

TPMC871
PC Card Interface
Windows NT Installation

Version 1.0
Issue 1.0
19 November, 1999

TEWS DATENTECHNIK GmbH
Am Bahnhof 7
D-25469 Halstenbek
Germany
Tel.: +49 (0)4101 4058-0
Fax.: +49 (0)4101 4058-19
E-Mail: Support@tews.com
Web: <http://tews.com>

TPMC871

PC Card Interface

Windows NT Installation

This document contains information, which is proprietary to TEWS DATENTECHNIK GmbH. Any reproduction without written permission is forbidden.

TEWS DATENTECHNIK GmbH has made any effort to ensure that this manual is accurate and complete. However TEWS DATENTECHNIK GmbH reserves the right to change the product described in this document at any time without notice.

This product has been designed to operate with PCI Mezzanine Card (PMC) compatible carriers. Connection to incompatible hardware is likely to cause serious damage.

TEWS DATENTECHNIK GmbH is not liable for any damage arising out of the application or use of the device described herein.

1999 by TEWS DATENTECHNIK GmbH

Issue	Description	Date
1.0	First Issue	19 November, 1999

Table of Contents

1 INTRODUCTION	4
2 INSTALLATION.....	5
3 MICROSOFT KNOWLEDGE BASE ARTICLES.....	6
HOWTO: Add PCMCIA Support to a Windows NT Device Driver	6
SUMMARY.....	6
MORE INFORMATION	6
Add PC Card Information to the PCMCIA Database.....	6
Finding the PC Card Resources	7
Using Registry Overrides for PCMCIA Cards	7

1 Introduction

The TPMC871 is supported by the default Windows NT PCMCIA device driver (*pcmcia.sys*). Because such PCMCIA features as dynamic configuration and hot-plugging are unsupported, cards installed while Windows NT is running, are detected the next time Windows NT is restarted.

There is an other important fact to know, conflicts (e.g. IRQ level) between supported PCMCIA devices and other devices are not detected or resolved; you must resolve these conflicts yourself.

2 Installation

After installation of the TPMC871 and booting Windows NT, the PCMCIA device driver *pcmcia.sys* must be started if not already done.

Perform the following steps to setup necessary drivers:

1. Activate the PCMCIA device driver by double-clicking the **Driver** icon in the **Control Panel** (be sure that you logged in with administrator rights). Select the **Pcmcia** driver and change the start value to "**System**".
2. Install the device driver for the installed PC Card. If this driver is shipped with Windows NT (e.g. Atdisk devices such as Flashdisks) activate this driver. Be sure that the start value of this drivers is set to "**System**".
3. Restart Windows NT

After restart the PCMCIA device driver should detect the TPMC871 and the installed PC Card.

Insure this by double-clicking the **PCMCIA** icon in **Control Panel**. Socket 0 of the TPMC871 should display manufacturer and product name of the installed PC Card. After double-clicking this entry a property windows appears. This windows contains information about the loaded PC Card driver and allocated resources.

If the PC Card is a disk device (e.g. Flashdisk), assign a drive-letter to the drive partition(s) with the disk manager.

If the PC Card is not recognized by the PCMCIA device driver, an entry must be added in the PCMCIA database. How to do this please refer to Microsoft article below.

Other PCMCIA related articles for Windows NT are available on this site:

<http://support.microsoft.com/support/search/c.asp>

3 Microsoft Knowledge Base Articles

HOWTO: Add PCMCIA Support to a Windows NT Device Driver

The information in this article applies to:

- Microsoft Win32 Device Driver Kit (DDK) for Windows NT, versions 3.51, 4.0
-

SUMMARY

PCMCIA support under Windows NT is in the form of basic PC Card enabling. Such PCMCIA features as dynamic configuration and hot-plugging are unsupported. Basically, a developer writes an ISA device driver and ensures that the PC Card is configured correctly (either via the CIS tuple information or the registry overrides described below). The ISA driver checks the registry to see if the card is configured and gets the resources from the resource list created by Pcmcia.sys. The ISA driver is also responsible for claiming the card resources via IoReportResourceUsage or a similar mechanism.

For additional information about how to mask specific IRQs to PCMCIA devices as well as detailed information about the InterruptMask value of the PCMCIA registry key that controls IRQ masking, please see the following articles in the Microsoft Knowledge Base:

[Q168303](#) Masking IRQs for PCMCIA Devices

MORE INFORMATION

For Pcmcia.sys to recognize the card, there must first be an entry in the PCMCIA database. This database is stored in the Windows NT registry. In the following discussion, note that HKLM indicates HKEY_LOCAL_MACHINE and CCS indicates CurrentControlSet.

WARNING: Modifying the registry can cause the system to become unstable and may cause the system to become unusable (requiring a reinstallation). Do not modify the registry unless you understand the implications of any changes you are making.

Note that the Pcmcia.sys driver shipped with Windows NT 4 does not support PCCards with a DMA (direct memory access) interface. Also, there are no registry overrides that can be used to configure these DMA-capable PCCards.

Add PC Card Information to the PCMCIA Database

1. Under HKLM\SYSTEM\CCS\Services\Pcmcia\DataBase add a key for Manufacturer ID. This information is in the Level 1 Version/Product Information Tuple. The Manufacturer ID is the first ASCII string after the Minor Version Number. For example:

```
HKLM\SYSTEM\CCS\Services\Pcmcia\DataBase\XYZ Corp
```

2. Under Manufacturer ID, add a key for the Device ID. The Device ID is the second ASCII string (just after the Manufacturer ID). For Example:

```
HKLM\SYSTEM\CCS\Services\Pcmcia\DataBase\XYZ Corp\XYZ device
```

3. For PCMCIA.SYS to configure the PC Card, you must specify the name of a driver under the Device ID key. Add this as a value under the Device ID key:

Driver : REG_SZ : Xyz

Note that the value of the driver specified here does not need to match the "real" driver's name. For example, the driver that actually controls a card may be `Abc.sys` and have a service key name "Abc", but the driver name specified in the PCMCIA database is "Xyz". In this case, the driver `Abc.sys` would look for the resource information under "Xyz" in the Hardware key (described below).

Finding the PC Card Resources

After adding the PCMCIA database information and rebooting, the `Pcmcia.sys` driver attempts to configure the card according to the CIS tuple data. If the tuple data is successfully parsed and the PC Card configured, PCMCIA creates a resource list under the following key:

```
HKLM\HARDWARE\DESCRIPTION\System\PCMCIA_PCCARDS
```

There should be a value under this key of the name specified in the PCMCIA database entry for this particular card. For the example presented thus far, the PCMCIA database described driver "Xyz". So, under the PCMCIA PCCARDS key should be a value similar to the following:

```
Xyz : REG_FULL_RESOURCE_DESCRIPTOR : ...
```

If the ISA driver for the PC Card finds this value, then the PC Card is configured and ready to go. The resource descriptor layout is described in the Windows NT DDK. The ISA driver should parse this resource list to get the information and map the I/O and memory spaces as required, as well as configure the interrupt.

If the resource list is found in the dynamic HARDWARE key, then the card is configured and usable. If the resource list is not there, then the card is not in the socket, or an override is required to make `Pcmcia.sys` configure the card.

There may be more than one resource list value under this key, so make sure to check that the value name matches the driver name specified in the PCMCIA database.

Using Registry Overrides for PCMCIA Cards

For PCMCIA overrides to be processed, they must specify either an interrupt or I/O port address (or both). If the card does not use either, then you will have to select a dummy I/O port address to force the override to be accepted.

4. Create a new driver key in the Services key. The name for this key should match the key specified in the PCMCIA database for this specific PCCARD, with a single digit ("1" through "9") appended to it. For example:

```
HKLM\SYSTEM\CCS\Services\Xyz1
```

5. Under the new driver key, add a Parameters key. For example:

```
HKLM\SYSTEM\CCS\Services\Xyz1\Parameters
```

6. Under the Parameters key, the following value is required. If it is not present or set to zero, the PCMCIA driver will not use this configuration and will search for another configuration in the registry. `Pcmcia.sys` will stop searching for configurations as soon as it finds a valid override. This is why you cannot use more than one PC Card with a specific Manufacturer ID and Device ID.

```
Pcmcia : DWORD : 1
```

7. Under the Parameters key, add the optional keywords as needed. Examples include:

```
AttributeMemorySize : DWORD : 0x1000  
InterruptNumber : DWORD : 0xf  
IoBaseAddress : DWORD : 0x170  
IoBaseAddress_1 : DWORD : 0x376  
IoLength : DWORD : 0x8  
IoLength_1 : DWORD : 0x2
```

Possible override values under:

HKLM\SYSTEM\CCS\Services\Pcmcia\DataBase\XYZ Corp\XYZ device

Value Name	Data Type	Description
CcrBase	REG_DWORD	Configuration register base
PortWidth16	REG_DWORD	1=16 bit accesses to I/O space
Driver	REG_SZ	Driver name
CardMemorySize	REG_DWORD	[0] Card memory length
CardMemorySize_1	REG_DWORD	[1] Card memory length
AttributeMemorySize	REG_DWORD	[0] Attribute memory length
AttributeMemorySize_1	REG_DWORD	[1] Attribute memory length

Possible override values for:

HKLM\SYSTEM\CCS\Services\Xyz1\Parameters

Value Name	Data Type	Description
Pcmcia	REG_DWORD	1=enable this configuration
InterruptNumber	REG_DWORD	Interrupt value
IoBaseAddress	REG_DWORD	[0] I/O port base address
IoLength	REG_DWORD	[0] I/O port length
IoBaseAddress_1	REG_DWORD	[1] I/O port base address
IoLength_1	REG_DWORD	[1] I/O port length
MemoryMappedBaseAddress	REG_DWORD	[0] Host base memory
MemoryMappedSize	REG_DWORD	[0] Host base memory length
PCCARDMemoryWindowOffset	REG_DWORD	[0] Card base (host base offset)
Address_16	REG_DWORD	1=16-bit memory access
MemoryMappedBaseAddress_1	REG_DWORD	[1] Host base memory
MemoryMappedSize_1	REG_DWORD	[1] Host base memory length
PCCARDMemoryWindowOffset_1	REG_DWORD	[1] Card base (host base offset)
Address1_16	REG_DWORD	1=16-bit memory access
MemoryMappedBaseAddress_2	REG_DWORD	[2] Host base memory
MemoryMappedSize_2	REG_DWORD	[2] Host base memory length
PCCARDMemoryWindowOffset_2	REG_DWORD	[2] Card base (host base offset)
Address2_16	REG_DWORD	1=16-bit memory access
MemoryMappedBaseAddress_3	REG_DWORD	[3] Host base memory
MemoryMappedSize_3	REG_DWORD	[3] Host base memory length
PCCARDMemoryWindowOffset_3	REG_DWORD	[3] Card base (host base offset)
Address3_16	REG_DWORD	1=16-bit memory access
PCCARDAttributeMemoryAddress	REG_DWORD	[0] Attribute memory
PCCARDAttributeMemorySize	REG_DWORD	[0] Attribute memory length
PCCARDAttributeMemoryOffset	REG_DWORD	[0] Attribute memory offset
AttributeMemory_16	REG_DWORD	1=16-bit memory access
PCCARDAttributeMemoryAddress_1	REG_DWORD	[1] Attribute memory
PCCARDAttributeMemorySize_1	REG_DWORD	[1] Attribute memory length
PCCARDAttributeMemoryOffset_1	REG_DWORD	[1] Attribute memory offset
AttributeMemory1_16	REG_DWORD	1=16-bit memory access
ModemFunction	REG_DWORD	1=multifunction modem device
PCCARDReadyInterrupt	REG_DWORD	* Not implemented *