# EMTS Installation Guide
# Version 2.10

# 1 Introduction

This guide is intended for people who wish to install the The Ethernet Multi-user Transputer System (EMTS). Please refer to the **EMTS User Guide** for an overview of the system and for general user information.

The next section gives a brief overview of the EMTS distribution. Section 3 describes the EMTS system requirements. Section 4 explains how to install the EMTS software. Section 5 explains how to configure the EMTS software. Section 6 gives guidance to software developers who wish to use the `emtslink` module to build new EMTS server programs.

# 2 Overview

The EMTS provides network access to a collection of transputer resources. The EMTS *stub* program (`estub`) runs on an IBM PC and hosts a transputer resource (one or more *target* transputer systems). An EMTS *server* program (`eserver` or `iserver`, see below) runs on an IBM PC or a Sun workstation and allows the user to access a remote target transputer system. The stub transparently relays data between server(s) and target(s). An EMTS *who* program (`ewho`) reports status information on who is using the transputer resources.

The EMTS distribution also includes:

- A number of utilities (eg start-up scripts, `ITERM` files) that support the Inmos toolsets and the Inmos D700E (TDS3) system.
- Example programs on how to exploit the server extensions provided by the `eserver`.
- Unix manual pages.
- Documentation in PostScript format.
- Source code (in the C language) for the `emtslink` module and other EMTS programs.

Two servers are provided with the distribution: the EMTS `eserver` and the Inmos `iserver`[1]. The `iserver` is the Inmos version 1.42h which has been linked with the EMTS link module (`emtslink`) to allow it to access remote transputers. The `eserver` is an enhanced transputer server program that can be used in place of the `iserver`. The `eserver` provides the following extensions to the Inmos `iserver`:

- Support for client-initiated `xterm` terminal emulator processes when the server is running under MIT's X11 or Sun's OpenWindows windowing environments. Each `xterm` process

---

1 *Support for other servers (eg `afserver`) will be added in a future release of the EMTS.*

gives its client a private pseudo-terminal for terminal i/o allowing multiple clients to share the resources of a single physical terminal.

- Support for client-initiated `system` processes. A `system` process is a client-specified operating system process, the standard input and standard output streams of which are connected to the client transputer process.

- Multi-threading. In order to provide the optimal environment for the above two features, the server has been programmed to use multiple threads and non-blocking system calls when support for this is provided by the host operating system. Each request packet from a client transputer process is handled by a separate server thread. If the request cannot be serviced immediately (eg it might be a blocking i/o request), the server suspends the thread until it becomes runnable. Other requests can still be processed by the server. If the (multi-process) transputer client has been programmed to exploit the multi-threaded nature of the server, the server can be accessed by other processes within the client.

Refer to the **EMTS User Guide** for further information on the EMTS server extensions. Refer to Section 3 for information on the system requirements in order to support the EMTS server extensions.

## 3 System Requirements

This section specifies the system requirements for running the EMTS software.

All the EMTS programs use the Berkeley BSD socket libraries and the TCP/IP network protocol suite for handling communications over the network. This makes them portable to a wide variety of platforms. Refer to Figure 1 for an illustration of the network software model.

The current EMTS stub program supports AT/ISA/EISA bus-hosted B004-compatible transputer interface adapters. One or more B004-compatible target transputer systems (eg Inmos B008 motherboard, Transtech TMB08) can be plugged into the PC expansion slots.

The current EMTS distribution includes `eserver`, `iserver`, `estub` and `ewho` binaries for the following platforms:

- Sun3 running SunOS 4.1.* (except for `estub`)
- Sun4 running SunOS 4.1.* (except for `estub`)
- IBM PC running MS-DOS with FTP Inc's PC/TCP network software V2.05 or later
- IBM PC running MS-DOS with Sun Microsystem's PC-NFS network software V4.0 or later
- IBM PC running Unix System V Release 4 (SVR4)

The `xterm` server extension requires support from the Unix[2] operating system and requires access to an X server compatible with MIT X11 Release 4. The `system` server extension requires

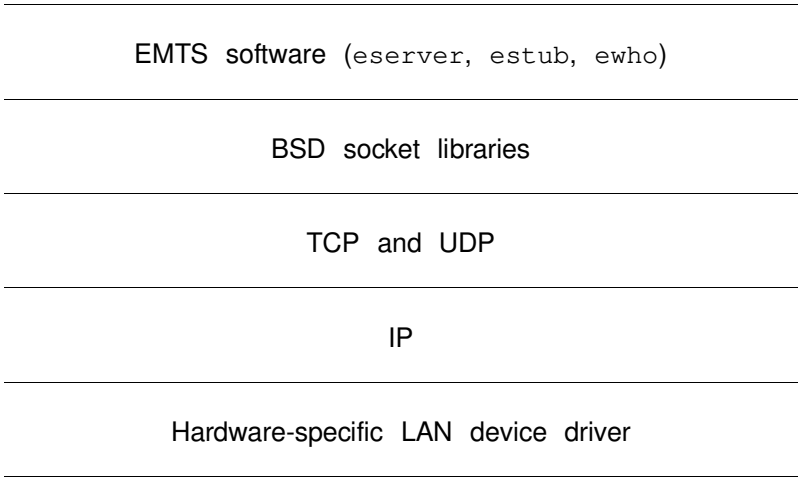


Figure 1: EMTS network programming model.

support from the Unix operating system. The `multi-threaded` server extension is Sun-specific[3] and requires support from SunOS 4.1.1 or later.

## 4 Installation

This section describes the contents of the distribution diskette, and explains how to install the EMTS software on your PC and workstation.

### 4.1 Contents of the distribution

The distribution comprises a number of 1.4 Mb 3.5 inch diskettes each containing an MS-DOS filing system. The standard distribution includes binaries for the platforms mentioned in Section 3. Copies of the code for other platforms may be available on request. Various utilities, documentation, Unix manual pages and example programs are also provided.

The directory tree under `\dos` contains the binaries for the IBM PC: the sub-directory `bin\pctcp` contains the the appropriate binaries if you are running FTP Inc's PC/TCP network software; the sub-directory `bin\pcnfs` contains the appropriate binaries if you are running Sun Microsystem's PC-NFS network software. The directory tree under `\unix` contains the binaries, manual pages and utilities for the Unix workstations: the directories `bin\sun3`, `bin\sun4` and `bin\i386` contain

---

**2** *In principle, any multi-tasking operating system will do. However, the current version of the EMTS software is based on Unix and MS-DOS.*

**3** *A platform independent version of the server is available based on the ANSA (Advanced Network Systems Architecture) distributed computing libraries. Contact your supplier for further information.*

the binaries for the sun3 architecture (SunOS), the sun4 architecture (SunOS) and the i386 architecture (SVR4) respectively. The directory `\unix\b004` contains the SVR4 driver "package" for the b004 transputer board. The directory tree under `\examples` contains example programs to illustrate use of the server extensions. The directory tree under `\doc` contains documentation in PostScript form. The directory tree under `\source` contains source code (in the C language) for the `emtslink` module and other EMTS programs.

Refer to the `README` file in the toplevel directory of the distibution for uptodate information on the contents of the EMTS distribution.

### 4.2 Installing the MS-DOS programs on an IBM PC

The MS-DOS programs are in the directories `\dos\bin\bin` and `\dos\bin\pcnfs` on your distribution diskettes. You need only install the binaries that are appropriate to your network software. To install the programs, insert the diskette into your floppy disk drive and copy the files to a suitable directory on your PC. For example, if your floppy disk is drive `a:`, your hard disk is drive `c:`, and you have decided to install the PC/TCP programs into a directory called `\bin` on your hard disk, use the command:

```
C:\>  copy  a:\dos\bin\pctcp\*.*  c:\bin
```

You may find it convenient to add the directory `c:\bin` to your `PATH` environment variable.

### 4.3 Installing the Unix programs and utilities

The remainder of the EMTS software (in directory `\unix`) needs be installed on a Unix filing system that is accessible to the Unix workstations that will need to use it. You need only install the binaries that are appropriate to your workstation architecture(s). You may also wish to copy over the examples from directory `\examples`, the documentation from directory `\doc`, and the source code from directory `\source`. Four possible ways of copying the server-related code from the MS-DOS distribution diskette to the Unix filing system are described below. If none of these are appropriate to your site, contact your supplier.

#### *4.3.1 Using the floppy disk drive in your Sun workstation*

If your Sun workstation has a 3.5 inch floppy disk drive (as do some 3/80s and most diskfull SPARCstations), you can insert the distribution diskette into your floppy disk drive and copy the software directly onto the hard disk. However, since the files on the distribution diskette are in MS-DOS format, you must first mount Sun's `pcfs` filing system (see your SunOS documentation). For example, if you have a SPARCstation running SunOS 4.1.x and you wish to copy the server-related software into `/usr/local/emts`, you would insert your diskette into the Sun floppy disk drive and issue the following commands:

```
example%  mount  /pcfs
example%  cp  -r  /pcfs/unix/*  /usr/local/emts
```

*4.3.2 Using PC-NFS*

If you have PC-NFS (either the Sun PC-NFS product or the FTP PC/TCP product) and you are using a Unix fileserver machine, you can copy the server-related software onto the fileserver using the MS-DOS `copy` or `xcopy` commands. For example, if your fileserver is mounted on the MS-DOS logical drive `e:` and you wish to copy the server-related software into `\emts` on drive `e:`, you would insert the diskette into your PC floppy drive (say `a:`) and issue the following command:

```
C:\>  xcopy  a:\unix\*.*  e:\emts  /s
```

Once the software is on your fileserver, you may wish to use Unix commands to move it to another part of the filing system.

*4.3.3 Using the FTP protocol*

If you have access to an FTP client program for the PC, you can use FTP (File Transfer Protocol) to copy the server-related software from MS-DOS to Unix. Refer to your FTP documentation for further information.

*4.3.4 Using the PC/TCP FTP program supplied with the distribution*

If you are using the FTP Software, Inc PC/TCP kernel, you can use the `ftp.exe` program supplied with the distribution together with the installation script `\dos\install\install.ftp` to copy the Unix software from MS-DOS to Unix.. For example, if you have an FTP server program available on the Unix workstation `jupiter` and you wish to copy the server-related software into the default installation directory `/usr/local/emts` on `jupiter`, you would insert the diskette into your PC floppy drive (say `a:`) and issue the following command:

```
C:\>  a:\dos\bin\pctcp\ftp  -s  jupiter  a:\dos\install\install.ftp
```

The program will prompt for the user name and password to use when connecting to the FTP server. You will need to change the ownership (using `chown`) and the access rights (using `chmod`) of the files once they have been copied to Unix. If it is necessary to use an installation directory other than `/usr/local/emts`, the `install.ftp` script will need to be edited accordingly. The source of the `ftp.exe` program is provided as `\source\ftp.c` on the distribution disk.

**4.4 Installing the SVR4 b004 driver package on an IBM PC**

If you intend to run the stub on an IBM PC running Unix System V Release 4, you need to install a b004 device driver into your Unix kernel. You are strongly advised to make a backup of your existing kernel first by typing:

```
examples%  cp  /unix  /unix.old
```

To install the driver, move to the `b004` directory and type:

```
examples%  make  install
```

To build a new kernel containing the b004 driver, type:

```
example%  make  build
```

You now need to reboot your machine.

By default, the driver package installs nine device files of the form `/dev/b004[xyz]`. The file `/dev/b004` is a synomyn for the file `/dev/b00400` and should be used to access a transputer board at (the default) i/o address 0x150. The files `/dev/b00401`, `/dev/b00402` and `/dev/b00403` should be used to access transputer boards at i/o addresses 0x200, 0x250 and 0x300 respectively. The files `/dev/b004128`, `/dev/b004129`, `/dev/b004130` and `/dev/b004131` can be used to access transputer boards at i/o addresses 0x150, 0x200, 0x250 and 0x300 respectively when the calling process is prepared to block and wait in the event that the board is in use by another process. The mapping between device file names and i/o addresses can be changed by editing the `System` and `Node` files (refer to your `idconfig` documentation).

You may wish to build and run the b004 driver test program to confirm that the driver is working correctly. To build the test program, type:

```
examples%  make  test
```

### 4.5 Installing the Unix manual pages

In order to make the Unix manual pages available to on-line users, the manual pages should be copied to the `man1` subdirectory of the standard system `man` directory tree. Alternatively, the `MANPATH` environment variable should be changed to include the EMTS `man` directory tree. Refer to your `man` documentation for further information. For example, if your site stores third-party manual pages in the directory tree with root `/usr/local/man`, you could copy the EMTS manual pages into `/usr/local/man/man1`. The manual pages are also available in PostScript format in the `\doc` directory on the distribution diskette.

## 5 Configuration

This section documents the preliminary configuration steps that are required to make your system operational. It goes on to explain how to further configure the software to support the toolsets and/or TDS3.

### 5.1 Setting the TCP/UDP port numbers

When the server starts up, it looks up the service name `emts` to discover what port it should use to connect to the stub. Likewise, when the stub starts up, it looks up the same service

name to discover on which port it should listen for incoming TCP connection requests. These port numbers must be the same to allow the server and stub to rendezvous[4].

The `ewho` program looks up the service name `ewho` to determine to which port it should send status request packets. Likewise, the stub looks up the same service name to discover on which port is should listen for incoming UDP status request packets. These port numbers must also be the same[5].

In the case of MS-DOS, the PC/TCP libraries look in the file `c:\etc\services` (by default) to discover the appropriate port number. The location of this file can be changed using the `FTP_CONFIG` environment variable (see FTP documentation) if required. The file should contain entries of the form:

```
emts        1074/tcp        #   Ethernet   Multi-user   Transputer   System
ewho        1075/udp        #   Ethernet   Multi-user   Transputer   System
```

The above port numbers have been chosen at random; you can choose any legal values, but they must be unique in your system (ie not being used by another service). The PC-NFS libraries can be configured to use the Network Information Service (NIS) instead of a file.

In the case of Unix, the lookup mechanism depends on whether or not the Network Information Service (formally known as Yellow Pages) is running. If the NIS is running, you must rebuild your NIS "services" map; see your Unix documentation. If NIS is not running, you should enter a line similar to the one above in the file `/etc/services`.

### 5.2 Toolset support on a Unix workstation

The Inmos toolsets are a collection of transputer-hosted and server-hosted tools that facilitate software development for the transputer. The *server-hosted* tools are executable binaries that run on the *server* host machine (IBM PC, Sun, etc). The *transputer-hosted* tools are bootable files that first need to be loaded onto a *transputer* before executing. The server-hosted version of a tool (when available) sometimes runs more quickly than the transputer-hosted version because it is quicker to load and because i/o operations are faster. The transputer-hosted tools are invoked by a common "driver" program that invokes the server to first boot and then serve the tool. A copy of (or a symbollic link to) the driver should exist for each tool under a different name; the driver uses the name by which it was invoked to compute which bootable file to load.

If you are using a Sun workstation and you have the Sun-hosted version of a toolset, there is very little that needs to be changed to allow you to use the toolset with the EMTS. However,

---

**4** *If the* `emts` *service lookup fails, the EMTS programs use the default port number 1073.*

**5** *If the* `ewho` *service lookup fails, the EMTS programs use the default port number 1072.*

if you have the PC-hosted version of a toolset (more likely), the changes are more considerable. The changes are described below.

*5.2.1 Sun-hosted toolsets*

If you have a Sun-hosted toolset, your server-hosted tools will run as before. In order to use the transputer-hosted tools with the EMTS, the toolset driver needs to invoke the EMTS `eserver` or EMTS `iserver` programs instead of the Inmos `iserver`. To achieve this, you can use the EMTS driver program (`\unix\bin\sun{3,4}\driver`) on the distribution diskette (source also provided in `\source\driver.c`), or you can use the Inmos driver by setting the environment variable `ISERVER` thus (assuming that `eserver` can be located via your shell search `PATH`):

```
example%  setenv ISERVER eserver
```

In addition, to run custom programs you should invoke the EMTS `eserver` or `iserver` instead of the standard Inmos `iserver`.

*5.2.2 PC-hosted toolsets*

If you have a PC-hosted toolset and you want to use it with the EMTS, you need to perform the following steps:

- Copy the transputer-hosted tools (`.btl` files) from your PC toolset `itools` directory to your Unix toolset `itools` directory (this may not be necessary if you are using PC-NFS and your Unix fileserver has access to your PC filestore).

- To support the EMTS driver program, set the environment variable `EMTS_ITOOLS` to point to the directory containing the transputer-hosted tools. **NB Do not add a trailing '\' character.**

- Make symbollic links to the driver under the names of the various tools provided by the toolset. A script `\unix\scripts\lndriver` is provided with the distribution which you may find useful for this purpose.

- Copy the contents of the PC toolset `libs` directory to your Unix toolset `libs` directory (this may not be necessary if you are using PC-NFS and your Unix fileserver has access to your PC filestore).

- Convert all text files (include files, linker indirection files, etc) in the Unix toolset `libs` directory from MS-DOS file format to Unix file format[6]. A simple script

---

**6** *MS-DOS text files use the ASCII "new-line" and "carriage-return" codes to mark the end of a line; Unix text files use the ASCII "new-line" code only.*

\unix\scripts\dos2unix is provided with the distribution for this purpose. Your Sun workstation may have a SunOS version of this utility which you may prefer to use.

- Set the ISEARCH environment variable to point to the Unix toolset libs directory.

- Set the IBOARDSIZE and IDEBUGSIZE environment variables as described in your toolset documentation.

As an example of this procedure, suppose you have the PC-hosted D7205 Occam toolset installed in c:\d7205, and you wish to use the toolset with the EMTS. Suppose you decide to install the Unix copy of the toolset in /usr/local/d7205 on your Unix workstation, which is assumed to be a Sun4 (SPARC). Suppose you have already installed the EMTS in /usr/local/emts, and that /usr/local/bin is included in your (C) shell search PATH. Following the above instructions, here is what you do:

```
example%  cd /usr/local/d7205
example%  mkdir itools
example%  <copy the contents of the PC D7205 itools directory
to your Unix D7205 itools directory using PC-NFS, pcfs, etc>
example%  setenv EMTS_ITOOLS /usr/local/d7205/itools
example%  cd /usr/local/bin
example%  cp /usr/local/emts/bin/sun4/driver /usr/local/bin
example%  /usr/local/emts/scripts/lndriver /usr/local/d7205/itools/*
example%  cd /usr/local/d7205
example%  mkdir libs
example%  <copy the contents of the PC D7205 libs directory
to your Unix D7205 libs directory using PC-NFS, pcfs, etc>
example%  cd libs
example%  /usr/local/emts/scripts/dos2unix *.lnk *.liu *.inc
example%  setenv ISEARCH /usr/local/d7205/libs/
example%  setenv IBOARDSIZE=#200000
example%  setenv IDEBUGSIZE=#200000
```

You will probably find it convenient to add the various setenv commands to your .login (C shell) or .profile (Bourne shell) files in your home directory so that the environment variables are set automatically when you first log in.

### 5.3 F Editor

If you wish to use the F editor with the EMTS, the f.btl file must be present on your Unix filing system together with a number of support files (see your toolset documentation). In particular, you must have a suitable ITERM file for the terminal or terminal emulator that you will be using for the editor, and you must set the environment variable FTERM to point to this file. A sample script \unix\scripts\f is provided with the distribution to start the F editor; you should read the comments in this file and then edit it to suit your environment.

A sample ITERM file \unix\iterms\f.itm together with its corresponding help file \unix\iterms\f.hlp are provided with the distribution. You may wish to customise these files to suit personal preferences. The sample ITERM file assumes that the terminal is either the system console (as is the case when no window system is running) or a pseudo-terminal implemented by a terminal emulator program (as is the case in a windowing environment). The terminal emulators that are supported include the SunView shelltool, the X11 xterm, and the OpenWindows shelltool programs.

If you are using the X11 xterm program and you want to use the f.itm file supplied with the distribution, you must start the program with the -sf switch to cause the function keys to generate SunView-compatible escape sequences. Please refer to the comments at the start of the f.itm file.

If you are using the Sun OpenWindows shelltool program and you want to use the f.itm file supplied with the distribution, you must have a suitable .ttyswrc file in your home directory to cause the function keys to generate SunView-compatible escape sequences. A sample .ttyswrc file \unix\scripts\ttyswrc is provided with the distribution.

### 5.4 TDS3 support on a Unix workstation[7]

The TDS3 (Transputer Development System 3) is an integrated set of *transputer-hosted* tools (except for the server) that predates the toolset. Because TDS3 is served by the iserver, it can be used with the EMTS on a Unix workstation if required.

Here is what you must do to allow the TDS3 to be used with the EMTS:

- Copy the TDS3 system files from your PC TDS3 system directory to your Unix TDS3 system directory (this may not be necessary if you are using PC-NFS and your Unix fileserver has access to your PC filestore).
- Copy the contents of the various PC TDS3 library directories to corresponding Unix TDS3 library directories (this may not be necessary if you are using PC-NFS and your Unix fileserver has access to your PC filestore).
- Copy the TDS3 tools from your PC TDS3 tools directory to your Unix TDS3 tools directory (this may not be necessary if you are using PC-NFS and your Unix fileserver has access to your PC filestore).
- Set the TDSSEARCH environment variable to point to the Unix TDS3 system directory.
- By running TDS3 on the PC, build an appropriate "toolkit" file that references the Unix TDS3 libraries and tools (see your TDS3 documentation).

---

**7** *The TDS has now been superseded by the occam toolset.*

- Make a symbollic link called `dir` from a directory in your shell's search `PATH` to the `/bin/ls` command. This is because when the TDS3 starts up, it issues the `dir` command to locate the `*.TOP` files in the current directory!

- Set the `ITERM` environment variable to point to a suitable `ITERM` file (see below).

- Set the `IBOARDSIZE` environment variable as specified by your TDS3 documentation.

As an example of this procedure, suppose you have the TDS3 installed on your PC in `c:\tds3`, and you wish to use the TDS3 with the EMTS. Suppose you decide to install the Unix copy of TDS3 in `/usr/local/tds3` on your Unix workstation. Suppose you have already installed the EMTS in `/usr/local/emts`, and that `/usr/local/bin` is included in your (C) shell search `PATH`. Following the above instructions, here is what you do:

```
example%  cd  /usr/local/tds3
example%  mkdir  system
example%  <copy  the  contents  of  the  PC  TDS3  system  directory
to  your  Unix  TDS3  system  directory  using  PC-NFS,  pcfs,  etc>
example%  mkdir  hostlibs
example%  <copy  the  contents  of  the  PC  TDS3  hostlibs  direc-
tory  to  your  Unix  TDS3  hostlibs  directory  using  PC-NFS,
pcfs,  etc>
example%  mkdir  mathlibs
example%  <copy  the  contents  of  the  PC  TDS3  mathlibs  direc-
tory  to  your  Unix  TDS3  mathlibs  directory  using  PC-NFS,
pcfs,  etc>
example%  mkdir  complibs
example%  <copy  the  contents  of  the  PC  TDS3  complibs  direc-
tory  to  your  Unix  TDS3  complibs  directory  using  PC-NFS,
pcfs,  etc>
example%  mkdir  iolibs
example%  <copy  the  contents  of  the  PC  TDS3  iolibs  directory
to  your  Unix  TDS3  iolibs  directory  using  PC-NFS,  pcfs,  etc>
example%  mkdir  tools
example%  <copy  the  contents  of  the  PC  TDS3  tools  directory
to  your  Unix  TDS3  tools  directory  using  PC-NFS,  pcfs,  etc>
example%  setenv  TDSSEARCH  /usr/local/tds3/system/
example%  <build  an  appropriate  toolkit  file  in  your  Unix
TDS3  system  directory  to  reference  the  Unix  TDS3  tools  and
libraries>
example%  cd  /usr/local/bin
example%  ln  -s  /bin/ls  dir
example%  setenv  ITERM  /usr/local/emts/iterms/tds3.itm
example%  setenv  IBOARDSIZE=#200000
```

A sample script \unix\scripts\tds3 is provided with the distribution to start the TDS3; you should read the comments in this file and then edit it to suit your environment.

A sample ITERM file \unix\iterms\tds3.itm together with its corresponding help file \unix\iterms\tds3.hlp are provided with the distribution. You may wish to customise these files to suit personal preferences. The sample ITERM file assumes that the terminal is either the system console (as is the case when no window system is running) or a pseudo-terminal implemented by a terminal emulator program (as is the case in a windowing environment). The terminal emulators that are supported include the SunView shelltool, the X11 xterm, and the OpenWindows shelltool programs.

If you are using the X11 xterm program and you want to use the tds3.itm file supplied with the distribution, you must start the program with the -sf switch to cause the function keys to generate SunView-compatible escape sequences. Please refer to the comments at the start of the tds3.itm file.

If you are using the Sun OpenWindows shelltool program and you want to use the tds3.itm file supplied with the distribution, you must have a suitable .ttyswrc file in your home directory to cause the function keys to generate SunView-compatible escape sequences. A sample .ttyswrc file \unix\scripts\ttyswrc is provided with the distribution. Furthermore, in order to be able to handle the ALT key, you should refer to the comments at the start of the tds3.itm file.

## 6 Emtslink module

The source of the emtslink module is provided in the \source directory of the distribution diskette. This module provides the standard iserver link interface. Programmers wishing to build new servers to work with the EMTS should link the emtslink module to their custom server modules. Please refer to the comments at the start of the source file for detailed programming information.