

TCL Device Drivers
for
IBM OS/2 LAN Distance
&
IBM OS/2 COM Devices

User Manual

© COPYRIGHT
IT Factor Ltd t/a TCL

No part of this publication may be reproduced without the written permission of Technology Concepts Ltd.

Software licences are granted for use with one processor and are not transferable

Technology Concepts Ltd. make no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose.

IT Factor Ltd t/a TCL
24 Thatcham House
Turners Drive, Thatcham
Berkshire, RG19 4QD
United Kingdom
Tel: + 44 (0) 1635 876754
Fax: + 44 (0) 1635 871739
Web: www.TheTCL.com
Email: info@teccon.co.uk

NOTICE

The information herein has been carefully checked and is believed to be entirely accurate at time of going to press. However no responsibility is assumed for any inaccuracies or typographical errors. Furthermore, no liability is assumed arising from the use of any product detailed within. TCL reserves the right to make alterations without notice and recognises that the information contained within does not convey to the purchaser any license under the patent rights of IT Factor Ltd t/a TCL.

All trademarks acknowledged.

How to Use this Manual

PCI-CC/B and CC/B Adapters

The PCI-CC/B and CC/B buffered serial I/O adapter cards are only supported as COM devices and are not supported by the TCL LAN Distance Device driver. To install the COM Device Driver for the CC or PCI-CC range of cards please refer to Sections 5 and 6 of this manual.

TCL Intelligent Multiport Adapters

The TCL intelligent Multiport Adapter cards support both the LAN Distance and COM Device Drivers. If you require LAN Distance and COM device support please start by reading Sections 1, 2, 3, 4, and 6.2.1 of this manual. If however you only require COM device support please start by reading Sections 5 and 6 of this manual.

Sections 7 contains a guide for the CC and TCL Intelligent Multiport COM device driver API. This guide lists some of the OS/2 COM device API calls that may be used to write applications that access the COM devices.

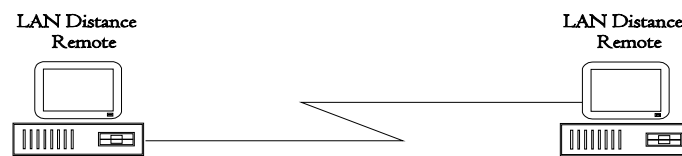
Table of Contents

How to Use this Manual	2
PCI-CC/B and CC/B Adapters	2
TCL Intelligent Multiport Adapters	3
1. Introduction - LAN Distance Device Driver	5
2. Quick LAN Distance Driver Installation	7
2.1 Quick LAN Distance Driver Removal	7
3. LAN Distance Driver Installation	8
3.1 Driver Installation Procedure	8
3.2 About the LAN Distance Installation Program	11
3.3 Verifying the LAN Distance Installation	12
3.3.1 Overview	12
3.4 Adding an additional TCL adapter (for LAN Distance) ..	13
3.5 Removing a TCL Adapter (from LAN Distance)	13
3.6 Removing the TCL LAN Distance Device Driver	14
4. LAN Distance Device Driver Operation	15
4.1 Introduction	15
4.2 Initial Port Settings	16
4.3 Initial RX/TX Buffer Settings	16
5. Introduction COM Device Driver	17
6. Serial COM Driver - Getting Started	18
6.1 CC/4B & CC/8B Direct Card Driver Installation	18
6.2 Intelligent Card Installation	20
6.2.1 Intelligent Card COM Device Driver Removal	21
6.2.2 Verifying COM driver Installation	21
6.2.3 TCL Intelligent COM Device Driver Operation	22
6.2.4 TCL Intelligent COM Device Naming Convention	23

7	Programming a Serial COM Device	24
7.1	What are Device Drivers and Serial Devices?	24
7.2	Programming a Serial Device	25
	7.2.1 Using Assembler to access serial device APIs.	25
	7.2.2 Using 'C' to access serial device API's	25
	7.2.3 Suggested Receive and Transmit Algorithms	26
7.3	Serial Device APIs	27
	7.3.1 @DosOpen	27
	7.3.2 @DosClose	27
	7.3.3 @DosRead	28
	7.3.4 @DosReadAsync	28
	7.3.5 @DosWrite	28
	7.3.6 @DosWriteAsync	29
	7.3.7 @DosDevIOctl	29
	7.3.8 Open/Close API Effects on Driver	46
8	The Driver Configuration File	47
8.1	Introduction	47
8.2	Configuration File Syntax	48
8.3	The Setcard Paragraph	48
8.4	The Setport Paragraph	49
8.5	Example Configuration File	51
	APPENDICES	53
	Appendix 1 Installation Disk Contents	53
	Appendix 2 TCLMODE command	54
	Appendix 3 Troubleshooting and Hints	55
	Appendix 4 Version Specific Restrictions	57

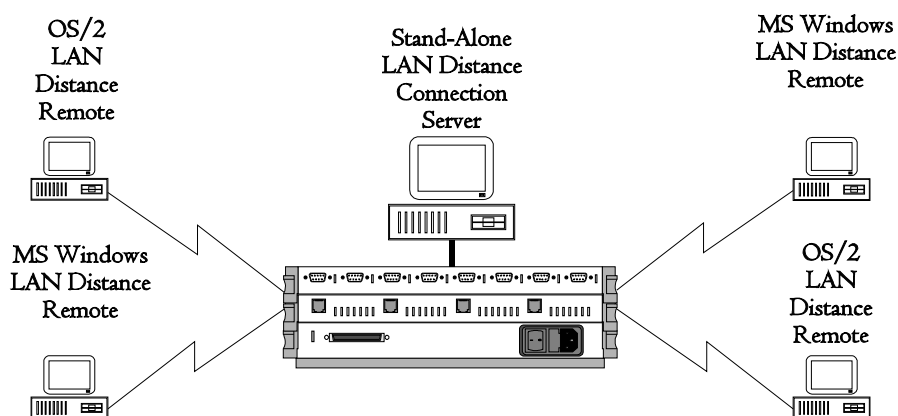
1. Introduction - LAN Distance Device Driver

The TCL LAN Distance device driver will allow any of the TCL intelligent Serial I/O cards to be used with the IBM OS/2 LAN Distance product. TCL supply a range of Intelligent Serial I/O cards, ranging from a two port card with one built in modem, to an eight to sixty-four port expandable system.



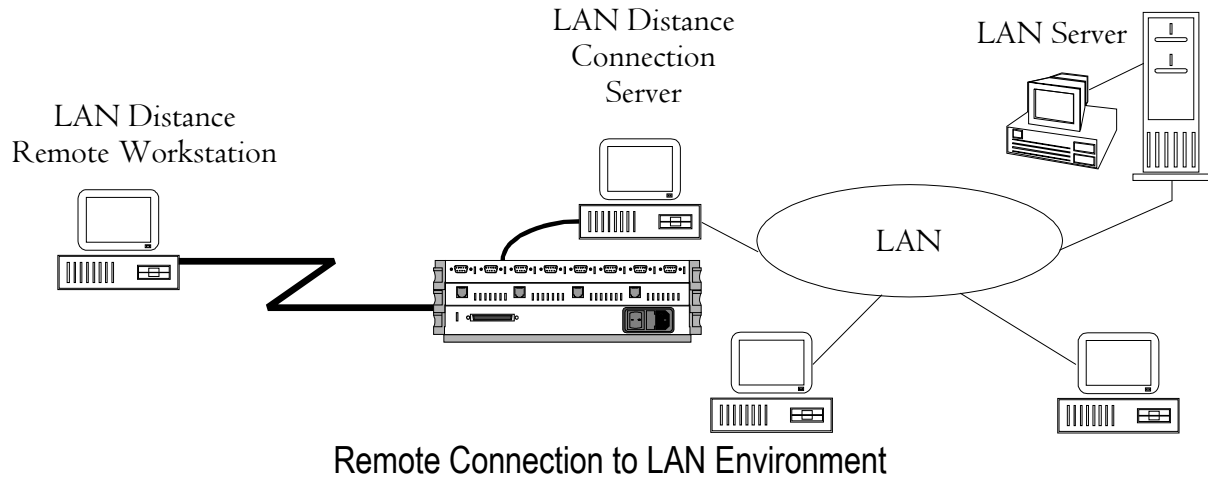
Remote to Remote Environment

The TCL LAN Distance Device driver has been designed to support up to eight TCL adapter cards, with a maximum of sixty-four ports per card. When used with the IBM OS/2 LAN Distance product the TCL hardware will allow asynchronous serial connections (either via Modems or ISDN Terminal Adapters) to remote users allowing them access to an individual computer, central computer, or network resources. The IBM Lan Distance product may also be used to bridge two remote computer networks together.

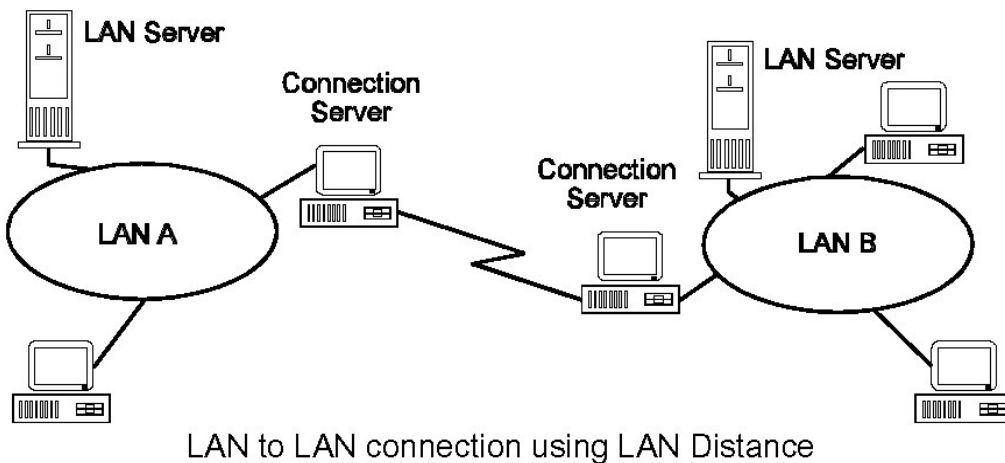


Stand-Alone Server in Remote to Remote Environment

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices



With the TCL Superport range of serial I/O hardware up to 64 serial ports can be supported from one ISA or MCA expansion slot. If the Superport system is using in built modems then a maximum of 32 channels can be supported from one card.



2. Quick LAN Distance Driver Installation

Install the TCL¹ Intelligent hardware according to the supplied instructions. (The hardware does NOT use I/O, DMA or IRQs)

Run the **INSTALL.COMD** program from the TCL Lan Distance Device Driver diskette, to install and prepare the TCL Driver software onto the hard disk. The system must then be re-booted for the software to recognise the TCL hardware.

Run the **TCLADDSP.EXE** program from the hard disk. All TCL ports will be added to the system.

The IBM OS/2 LAN Distance program may now be started. Select the "**SETTINGS**" notebook to display the installed ports and to assign modem types.

NOTE: If the TCL hardware includes built in modems use the **TCLBLAST** modem type setting.

After configuration of the LAN Distance settings, the system will need to be re-booted again before it can be used.

2.1 Quick LAN Distance Driver Removal

From the IBM OS/2 LAN Distance program select the **SETTINGS** notebook. Delete all TCL ports from the **PORTS** tab.

Insert the installation diskette and run the TCLREMOV.COMD program. Follow the displayed instructions which will remove the TCL software from the hard disk.

Edit the **CONFIG.SYS** file and remove the lines:

```
"DEVICE = \TCL\TCLDRV.OS2"  
"DEVICE = \TCL\TCLCOM.SYS"  
"DEVICE = \IBMCOMMACS\TCLXVMAC.OS2"
```

Shut down the system and remove all TCL Hardware.

¹ The TCL CC/4B and CC/8B buffered cards are not supported by the TCL LAN Distance Device Driver.

3. LAN Distance Driver Installation

3.1 Driver Installation Procedure

IMPORTANT Before proceeding with the TCL Driver installation, the TCL² adapter card **must** be installed in the computer. Please see the TCL Hardware Installation manual that came with the TCL adapter card for more information on installing the TCL adapter(s).

Install the IBM LAN Distance Software product (See the IBM LAN Distance Advanced Guide).

Version 1.1 and 1.11 of IBM LAN Distance program **MUST** have the Corrective Service Upgrades applied for LAN Distance to work correctly. (See IBM Service Pack IP08175 or later).

With IBM LAN Distance installed and service pack applied, run the LAN Distance program and select the **SETTINGS** notebook. Select the LAN Distance Logical Adapter to be:

0 - IBM OS/2 NetBIOS

All other Adapter types must be removed from this field.

NOTE: If the COM ports do NOT appear in the adapter list, then the OS/2 system is probably using an incorrect version of the **LAPS** program. To install a current version run the **PRODINST** program from the root directory of *OS/2 WARP Connect*, then select to install the **MPTS** (Multi-Protocol Transport Services) program. This will install new versions of **LAPS** and **MPTS**.

If LAN Server product is to be used with LAN Distance then some of the heuristic parameters in the IBMLAN.INI file must be modified. See the "*IBM LAN Distance Advanced Guide*" for current details.

(bit position -->) 012345678901234*678
SRVHEURISTICS = 1111014111131102133

SESSTIMEOUT = 300

(bit position -->) 01234567890***45678901234567890123
WRKHEURISTICS = 1111111121300011110001011120111221

Install the TCL Intelligent Serial I/O adapter card(s).

² The TCL CC/4B and CC/8B adapters are not supported by the TCL LAN Distance Device Driver.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

Place the **TCL Intelligent Drivers for LAN Distance** diskette into a diskette drive (typically Drive A:).

Open an OS/2 window and change to the diskette drive (by typing the drive letter followed by a colon (e.g. **A:<Enter>**),

then at the prompt type: **INSTALL A C<Enter>**

Where **A** is the diskette drive letter containing the TCL install diskette and **C** is the destination drive for the installation files. You should enter the appropriate drive letters for your own installation. (e.g. If you have placed the TCL Install diskette in drive B you should type **INSTALL B C<Enter>**).

```
[A:\] install a c

TCLADDSP v1.02.00 Add TCL Serial ports to
LAN Distance. © 1996 TCL
Please re-boot the system and run
TCLADDSP.EXE again to install ports.
```

Once the INSTALL program has completed, remove the TCL Installation diskette from the diskette drive. Exit the OS/2 text window (by typing "**EXIT**"), and select the Shut Down icon to close down the OS/2 operating system.

Once the System is down re-boot the computer (Type **Ctrl-Alt-Del** at the keyboard, or press the computers RESET button, or switch the Power off and on).

On boot up the TCL Hardware Interface driver (TCLDRV.OS2) will detect and record the number of TCL cards and ports found with in the system.

Select an OS/2 Text window and run the program **TCLADDSP**. All found TCL ports will be added to the system. It is possible to limit the number of ports added by passing a numeric parameter to the program. **NOTE:** It is not possible to add more TCL ports than physically exist in the computer.

```
[A:\] tcladdsp

TCLADDSP v1.02.00 Add TCL Serial ports to LAN
Distance. © 1996 TCL
8 TCL Ports have been successfully added to
the system.
LAN Distance may now be run to configure the
ports using the settings note book.

[A:\]
```

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

Other examples:

TCLADDSP	adds all TCL ports for LAN Distance use in to the system.
TCLADDSP 12	adds a maximum of 12 TCL ports for LAN Distance use into the system.

The IBM OS/2 LAN Distance program should now be started.

Select the **SETTINGS** notebook from the LAN Distance program to display the installed ports and to assign modem types. First select the "**Ports**" tab from the notebook to check that the TCL ports have been added to the system. The entries should be displayed as:

TCL0001	Superport-RISC	Port 1
TCL0002	Superport-RISC	Port 2
TCL0003	Superport-RISC	Port 3
etc...		

Assign the modem types to each TCL port as required. If the TCL hardware includes built in modems then use the **TCLBLAST** modem type setting for those ports.

To assign a modem type select the "**Modem**" tab. Then select the "**Assign**" tab to bring up a list of supported modem types. Select the required modem type. For TCL hardware with built in modem(s) select the **TCLBLAST** modem type.

If required a Phone number for the modem to use may also be entered. Then assign the modem to a port. **NOTE:** This must be done for ALL ports that are to be used with a modem attached to them.

Then select the "**Answer**" tab and set the answer mode you require LAN Distance to operate in. Typically this would be **Enable answer mode on startup**.

Close the settings note book, and answer **Yes** to the **save settings** prompt.

After configuration of the LAN Distance settings, shut down the OS/2 system, and re-boot the computer. When the system is re-booted it will then be configured for IBM OS/2 LAN Distance to use the TCL ports.

Note that this installation process will also automatically install the Serial COM driver (TCLCOM.SYS) for the TCL ports. This allows the TCL ports to be used as a general purpose Serial I/O device.

3.2 About the LAN Distance Installation Program

The installation program INSTALL.COMD actually does the following :

Copies the following files from installation diskette to the system drive:

Installation Diskette Files	Description	Destination Directory
TCLDRV.OS2	TCL H/W interface driver	\TCL\
TCLCOM.SYS	TCL Physical COM driver	\TCL\
TCLRISC.SYS	RISC card executive	\TCL\
TCL186.SYS	TCL 186 cards executive	\TCL\
TCLMODE.EXE	TCL Version of mode	\TCL\
TCLID.BIN	TCL card identification code	\TCL\
TCLHW.DEF	TCL card capability file	\TCL\
<hr/>		
TCLXVMAC.OS2	TCL MAC device driver	\IBMCOMMACS\
TCLXVMAC.NIF	TCL Network info file	\IBMCOMMACS\
<hr/>		
TCLBLAST.PIF	TCL Modem Definition file	\WAL\
TCLADDSP.EXE	TCL Install / Update program	\WAL\

Adds the lines to the CONFIG.SYS file:

**"DEVICE = \TCL\TCLDRV.OS2"
"DEVICE = \TCL\TCLCOM.SYS"**

3.3 Verifying the LAN Distance Installation

3.3.1 Overview

There are several methods of checking that the driver has been installed successfully and that it is functioning correctly :

Check the event log in system drive \IBMCOM\LANTRAN.LOG this should list the TCL ports that have been initialised by the LAN Distance driver. Correct TCL initialisation should be logged as:

```
Initialising   TCL001
Initialising   TCL002
Initialising   TCL003
.....         etc.
```

Check contents of CONFIG.SYS file should contain the following lines. Check that the drive letter defined is the one that the TCL and IBMCOM directories are residing on.

```
DEVICE=C:\TCL\TCLDRV.OS2
DEVICE=C:\TCL\TCLCOM.SYS          (<-This line optional)
DEVICE=C:\IBMCOMM\TCLXVMAC.OS2  (Added by LAN Distance)
```

Check the contents of the PROTOCOL.INI file has entries to match the available TCL ports. Typical entries should be:

```
[TCL001]
  DRIVENAME      = TCLXVMAC
  PORTNAME      = "TCL001"

[TCL002]
  DRIVENAME      = TCLXVMAC
  PORTNAME      = "TCL002"

[TCL003]
  DRIVENAME      = TCLXVMAC
  PORTNAME      = "TCL003"

.....  etc.
```

3.4 Adding an additional TCL adapter (for LAN Distance)

With the computer switched off and disconnected from the mains power supply, remove the cover and install the TCL adapter card according to the instructions supplied with the new TCL adapter card.

NOTE: Check that the memory address of the new adapter does not clash with any adapter within the computer, and the new adapter is at a **higher** memory address than any other TCL adapters currently installed in the computer.

Power up the computer and start the OS/2 operating system.. From an OS/2 window run the program TCLADDSP.

Run the IBM LAN Distance product and enable the ports (see section 3.1).

Close Down LAN Distance program - waiting for the "beep" tone indicating it has completed shut down, then Shut-Down the OS/2 operating system.

Re-boot the computer. The new ports will now be available for use with LAN Distance.

3.5 Removing a TCL Adapter (from LAN Distance)

Run the IBM OS/2 LAN Distance program.

From the LAN Distance program select the settings notebook. Select the **Ports** tab. Then **Delete** all TCL ports associated with the TCL adapter you wish to remove.

Shut down the OS/2 operating system.

Power down the computer and then remove the TCL adapter card.

3.6 Removing the TCL LAN Distance Device Driver

The TCL driver can be removed from the system by running the TCLREMOV.COM program which is on the TCL LAN Distance device driver diskette. The process of removing the driver basically consists of removing all the TCL files from the hard disk drive. The procedure is as follows :

Run the IBM OS/2 LAN Distance product, and select the **SETTINGS** notebook. Then **Delete** all TCL ports from the **Ports** tab.

Insert the TCL LAN Distance device driver diskette into drive A:.

Open an OS/2 window, log on to drive A: (by typing **A:<Enter>**), then run the program **TCLREMOV**.

TCLREMOV.COM should then display messages indicating whether or not it has removed the driver successfully.

Edit the **CONFIG.SYS** file and **remove³** the lines:

```
DEVICE = \TCL\TCLDRV.OS2
DEVICE = \TCL\TCLCOM.SYS
DEVICE = \IBMCOM\MACS\TCLXVMAC.OS2
```

Exit the OS/2 window (by typing "**EXIT**"), and select the Shut Down icon to close down the Operating System.

Power off the computer and remove the TCL adapter card(s).

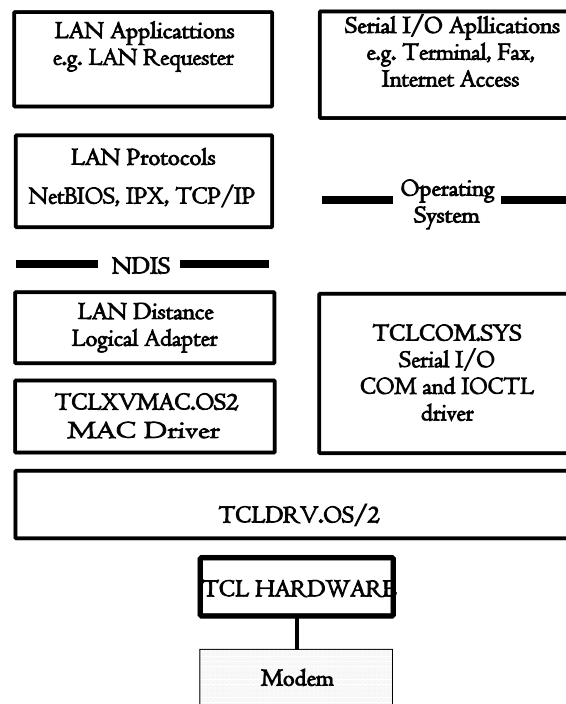
³ If you wish to remove LAN Distance but retain the TCL COM driver only remove the **DEVICE=IBMCOM\MACS\TCLXVMAC.OS2** line.

4 LAN Distance Device Driver Operation

4.1 Introduction

TCL supply three drivers on the TCL LAN Distance device driver diskette. The first driver is a physical device interface driver (TCLDRV.OS2) which provides an interface to the TCL hardware. This driver must be loaded before any of the other TCL device drivers.

The TCLXVMAC.OS2 device driver is a MAC level driver which provide an interface between the IBM LAN Distance Logical Adapter software and the TCL physical device interface driver.



The third driver TCLCOM.SYS provides an interface between the OS/2 serial I/O and IOCTL COM functions and the TCL physical device interface driver.

When the TCLDRV.OS2 driver is loaded it will scan for all TCL intelligent cards. If present it will use the file TCLCF.INI to determine if cards are to be omitted, or initialised with any special settings, such as initial baud rates, DTR/RTS handshake signal states, etc.

The TCLXVMAC.OS2 and TCLCOM.SYS drivers will only load if the TCLDRV.OS2 has been loaded successfully and initialised at least one TCL card.

4.2 Initial Port Settings

When the driver is loaded, if there is no driver configuration file (TCLCF.INI), it sets up all the TCL ports to the following default configuration :

Tx Buffer Size	256 bytes
Rx Buffer Size	1024 bytes
Baud Rate	19200 bps
Parity	None
Stop Bits	1
Data Bits	8
Flow Control	None
DTR/RTS	Low/Low

These default settings can be overridden by using a driver configuration file (see section 5).

4.3 Initial RX/TX Buffer Settings

The OS/2 TCLDRV.OS2 will allocate Receive and Transmit buffers in the TCL card memory. The default sizes are typically 256 bytes for the Transmit buffer and 1024 bytes for the Receive buffer. These default sizes can be supported by all the TCL hardware. Buffer sizes are only allocated when the driver is loaded. The default buffer sizes may be customised by creating an ASCII text configuration file (TCLCF.INI see Section 5) specifying the size of buffers to use. Buffers may be allocated on a per port basis.

When allocating RX/TX buffer size care should be taken not to specify more memory than is available on the TCL card, as this will lead to some ports becoming inoperable due to not enough memory.

As a guide the 186 based TCL cards (Superport-186, DataPump-4, DataServer-8/16, TwinSync) use approximately **32Kbytes** of memory for code and local data areas, the remaining on board memory may be used for allocating RX/TX buffers. The smallest buffer size allocatable is 16 bytes and the largest buffer allocatable is 65520 bytes.

The Superport-RISC adapter uses approximately **48Kbytes** of memory for code and local data areas, the remaining on board memory may be used for allocating RX/TX buffers. The smallest buffer size allocatable is 16 bytes and the largest buffer allocatable is 65536 bytes.

The TCL driver only reads the configuration file when the TCLDRV.OS2 is loaded, so for any changes made to the TCLCF.INI file to take effect the OS/2 system must be Shut-Down and then restarted.

5. Introduction COM Device Driver

The TCL Multiport cards provide multiple individual RS232 or RS422/485 I/O ports for the user. The TCL Intelligent and Direct OS/2 device drivers enable an application to access all of these ports through standard OS/2 operating system calls.

Access to the ports from a user program is common to all cards and is provided by file handling and device IOCTL calls.

Functions are provided to allow port communications parameters to be set up, ie. setting the baud rate, parity, flow control etc.

Installation and getting started is described in sections 6.1 and 6.2. Access to the ports via file handling and I/O control calls is covered in section 7.

6 Serial COM Driver - Getting Started

6.1 CC/4B & CC/8B Direct Card Driver Installation

To install the direct card and COM driver onto your OS/2 system first perform the following steps and then reboot the system.

Direct card hardware installation.

Refer to the hardware installation manual for instructions on how to do this, and Appendix 4 of this manual for any hardware restrictions that might apply. When installing the card make a note of the cards I/O address and IRQ number. (This can be determined from the card's switches and jumper settings and the PAL id.)

Driver software installation:

Insert the "TCL Utilities" diskette into your computers diskette drive (typically driver A:).

From the OS/2 desk top double click on the **OS/2 System** icon and from the displayed folder, double click on the **System Setup** icon.

In the **System Setup** folder the **Device Driver Install** icon should be displayed - double click on the **Device Driver Install** icon.

The Device Driver Install dialogue box is displayed showing the Source directory (*typically A:*) and the destination directory (*typically C:*). Check that the source directory is set to point to the diskette drive containing the "TCL Utilities" diskette, and the destination directory is pointing to the root directory of the OS/2 hard disk drive.

Click the **Install** button in the Device Driver Install dialogue box, which will list the device drivers that are available to install from the TCL Device driver diskette.

For CCPCI adapter card select:

TCL CCPCI Card OS/2 COM Device Driver OS2DDRV.SYS

For CC adapter card Select:

TCL CC Card OS/2 COM Device Driver OS2DDRV.SYS

Click on one of the above text lines to highlight it, then click the **Install** button. The system will then copy the appropriate files to the OS/2 hard disk and will update the CONFIG.SYS file.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

If a CC card has been selected the user must edit the CONFIG.SYS file and set up the /A snd /I parameters to match the Address and IRQ setting of the CC adapter card.

The driver installation is then complete.

The above installation process may be performed manually by copying the following files to the C:\TCL directory (create the directory if it does not exist):

OS2DDRV.SYS
TCLMODE.EXE

To tell OS/2 to load the device driver use the system editor to add the following line to the CONFIG.SYS file:

For ISA-bus CC card: **DEVICE=TCL\OS2DDRV.SYS /A0280,I03,B00**

For PCI-bus CCPCI card: **DEVICE=TCL\OS2DDRV.SYS /B00**

The numbers following A,I and B are hex digits and have the following meaning:

/Axxxx	The card I/O address. Noted when installing an ISA bus card. <i>(NOT required for CCPCI adapter cards.)</i>
Ixx	The card IRQ number. Noted when installing an ISA bus card. <i>(NOT required for CCPCI adapter cards.)</i>
Bxy	The receive/transmit buffer sizes where x=receive size and y=transmit size. The actual buffer size will be (x+1) * 100h bytes.
Fxx	An optional switch which may be added allowing the size of the UART FIFO buffer to be specified. Possible values are 01,04,08 and 0E.
Nxx	An optional switch which may be added allowing the renaming of the serial ports from Port001 .. Port008 to Comx .. Com(x+7). A leading zero is required. For example, to rename the serial ports as Com5 .. Com12, add N05 to the command line.

The IBM LAN Distance program cannot use the CC/4B or CC/8B ports. TCL do not support a CC/xB LAN Distance Device Driver.

The CC and CCPCI cards support baud rates up to 57600.

The address and IRQ allocated by the PCI-bus system will be displayed in the driver header on initialisation only.

6.2 Intelligent Card Installation

If you have installed the TCL LAN Distance Device driver, then that will have automatically installed the TCLCOM.SYS driver (for COM device support) and you do not need to follow this procedure.

To install the intelligent card and driver in your OS/2 system first perform the following steps and then reboot the system.

Intelligent card hardware installation:

Refer to the hardware installation manual that came with the TCL adapter card for information on installing the TCL adapter(s).

Driver software installation:

Insert the "TCL Utilities" diskette into your computers diskette drive (typically driver A:).

From the OS/2 desk top double click on the **OS/2 System** icon and from the displayed folder, double click on the **System Setup** icon.

In the **System Setup** folder the **Device Driver Install** icon should be displayed - double click on the **Device Driver Install** icon.

The Device Driver Install dialogue box is displayed showing the Source directory (*typically A:*) and the destination directory (*typically C:*). Check that the source directory is set to point to the diskette drive containing the "TCL Utilities" diskette, and the destination directory is pointing to the root directory of the OS/2 hard disk drive.

Click the **Install** button in the Device Driver Install dialogue box, which will list the device drivers that are available to install from the TCL Device driver diskette.

Select the "**TCL Intelligent Multiport OS/2 Device Driver TCLCOM.SYS**" driver by clicking on the text to highlight it, then click the **Install** button. The system will then copy the appropriate files to the OS/2 hard disk and will update the CONFIG.SYS file.

When the TCL driver installation is complete, exit the **Device Driver Install** dialogue box and close the **System Setup** and **OS/2 System** folders then reboot the system to load the TCL driver, which will then allow access to the TCL Serial I/O hardware.

NOTE: If you are using OS/2 v2 then from an OS/2 command line run the DDINSTAL utility and follow the programs instructions to install the TCL COM device driver.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

6.2.1 Intelligent Card COM Device Driver Removal

If you wish to remove the TCL LAN Distance Driver please refer to section 3.6

If you are using the TCL LAN Distance Device Driver and do NOT require the TCL Serial COM driver it is possible to remove the TCL Serial COM device Driver by editing the CONFIG.SYS file and remove the line:

DEVICE=\TCL\TCLCOM.SYS

This action will not affect the TCL LAN Distance Device Driver operation.

If you have not installed the TCL LAN Distance Device Driver and wish to remove the TCL COM device driver, then edit the CONFIG.SYS file and remove the two lines:

***DEVICE=\TCL\TCLDRV.OS2
DEVICE=\TCL\TCLCOM.SYS***

All files in the **\TCL** directory may then be deleted if the TCL drivers are no longer required.

6.2.2 Verifying COM driver Installation

The TCL Intelligent card COM Device Driver installation process actually does the following :

Copies the following files from installation diskette to the system drive:

Installation Diskette Files	Description	Destination Directory
TCLDRV.OS2	TCL H/W interface driver	\TCL\
TCLCOM.SYS	TCL Physical COM driver	\TCL\
TCLRISC.SYS	RISC card executive	\TCL\
TCL186.SYS	TCL 186 cards executive	\TCL\
TCLMODE.EXE	TCL version of mode	\TCL\
TCLID.BIN	TCL card identification code	\TCL\
TCLHW.DEF	TCL card capability file	\TCL\

Adds the lines to the CONFIG.SYS file:

"DEVICE = \TCL\TCLDRV.OS2"

“DEVICE = \TCL\TCLCOM.SYS”

6.2.3 TCL Intelligent COM Device Driver Operation

The TCL OS/2 COM Device Driver may either be used by itself or in conjunction with the TCL LAN Distance Device driver. For the TCL serial devices to be accessed as COM devices, both the **TCLDRV.OS2** device driver and the **TCLCOM.SYS** driver must be loaded in the system.

The TCLDRV.OS2 driver provides an interface between the TCL Intelligent Serial I/O adapters and the other TCL device drivers. This driver scans for the TCL hardware to determine how many TCL cards/ports are available in the system. The TCLDRV.OS2 driver will use an ASCII file TCLCF.INI (if present) to force the driver to omit certain cards, or initialise port parameters to certain values.

Cards are searched for in ascending address order. The “first” card will be the one found at the lowest memory address, and the first port found on that card, will be the “first” TCL port.

The TCLCOM.SYS device driver provides the COM interface allowing the OS/2 read/write and IOCTL functions to operate on TCL ports.

The TCL Driver software will manage access of multiple TCL drivers to a single port. Thus if a LAN Distance application is using a TCL port then the TCLCOM.SYS driver will be prevented from opening the same port as a COM device - until the LAN Distance driver has released the port.

6.2.4 TCL Intelligent COM Device Naming Convention

The TCLCOM.SYS device driver defaults to using **COM** for its device names. The driver will attempt to *open* a Device with the name **COM1**, if successful it assumes that the device already exists and that name cannot be used by a TCL port. It then increments the Device number (ie to **COM2**) and repeats the process, which is continued until the *open* returns an error - indicating that the device does not exist, in which case the name can be allocated to a TCL port. This process is repeated until all TCL ports found by the **TCLDRV.OS2** driver have been given a logical name by the **TCLCOM.SYS** driver. Typically on a machine containing COM1 and COM2 ports the first TCL port would appear as COM3.

A command line switch may be placed in the CONFIG.SYS file at the end of the line **DEVICE=\TCL\TCLCOM.SYS** which allows the user to define their own device name for TCL ports. This facility may also be used to force the TCL device names to start at a specific number.

DEVICE=\TCL\TCLCOM.SYS N=<name>{number}

The device name/number must not exceed eight characters/digits. Note in a large system the with 100 plus ports, three digits are used by the device number leaving five characters for the name.

DEVICE=\TCL\TCLCOM.SYS N=COM25 Force TCL port names to start at COM25.

DEVICE=\TCL\TCLCOM.SYS N=TCCOM

TCL port devices named TCCOM. The first device will be TCCOM1 assuming no other devices called TCCOM are present in the system.

DEVICE=\TCL\TCLCOM.SYS N=PORT

A special name kept for backward compatibility. The numbering system used with this name is always 3 digits using leading zeros where necessary. (i.e PORT001)

7 Programming a Serial COM Device

7.1 What are Device Drivers and Serial Devices?

A device driver is a piece of software that allows the operating system to access an installed piece of hardware in a standard way.

A serial device is a piece of hardware that converts parallel data on the PC into a serial data stream and vice versa.

The TCL device driver allows access to the serial devices (ports) on the TCL multiport cards through standard Application Programming Interface (API) calls. These APIs include file handling calls and serial I/O control calls.

A multiport card may, for instance, have 8 serial ports. To identify these ports in the system a unique name is given to each. Such as:

PORT001, PORT002... PORT008 or COM1, COM2, COM3 ... COM8

The file handling APIs allow the following simple data transfers to be performed:

<i>COPY filename.ext PORT001</i>	This will send the contents of a file to the first serial port to be transmitted.
<i>ECHO Hello >COM8</i>	This will redirect the string 'Hello' to the eighth serial port.
<i>TYPE PORT001</i>	This will display all data contained in the serial ports receive buffer.

7.2 Programming a Serial Device

7.2.1 Using Assembler to access serial device APIs.

The OS/2 assembler comes with include files containing macros which define the APIs. The serial device APIs are defined in the file DOSCALLS.INC. The implementation of the macros is in the library file DOSCALLS.LIB which must be included in your programs LINK statement.

The following API macro names can be used to access serial ports:

@DosOpen	Open a device session
@DosClose	Close a device session
@DosWrite	Transfer data to device
@DosWriteAsync	Transfer data to a device asynchronously
@DosRead	Transfer data from device
@DosReadAsync	Transfer data from a device asynchronously
@DosDevIOctl	Alter/request device parameters

These will be described in detail in the following sections.

7.2.2 Using 'C' to access serial device API's

As with assembler, 'C' comes with include files containing macros definitions of the API's. This file is usually named OS2.H.

To perform the Open, Close, Read and Write you will not need to access the APIs directly, just use the standard file handling calls defined in the 'C' run-time library ie:

 fopen(), fclose(), fread(), fwrite(), fprintf()...

or open(), close(), read(), write()...

The 'C' compiler will automatically use the APIs and link the correct libraries if you have instructed it to generate an OS/2 application.

Notes:

When using file handling calls, 'C' defines its own internal buffers, if you wish to access serial data from the port on a byte by byte basis use the function setbuf(portfp,NULL) after you open the device to disable this feature.

To confirm that your program is accessing a serial device name and not a disk file use the function isatty (porthandle).

Serial device I/O control is not part of the standard ANSI 'C' library and must be accessed directly using the macro @DosDevIOctl().

7.2.3 Suggested Receive and Transmit Algorithms

If you wish to transfer data to/from the serial devices the following pseudo 'C' procedure is one way of doing this.

Receiving data.

```
FILE *devicefp;          /* Device file pointer */
char rx_char;           /* Receive character */

{   if((devicefp = fopen("PORT001", "rb+") == NULL)
    {
        fprintf(stderr, "Open failed");
        exit(1);
    }
    setbuf(devicefp, NULL); /* Disable 'C'
    b_u_f_f_e_r_i_n_g */
    while((fread(&rx_char, sizeof(char), 1, devicefp) < 1);
          /* Poll the device */
          /* The received data may now be accessed */
    fclose(devicefp);
}
```

Transmitting data.

```
FILE *devicefp;          /* Device file pointer */
char *tx_string = "This is a test string";
                          /* Transmit string */

{   if((devicefp = fopen("PORT001", "rb+") == NULL)
    {
        fprintf(stderr, "Open failed");
        exit(1);
    }
    setbuf(devicefp, NULL); /* Disable 'C' buffering*/
    while((fwrite(tx_string, sizeof(char),
                  strlen(tx_string), devicefp)
           < strlen(tx_string));
          /* Poll the device */
    fclose(devicefp);
}
```

Note: A simpler way of doing this is:

```
fprintf(devicefp, "This is a test string");
```

7.3 Serial Device APIs

The following sections describe the serial device APIs in detail. The format is given as a macro definition. Full details of how to make a call and all possible values of parameter and return codes may be found in the appropriate IBM technical reference manual.

7.3.1 @DosOpen(name,handle,action,size,attrib,flag,mode,rsrvd)

WHERE:

name	ASCIIZ	Device name. Used to identify the serial device.
handle	WORD	Returned file handle.
action	WORD	Returned action.
size	DWORD	File size. (Not applicable)
attrib	WORD	File attribute
flag	WORD	Open flag, action taken if file exists, normally set to 1, open file.
mode	WORD	Open mode of file, normally set to 0xC2, inherit=true, share=deny none, access=read/write.
rsrvd	DWORD	Reserved. Must be set to zero.

PURPOSE: This function opens a session with the named serial device and returns a file handle which is required for all other API calls.

RETURNS: If 0 is returned there is no error, otherwise an error code is returned.

NOTE: The Open function causes various actions to be performed within the driver. See section 7.3.8 for more details.

7.3.2 @DosClose(handle)

WHERE:

handle	WORD	File handle obtained by open call.
--------	------	------------------------------------

PURPOSE: This function ends a previously opened session with a serial device.

RETURNS: If 0 is returned there is no error, otherwise an error code is returned.

NOTE: The Close function causes various actions to be performed within the driver. See section 7.3.8 for more details.

7.3.3 @DosRead(handle,buffer,len,bytesread)

WHERE:

handle	WORD	File handle obtained by open call.
buffer	BYTES	Area for data to be returned to.
len	WORD	Size of buffer area to be filled.
bytesread	WORD	Returned number of bytes read.

PURPOSE: To fill the buffer area with the requested number of bytes.

RETURNS: If 0 is returned there is no error, otherwise an error code is returned.

7.3.4 @DosReadAsync(handle,ramsem,return,buffer,len,bytesread)

WHERE:

handle	WORD	File handle obtained by open call.
ramsem	DWORD	RAM semaphore.
return	WORD	Returned Code.
buffer	BYTES	Area for data to be read to.
len	WORD	Size of buffer area to be filled.
bytesread	WORD	Returned number of bytes read.

PURPOSE: To fill the buffer area with the requested number of bytes, waiting for the result.

RETURNS: The returned information may not be read until the RAM semaphore is cleared by the system. If 0 is returned there is no error, otherwise an error code is returned.

7.3.5 @DosWrite(handle,buffer,len,byteswritten)

WHERE:

handle	WORD	File handle obtained by open call.
buffer	BYTES	Area of data to be written.
len	WORD	Size of buffer to be written.
bytesread	WORD	Returned number of bytes written.

PURPOSE: To write the specified number of bytes to the serial device.

RETURNS: If 0 is returned there is no error, otherwise an error code is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.6 @DosWriteAsync(handle,ramsem,ret,buff,len,byteswritten)

WHERE:

handle	WORD	File handle obtained by open call.
ramsem	DWORD	RAM semaphore.
ret	WORD	Returned Code.
buff	BYTES	Area of data to be written.
len	WORD	Size of data to be written.
byteswritten	WORD	Returned number of bytes written.

PURPOSE: To fill the buffer area with the requested number of bytes, waiting for the result.

RETURNS: The returned information may not be read until the RAM semaphore is cleared by the system. If 0 is returned there is no error, otherwise an error code is returned.

7.3.7 @DosDevIOCtl(data,param,function,category,handle)

WHERE:

data	----	Function dependant return buffer.
param	----	Function dependant parameter buffer.
function	WORD	I/O Function to be done.
category	WORD	Category of function.
handle	WORD	File handle obtained by open call.

PURPOSE: To perform I/O control on the specified device.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

OS/2 IOCTL Function Categories		
Category	Function	Description
0x01	0x41	Set Bit Rate
0x01	0x42	Set Line Characteristics
0x01	0x43	Set Extended Bit Rate
0x01	0x44	Transmit Byte Immediate
0x01	0x45	Set BREAK Off
0x01	0x46	Set Modem Control Signals
0x01	0x47	Behave as if XOFF received
0x01	0x48	Behave as if XON received
0x01	0x4B	Set BREAK On
0x01	0x53	Set Device Control Block Parameters
0x01	0x61	Query Bit Rate
0x01	0x62	Query Line Characteristics
0x01	0x63	Query Extended Bit Rate
0x01	0x64	Query COM Status
0x01	0x65	Query Transmit Data Status
0x01	0x66	Query Modem Output Signals
0x01	0x67	Query Modem Input Signals
0x01	0x68	Query No. of Characters in Receive Queue
0x01	0x69	Query No. of Characters in Transmit Queue
0x01	0x6D	Query COM Error
0x01	0x72	Query COM Event Information
0x01	0x73	Query Device Control Block Parameters
0x0A	0x40	Register Monitor
0x0B	0x01	Flush Input Buffer
0x0B	0x02	Flush Output Buffer
0x0B	0x60	Query Monitor Support

7.3.7.1 Set Bit Rate - Category 0x01 Function 0x41

PARAMETER PACKET:			
Bit Rate	WORD	Integer value of the bit rate. Possible values are:	
		50	1200
		62	1800
		75	2000
		110	2400
		134	3600
		150	4800
		200	7200
		300	9600
		600	19200
			38400
			57600

DATA PACKET: NULL

PURPOSE: To set the current bit rate of the device.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

If the device is BUSY transmitting, a general failure error is returned to the application. The application may retry after an appropriate delay.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.2 Set Line Characteristics - Category 0x01 Function 0x42

PARAMETER PACKET:		
Databits	BYTE	Set the number of data bits in each transmitted byte. 5 - 5 data bits 6 - 6 data bits 7 - 7 data bits 8 - 8 data bits
Parity	BYTE	Set the parity for each transmitted byte. 0 - no parity 1 - odd parity 2 - even parity 3 - mark parity 4 - space parity
Stopbits	BYTE	Set the number of stop bits for each transmitted byte. 0 - 1 stop bit 1 - 1.5 stop bits 2 - 2 stop bits

DATA PACKET: NULL

PURPOSE: To set the line characteristics of the serial device.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

If the device is BUSY transmitting, a general failure error is returned to the application. The application may retry after an appropriate delay.

7.3.7.3 Set Extended Bit Rate - Category 0x01 Function 0x43

PARAMETER PACKET:		
Bit Rate	DWORD	Integer value of the bit rate. Possible values are:
		50 1200 38400
		62 1800 57600
		75 2000 76800
		110 2400 115200
		134 3600
		150 4800
		200 7200
		300 9600
		600 19200
Fraction	BYTE	Integer value representing the fraction of the bit rate to be set. Usually set to 0.

DATA PACKET:
NULL

PURPOSE: To set the current bit rate of the device.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

If the device is BUSY transmitting, a general failure error is returned to the application. The application may retry after an appropriate delay.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.4 Transmit Byte Immediate - Category 0x01 Function 0x44

<u>PARAMETER PACKET:</u> Character BYTE Character to be transmitted immediately.

<u>DATA PACKET:</u> NULL

PURPOSE: To send the specified character immediately ahead of any other data waiting to be transmitted. The character will not be sent immediately if flow control is preventing transmission or if BREAK is being sent.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

This function will return immediately with a general failure error if the character can not be sent.

7.3.7.5 Set Break Off - Category 0x01 Function 0x45

<u>PARAMETER PACKET:</u> NULL

<u>DATA PACKET:</u> COM error WORD This status word is returned if there is no error returned. See function 0x6D - Query COM Error for definition of this word bits.
--

PURPOSE: Stops transmitting the BREAK signal.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.6 Set Modem Control Signals - Category 0x01 Function 0x46

<u>PARAMETER PACKET:</u> ON Mask BYTE	This mask indicates the signals to be raised. Where: Bit: 7..2 Undefined 1 RTS 0 DTR
OFF Mask BYTE	This mask indicates the signals to be lowered. If the ON and OFF mask are both set for a given signal then that signal is raised.

<u>DATA PACKET:</u> COM error WORD	This status word is returned if there is no error returned. See function 0x6D - Query COM Error for definition of this word bits.
---------------------------------------	---

PURPOSE: To raise/lower the modem output signals.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

7.3.7.7 Behave as if XOFF received - Category 0x01 Function 0x47

<u>PARAMETER PACKET:</u> NULL

<u>DATA PACKET:</u> NULL

PURPOSE: If XON/XOFF flow control has been configured, then this function causes transmission to be stopped.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.8 Behave as if XON received - Category 0x01 Function 0x48

PARAMETER PACKET:
NULL

DATA PACKET:
NULL

PURPOSE: If XON/XOFF flow control has been configured, then this function causes transmission to be started.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

7.3.7.9 Set Break On - Category 0x01 Function 0x4B

PARAMETER PACKET:
NULL

DATA PACKET:
COM error WORD

This status word is returned if there is no error returned. See function 0x6D - Query COM Error for definition of this word bits.

PURPOSE: Starts transmitting the BREAK signal.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.10 Set Device Control Block Parameters - Category 0x01 Function 0x53

PARAMETER PACKET:

Write timeout WORD The timeout period in units of 10mS to be used with write processing.

Read timeout WORD The timeout period in units of 10mS to be used with read processing.

Flags1 BYTE This has the following bits:

7	Reserved	
6	DSR input sensitivity	
5	DCD output handshaking	
4	DSR output handshaking	
3	CTS output handshaking	
2	Reserved	
1,0	DTR control mode	11 = invalid 10 = input handshaking 01 = enabled 00 = disabled

Flags2 BYTE This has the following bits:

7,6	RTS control mode	11 = toggling on transmit 10 = input handshaking 01 = enabled 00 = disabled
5	undefined	
4	enable BREAK replacement	
3	enable NULL stripping	
2	enable ERROR replacement	
1	XON/XOFF Rx flow control	
0	XON/XOFF Tx flow control	

Flags3 BYTE This has the following bits:

7..3	undefined	
2,1	read timeout processing	11 = no timeout 10 = wait infinite 01 = normal timeout 00 = invalid
0	write timeout processing	0 = normal timeout 1 = wait infinite

ERROR Char. BYTE This defines the character to be returned when error replacement is enabled.

BREAK Char. BYTE This defines the character to be returned when break replacement is enabled.

XON Char. BYTE This defines the XON character to be used with XON/XOFF flow control.

XOFF Char. BYTE This defines the XOFF character to be used with XON/XOFF flow control.

DATA PACKET: NULL

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

PURPOSE: To set various handshaking, flow control and transfer characteristics of the serial device.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

If the device is BUSY transmitting, a general failure error is returned to the application. The application may retry after an appropriate delay.

7.3.7.11 Query Bit Rate - Category 0x01 Function 0x61

PARAMETER PACKET:
NULL

DATA PACKET:
Bit rate WORD Integer value representing the current bit rate.

PURPOSE: To return the current bit rate of the serial device.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

7.3.7.12 Query Line Characteristics - Category 0x01 Function 0x62

PARAMETER PACKET:
NULL

DATA PACKET:

Databits	BYTE	The current data bits setting.
Parity	BYTE	The current parity setting.
Stopbits	BYTE	The current stop bits setting.
Tx break	BYTE	If 1 then transmitting break, else not.

PURPOSE: To return the current line characteristics of the serial device. Refer to function 0x42 - Set Line Characteristics for definitions of the bytes.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.13 Query Extended Bit Rate - Category 0x01 Function 0x63

PARAMETER PACKET:
NULL

DATA PACKET:		
Current bit rate	DWORD	Integer value representing the current bit rate.
Current fraction	BYTE	Integer value representing the current fraction.
Minimum bit rate	DWORD	Integer value representing the minimum bit rate.
Minimum fraction	BYTE	Integer value representing the minimum fraction.
Maximum bit rate	DWORD	Integer value representing the maximum bit rate.
Maximum fraction	BYTE	Integer value representing the maximum fraction.

PURPOSE: To return the current, minimum and maximum bit rate of the serial device.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

7.3.7.14 Query COM Status - Category 0x01 Function 0x64

PARAMETER PACKET:
NULL

DATA PACKET:		
COM Status	BYTE	This status byte has the following bits:
		7 Rx waiting for DSR to turn on
		6 Char waiting to Tx immediately
		5 Tx waiting as BREAK Tx'ing
		4 Tx waiting as XOFF Tx'ing
		3 Tx waiting as XOFF Received
		2 Tx waiting for DCD to turn on
		1 Tx waiting for DSR to turn on
		0 Tx waiting for CTS to turn on

PURPOSE: To return the current receive/transmit data status.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.15 Query Transmit Data Status - Category 0x01 Function 0x65

PARAMETER PACKET:
NULL

DATA PACKET:
Transmit Status BYTE This status byte has the following bits:

7..6	Undefined
5	Waiting to transmit XOFF
4	Waiting to transmit XON
3	Char waiting to Tx immediately
2	Hardware currently transmitting data
1	Data in device transmit queue
0	Write request queued

PURPOSE: To return the current transmit data status of the serial device.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

7.3.7.16 Query Modem Output Signals - Category 0x01 Function 0x66

PARAMETER PACKET:
NULL

DATA PACKET:
Output signals BYTE This status byte indicates the current state of the modem output signals:

7..2	undefined
1	RTS
0	DTR

PURPOSE: To return the current state of the modem control output signals.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.19 Query No. of Characters in Transmit - Category 0x01 Function 0x69

PARAMETER PACKET:
NULL

DATA PACKET:

No. of chars WORD Number of characters in the transmit queue.
Queue size WORD Size of the transmit queue.

PURPOSE: To return the number of characters in the transmit queue and the size of the transmit queue.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

7.3.7.20 Query COM Error - Category 0x01 Function 0x6D

PARAMETER PACKET:
NULL

DATA PACKET:
COM Error WORD

This status word has the following bits:

15-4	undefined
3	hardware framing error
2	hardware parity error
1	receive hardware overrun
0	receive queue overrun

PURPOSE: To return the current COM error status of the serial device. The COM Error is cleared once read.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.21 Query COM Event Information - Category 0x01 Function 0x72

PARAMETER PACKET:
NULL

DATA PACKET:

COM Event WORD

This status word has the following bits:

15-9	undefined
8	trailing edge of RI
7	parity, framing, overrun error
6	Break detected
5	DCD changed state
4	DSR changed state
3	CTS changed state
2	last char in Tx queue sent
1	receive hardware timeout
0	characters in receive queue

PURPOSE: To return COM Event information from the serial device. The word is cleared once read.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.22 Query Device Control Block Parameters - Category 0x01 Function 0x73

PARAMETER PACKET:			
NULL			
DATA PACKET:			
Write timeout	WORD	The timeout period in units of 10mS for normal WR processing.	
Read timeout	WORD	The timeout period in units of 10mS for normal RD processing.	
Flags1	BYTE	7	Reserved
		6	DSR input sensitivity
		5	DCD output handshaking
		4	DSR output handshaking
		3	CTS output handshaking
		2	Reserved
		1,0	DTR control mode
			11 = invalid
			10 = input handshaking
			01 = enabled
			00 = disabled
Flags2	BYTE	7,6	RTS control mode
			11 = toggling on transmit
			10 = input handshaking
			01 = enabled
			00 = disabled
		5	undefined
		4	enable BREAK replacement
		3	enable NULL stripping
		2	enable ERROR replacement
		1	XON/XOFF Rx flow control
		0	XON/XOFF Tx flow control
Flags3	BYTE	7..3	undefined
		2,1	read t/o processing
			11 = no timeout
			10 = wait infinite
			01 = normal timeout
			00 = invalid
		0	write timeout processing
			0 = normal timeout
			1 = wait infinite
ERROR Char.	BYTE	This defines the character to be returned when error replacement is enabled.	
BREAK Char.	BYTE	This defines the character to be returned when break replacement is enabled.	
XON Char.	BYTE	This defines the XON character to be used with XON/XOFF flow control.	
XOFF Char.	BYTE	This defines the XOFF character to be used with XON/XOFF flow control.	

PURPOSE: To return the current Device Control Block (DCB) parameters.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.23 Register Monitor - Category 0x0A Function 0x40

PARAMETER PACKET:
Command Info BYTE

DATA PACKET:
Placement Flag WORD
Index WORD
I/P buffer add DWORD
O/P buffer off WORD

PURPOSE: To register a monitor. This function is NOT supported.

RETURNS: If 0 is returned there is no error, if there is a problem with the PARAM or DATA buffers a general failure error is returned.

7.3.7.24 Flush Input Buffer - Category 0x0B Function 0x01

PARAMETER BUFFER:
Command Info BYTE Must be set to zero.

DATA BUFFER:
Reserved BYTE Set to zero.

PURPOSE: To flush the input buffer.

RETURNS: None

7.3.7.25 Flush Output Buffer - Category 0x0B Function 0x02

PARAMETER BUFFER:
Command Info BYTE Must be set to zero.

DATA BUFFER:
Reserved BYTE Set to zero.

PURPOSE: To flush the output buffer.

RETURNS: None

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

7.3.7.26 Query Monitor Support - Category 0x0 Function 0x60

PARAMETER BUFFER:

Command Info BYTE Must be set to zero.
--

DATA PACKET:

Reserved BYTE Set to zero.

PURPOSE: To determine if monitors can be supported by the driver.

RETURNS: If 0 is returned, then monitors are supported, otherwise the system error MONITORS NOT SUPPORTED is returned.

7.3.8 Open/Close API Effects on Driver

Open and Close calls cause various actions to be performed within the driver. A record is kept of the number of Open/Close calls made to a given device. Many open/close calls may be made to a given device. The first open (first level open) and the last close (last level close) are special cases. The following sections detail the actions performed in these cases.

7.3.8.1 First Level Open

The following actions are performed on a first level open:

- If DTR is set to enable or handshaking, then it is set high.
- If RTS is set to enable or handshaking, then it is set high.
- If transmit XON/XOFF flow control is set, then assume that XON has been received.
- Set the XON character to be 0x11, and the XOFF character to be 0x13.
- Disable break replacement, error replacement and null stripping.
- Set break and error replacement characters to be 0.
- Set read and write timeout periods to be 1 minute.
- Set read timeout to be normal timeout processing.
- Clear the COM error and COM event words.

7.3.8.2 Last Level Close

The following actions are performed on a last level close:

- The driver will wait until all characters have been transmitted, or until a transmit timeout occurs for the period set in the DCB.
- When all characters have been transmitted/timeout, then DTR/RTS are lowered.
- If receive XON/XOFF flow control is enabled, then XON is transmitted.
- Set break condition to be off.

8 The Driver Configuration File

8.1 Introduction

The driver configuration file is a user-supplied ASCII text file which specifies the configuration and port setup information for all TCL cards in the system. A configuration file can be created using any text editor that creates plain ASCII text. The name of the file should be **TCLCF.INI**

The only reason to create a configuration file is if the user wishes to change any of the initial default settings of the TCL driver.

The main uses of the driver configuration file are as follows :

- To specify non-default transmit and receive buffer sizes. The default buffer size is 256 bytes for the transmit buffer and 1024 bytes for the receive buffer.

- To limit the number of ports used on a particular card. The default is to use all ports found on the card.

- To specify which TCL cards in the system will be used. The default is to use all found cards up to a maximum of 8.

- To specify the default port configuration when the card is loaded. **Note** that this may subsequently be changed by any system software using the ports.

- To specify the use of a non-standard executive to load onto a TCL card.

The configuration file TCLCF.INI must be placed in the same directory as the driver file TCLDRV.OS2

8.2 Configuration File Syntax

The configuration file consists of paragraphs which begin with key words identifying the paragraph type. The following paragraph types are permitted :

setcard
setport

The *setcard* paragraph is used to indicate the card number, the number of ports to use on that card and the name of the executive to load onto the card. Each *setcard* paragraph may be followed by a number of *setport* paragraphs.

The *setport* paragraph is used to specify port setup information for the ports on the card mentioned in the immediately preceding *setcard* paragraph.

The paragraph type key word is followed by a number of optional parameters as described in the following sections.

Note that if there is more than one TCL card in the system, the administrator must know the address of each TCL card in the system in order to be able to create a configuration file. This is because the cards are referred to in ascending address order.

8.3 The Setcard Paragraph

The *setcard* paragraph has the following format :

setcard *[cX]* *nports=Y* *exec=Z*

[cX] Here X is the card number and can have a value from 1 to 8. Note that cards are sorted by their physical address with the card at the lowest address being Card 1. The *[cX]* parameter must be present.

nports=Y The value Y is used to indicate how many ports to use on this card. The *nports* parameter is optional. If not present, the number of ports actually found on the card will be used. Note that the ports used always start at the first port on the card.

exec=Z Here Z is the filename of the executive to be loaded onto the card. The full path to the file should be specified.

Some examples of valid *setcard* paragraphs :

setcard [c1] nports=32 exec=c:\ tcl \ myexec.sys

setcard [c2] nports=8

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

If the driver finds a valid configuration file, then only those TCL cards mentioned in the configuration file are loaded with an executive and used by the driver. If a configuration file is not found, all found TCL cards are loaded with TCL186.SYS for Intel based cards or TCLRISC.SYS for RISC based cards.

Note that a minimal configuration file just contains *setcard* paragraphs with card numbers saying which cards are to be used. For example, if only card 2 and card 5 are required, then the following configuration file might be appropriate :

setcard [c2]
setcard [c5]

8.4 The Setport Paragraph

The *setport* paragraph has the following format :

setport [PN] baud=B, wordlen=W, stopbits=S, parity=P, prot=F, rxbuff=R, txbuff=T

[PN] The value *N* indicates the port being referred to and can take any value from 1 to the number of ports specified in the preceding *setcard* paragraph. A group of ports is specified using the format *[PN-M]* (where $M \geq N$).

baud=B *B* is an integer equal to the requested baud rate. Although this may be any value, the driver will round down non-standard values to the next lowest from the following set :

50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2000,
2400, 3600, 4800, 7200, 9600, 19200, 38400, 57600, 76800,
115200.

wordlen=W This parameter specifies the number of databits. Permitted values for *W* are :

5, 6, 7, 8.

stopbits=S The value of *S* specifies the number of stopbits. Permissible values :

1, 1.5, 2.

parity=P *P* indicates the parity mode with the following permissible settings :

none, even, odd, mark, space.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

prot=F This parameter specifies the flow control method to use for the port. If more than one flow control method is needed, this can be specified by specifying multiple values separated by a plus sign (e.g. *txxon+cts+dtr*). Any sensible combination of the following values is allowed :

<i>txxon</i>	Transmit XON/XOFF.
<i>rxxon</i>	Receive XON/XOFF.
<i>dtr</i>	Receive DTR.
<i>rts</i>	Receive RTS.
<i>dsr</i>	Transmit DSR.
<i>cts</i>	Transmit CTS.
<i>dcd</i>	Transmit DCD.
<i>none</i>	No flow control.

rxbuff=R *R* is an integer indicating the required size in bytes of the receive buffer. The smallest permitted buffer size is 16. The upper limit is 64Kb but also depends on the amount of memory on the card.

txbuff=T *T* is an integer indicating the required size in bytes of the transmit buffer. The smallest permitted buffer size is 4. The upper limit depends on the amount of memory on the card.

rts=r This field indicates the initial state for RTS. The parameter *r* can be either *on* or *off*. The default is *rts=off*.

dtr=d This field indicates the initial state for DTR. The parameter *d* can be either *on* or *off*. The default is *dtr=off*.

break=b The parameter *b* can be either *on* or *off*. The *break=on* statement if used, means that the port is initialised with the transmitter in the break state. The default is *break=off*.

The *baud*, *wordlen*, *stopbits*, *parity*, *prot*, *rxbuff* and *txbuff* parameters are all optional and may appear in any order.

Examples of valid *setport* paragraphs are as follows :

***setport* [p15] baud=9600, wordlen=7, parity=odd, txbuff=512**

***setport* [p16-17] baud=38400, wordlen=8, stopbits=1, parity=none, prot=rts+cts, rxbuff=256, txbuff=512**

***setport* [p18-19] baud=19200**

The initial default values are as follows :

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

baud=19200, wordlen=8, stopbits=1, parity=none, prot=none, rxbuff=1024, txbuff=256, dtr=off, rts=off, break=off

Note that the *setport* paragraphs are dealt with sequentially and that if not all parameters are specified in a *setport* paragraph, it is assumed that the parameters not specified will retain their default value. Therefore if a port is named implicitly or explicitly more than once, any parameter not mentioned in a later *setport* paragraph will not be altered from what it was set to in earlier *setport* paragraphs.

This is illustrated in the following example where the end result is that port 1 on a card is set to 19200 baud, 8 bits, 1 stopbit, even parity, DTR/DSR flow control, with a 256 byte transmit buffer and 1024 byte receive buffer :

***setport [P1-8] parity=even
setport [P1] prot=dtr+dsr, rxbuff=1024***

8.5 Example Configuration File

This section illustrates the use of a driver configuration file. A configuration file is not supplied on the driver distribution disk since the contents of a driver configuration file depend entirely on the user's individual needs.

In this example configuration, a system contains 24 TCL ports as follows :

DataServer-8 (8 ports)	(Address D6000)
Superport-186, two layers (16 ports)	(Address D8000)

There may also be other (non-TCL) serial ports in the system but the configuration file affects only the TCL driver.

```
setcard [C1]          nports=4
setport  [P1-4]       txbuff=4096, rxbuff=16384

setcard [C2]
setport  [P1-2]       txbuff=256, rxbuff=256, prot=none,baud=300,
                    wordlen=7, stopbits=1,
                    parity=mark
setport  [P3-4]       txbuff=256,
                    rxbuff=256,prot=txxon+rxxon+rts+cts,baud=2400,
                    wordlen=7, stopbits=1,parity=mark
setport  [P5-6]       txbuff=4095, rxbuff=4095,
                    prot=txxon+rxxon,baud=4800
setport  [P7-8]       txbuff=4095, rxbuff=4095, prot=none,baud=9600,
                    wordlen=7
setport  [P9-10]      prot=none,baud=19200, wordlen=8,
                    stopbits=1,parity=odd
setport  [P11]        prot=dtr+dsr,baud=115200
```

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

In this example, the system administrator decides to use all the ports on the Superport-186 but only ports 1 to 4 on the DataServer-8. The example configuration file shows how the administrator can set this up and at the same time set individual ports and groups of ports to default to different configurations when they are initialised.

Note that the DataServer-8 card is at the lower address so it is referred to as card 1 in the configuration file.

Some points to note about this example file :

- The *nports=4* statement is necessary for card 1 since not all the ports on the card are being used.
- The *txbuff=* statement is the only way to achieve buffer sizes other than the default. The buffer size is set when the driver is loaded and cannot be altered afterwards without reloading the driver.
- Ports 12 to 16 on the Superport-186 are set to the default configuration.

APPENDICES

Appendix 1 Installation Disk Contents

The install diskette for the TCL LAN Distance and COM device drivers contains the following files :

READ.ME	Release notes
OS2DDRV.SYS	TCL CC/4B and CC/8B COM Device Driver
INSTALL.CMD	TCL LAN Distance Driver install program
TCLREMOV.CMD	TCL LAN Distance Driver removal program
TCLADDSP.EXE	TCL LAN Distance Configuration program
TCLOS2.DDP	TCL COM Device Installation Script
TCLDRV.OS2	TCL OS/2 Physical Device Driver
TCLCOM.SYS	TCL OS/2 Serial COM Device Driver
TCLXVMAC.OS2	TCL OS/2 MAC Driver for LAN Distance
TCLRISC.SYS	Executive for TCL Superport-RISC cards
TCL186.SYS	Executive for TCL 186-based cards
TCLMODE.EXE	TCL version of MODE.EXE
TCLID.BIN	Card identification file - used by the driver
TCLHW.DEF	Card capabilities database - used by the driver

Appendix 2 **TCLMODE** command

This is a serial port configuration utility allowing the user to set the ports characteristics and handshaking from the OS/2 command line. The syntax of the command is identical to that of the OS/2 **MODE** command (when used with serial ports) and is as follows:

TCLMODE <device>: {baud},{parity},{databits},{stopbits},
 {options}

Where:

<device>	-	name of the serial device to be configured
{baud}	-	bit rate (50..115200)
{parity}	-	parity (N,O,E,M,S)
{databits}	-	data bits (5,6,7,8)
{stopbits}	-	stop bits (1,1.5,2)
{options}	-	protocol settings as follows:
to=on off	-	infinite tx timeout on or off
xon=on off	-	XON/XOFF input/output flow control
idsr=on off	-	DSR input sensitivity
odsr=on off	-	DSR output handshaking
octs=on off	-	CTS output handshaking
dtr=on off hs	-	DTR on,off or handshaking
rts=on off hs tog	-	RTS on,off,handshaking or toggle on transmit

Example: To set serial port 3 to 19200 baud with XON/XOFF flow control:

TCLMODE port003:19200,,,,xon=on

Note: If N=COM is used with the CONFIG.SYS declaration of the driver generating COM1, COM2... devices, then the OS/2 MODE command may be used for devices COM1 to COM8.

Appendix 3 Troubleshooting and Hints

This section details some of the commonly occurring problems and their solutions.

TCL Intelligent COM Driver TCLDRV.OS2 / TCLCOM.SYS

The following error messages may be displayed during initialisation:

- MESSAGE: "ERROR : TCL Intelligent Multiport Device Driver vX.XX.X Failed to Install"
- CAUSE: The TCLDRV driver did not find any TCL intelligent adapter cards or an adapter card was found but no ports were detected on the adapter card.
- REMEDY: Install the TCL card, or check the switch settings.
Check the TCL adapter does not conflict with other hardware, eg: SCSI Disk adapter or Network Controller cards.
Disable ROM Shadowing / Cache Memory (ROM Bios Setup).
If Plug-and-Play Bios, then enable High Memory block occupied by TCL adapter.
If using a Superport or PCC/IF adapter check the external unit is connected to the correct TCL adapter card, and that the unit is powered on.
- MESSAGE: "ERROR : Unable to open TCLDRV\$"
- CAUSE: The TCLDRV.OS2 driver has not been loaded.
- REMEDY: Check the line **DEVICE=\TCL\TCLDRV.OS2** exists in the CONFIG.SYS file, and that it is declared on the line **before** the **TCLCOM.SYS** declaration.
Check the file TCLDRV.OS2 exists in the directory as specified in the **DEVICE=.....\TCLDRV.OS2** line in CONFIG.SYS.
- MESSAGE: "ERROR : No ports found on the card"
- CAUSE: The driver was unable to detect any serial ports on the multiport card.
- REMEDY: If using a Superport-186 or Superport-RISC card, check that the cable to the Superport unit is correctly seated and that the power light on the unit is on. Otherwise check that you have the correct multiport card.
- MESSAGE: "ERR_BAD_PARAM Unable to import file *FILENAME.EXT* image for loading"
- CAUSE: The loader was unable to locate the executive program to load onto the card, or the name specified in the TCLCF.INI file (if present) contains an error.
- REMEDY: Ensure the *FILENAME.EXT* file exists in the SAME directory as the TCLDRV.OS2 driver or if a path has been specified in the TCLCF.INI file check that the *FILENAME.EXT* file is in that path.
- MESSAGE: "ERR_BAD_EXECVER: Executive version X.X.X.X is not compatible with this software."
- CAUSE: The executive file has been corrupted, or an incorrect copy of the executive file has been copied into the TCL driver directory, or a problem has occurred with the multiport hardware.

TCL Serial Device Driver for OS/2 Lan Distance and COM Devices

REMEDY: Check that the executive file has not been corrupted, or contact TCL. If you have been supplied with diagnostic software then run this first to check there is no hardware problem.

If problems persist contact TCL.

Direct Driver OS2DDRV.SYS for CC/4B and CC/8B cards.

The following messages may be displayed during initialisation:

MESSAGE: "Bad parameter on the command line"

CAUSE: A syntax error has occurred in the CONFIG.SYS declaration.

REMEDY: Check the declaration against the instructions in Section 6.

MESSAGE: "The address specified was incorrect"

CAUSE: The address specified in the /Axxxx CONFIG.SYS declaration is not supported by the driver.

REMEDY: Refer to the installation instructions and the hardware manual to select a valid address.

MESSAGE: "The fifo mode specified was incorrect"

CAUSE: The level specified in the Fxx CONFIG.SYS declaration was not a valid value.

REMEDY: Refer to the installation instruction for a correct value.

MESSAGE: "Unable to set fifo mode on card"

CAUSE: The multiport card UARTs do not support fifo mode operation.

REMEDY: Check that you have the correct multiport card.

MESSAGE: "Unable to install hardware interrupt"

CAUSE: Another device is already using the requested interrupt.

REMEDY: Refer to the installation notes and the hardware manual to select a different hardware interrupt on the card.

MESSAGE: "The card has failed the tests"

CAUSE: Internal tests which confirm that a card is present at the address specified have failed.

REMEDY: Check the CONFIG.SYS declaration against the cards switch settings.

If problems persist contact TCL.

Using the Device Driver with Commercial Applications

If the application allows the serial device to be accessed by name, then use the devices name, displayed during TCL driver initialisation of the TCL driver.

If the application is restricted to using COM1, COM2... the TCL Intelligent COM Device Driver defaults to using the COM name. Different Device names can be defined by using the N=*name* switch in the CONFIG.SYS declaration. (Note: this feature is not available with direct CC/4B and CC/8B card drivers)

If the application uses the MODE utility to configure the serial ports, then replace this with the TCLMODE utility which uses the same syntax.

If the application is using the serial ports to communicate with a modem or similar, then check that the multiport card supports all of the signals required. See the hardware installation manual for details of connections.

Appendix 4 Version Specific Restrictions

The following list of functions and features are **not** supported by the current versions of the drivers.

Intelligent Driver TCLCOM.SYS Version 1.1.1

Device Monitors are NOT supported.

Direct Driver OS2DDRV.SYS Version 1.0h

The driver only supports the following I/O address combinations:

BASE I/O ADDRESS	LATCH	PORTS
01A0	01BF INVERTED	4
02A0	02BF INVERTED	4
0180	0187	8
0280	0287	8
02C0	02C7	8

Please refer to the hardware installation manual supplied with your card to determine which addresses are available and the switch settings needed to obtain them.

NOTES
