -D   FRM file. Note that you should not type the file extension name since it is predefined as "AP".

For example, to generate an FRM file from the command line, add the following line after tlink in the batch file.

```
com2frm -K23icky9003cig -Ssdkser23.com -Fnp4511.rom -Dsdk
ser23
```

Field Utility

SDK Manager is the main field utility for configuration and troubleshooting, but NPort PCG's SDK also provides you the DSCI Library that can be used for developing your own proprietary utility. For more information, please refer to next section.

Developing a PCG Utility

NPort PCG's SDK provides the DSCI Library for customers who want to develop their own utility for use on a Windows computer.

DSCI Library

DSCI (Device Server Configuration Interface) is a set of APIs that run on a Windows 95/98/ME/NT/2000/XP system to search, locate, and config the PCG over the network. The DSCI library can be found in the folder .\SDK\DSCILIB\LIBRARY. For more information, refer to document Svrctl.chm in that directory, and examples located in .\SDK\DSCILIB\EXAMPLE.
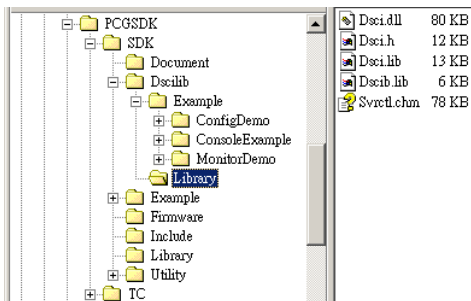
AP ID

NPort PCG has a special parameter, called "AP ID," that is of particular interest to PCG programmers who intend to repackage NPort PCG with their own application, new product name, and new model number. The AP ID can be used to distinguish between different application programs.

You can develop several versions of an application for use with different projects. In this case, it is necessary for the host utility to identify which application is running on the PCG. NPort PCG uses the AP ID to identify which PCG is associated with which DSCI API. The AP ID is stored in the System Parameter Block within the PCG.

To set up the AP ID, you will need to insert the following code at the beginning of your own source code (refer to the SDK Library System Control API).

```
void sys_Set_RegisterID( unsigned long id);
```

The AP ID can be read by both NPort SDK Manager, as well as by your own management utility created with DSCI Library.

5
PCG FAQs

In this chapter, we present a short list of Questions and Answers that you can refer to to solve frequently encountered problems.

Q: Why am I unable to compile a PCG application?
Ans.
1. Check the source code. You may have forgotten to include a header file, or you could have an instruction or syntax error.

2. Check the compiling command arguments. (Refer to the Chapter 3, Compiling and Linking your Application).
    Correct command: **tcc –O –Z –l –mt –c –w ap.c**

3. Check your compiler. PCG is 16-bit platform, and only 16-bit compilers, such as Turbo C, can be used to compile the source code.

Q: Why am I unable to link a PCG application?
Ans.
1. Check to make sure the source code is compiled, and the correct *.obj file was created (refer to the Chapter 3, Compiling and Linking your Application).
2. Check the linking command arguments (refer to Chapter 3, Compiling and Linking your Application).
    Corrrect command: **tlink /t /s c0sdk+ap, ap, ap, moxa_sdk**
3. In addition to using PCG's SDK Lib, some standard Turbo C instructions are also used. Refer to Chapter 4 of the "NPort PCG API Reference."

Q. How can I set a password?
Ans.
1. You can set the password using SDK Manager's Configuration Function. For detailed instructions, see Chapter 3, Using SDK Manager.

Q. What is the password of PCG used for?

Ans.

1. It protects PCG's parameters from being modified unwittingly. There could be more than one PCG connected to the network, and users might modify the configuration of the incorrect PCG.
2. If the PCG's network is connected to a WAN, this protection will prevent someone from intentionally changing the configuration.

Q. How can I protect my AP (Application Program)?

Ans.

1. Moxa provides a "Private Key" for protecting against downloading the wrong program. Refer to Chapter 3, Compiling and Linking your Application, for more information.

Q. Why can't I download a program to the PCG?

Ans.

1. Before downloading the generated file (*.com) to the PCG, it must be transformed into the correct download file type. Refer to the Chapter 3, Compiling and Linking your Application, for more information.
2. Make sure that you download a program that has the correct format. Moxa PCGs support three types software that can be downloaded: "Application Program," "PCG Kernel," and "Firmware."
3. The "Private Key" in the PCG and the private key in the downloaded program might not match. Refer to Chapter 3, Using SDK Manager, for more information.

Q. What is the purpose of the Private Key?

Ans.

1. The "Private Key" is a special function provided for use with Moxa PCGs. It is used to prevent mismatch downloading to PCG when there is more than one PCG connected to the network.
2. If the PCG is set up with a Private Key, only AP files with the same Private Key can be downloaded to the PCG.

Q. What is the difference between "Download AP" and "Download Firmware?"
Ans.
1. "Download AP" is for downloading the <u>A</u>pplication <u>P</u>rogram to the PCG, but "Download Firmware" is downloading a file that combines the <u>A</u>pplication <u>P</u>rogram and "Kernel."
2. The Firmware contains both the PCG's application program and kernel. This is convenient for developers when distributing the final software package to end users.

Q. Is there any method available to create my own utility?
Ans.
1. Moxa provides another library, "DSCI Lib." In fact, DSCI Lib was used by Moxa's software engineers to create the SDK Manager utility. Refer to the %PCGSDK\SDK\DSCILIB folder for a detailed introduction to the DSCI Library.