TSX-PLUS
Systems Managers Guide

s&h computer systems, inc.

TSX-PLUS
Systems Managers Guide

First printing -- December, 1980

S&H Computer Systems, Inc.
1027 17th Avenue South
Nashville, Tennessee 37212
USA
(615)-327-3670

The information in this document is subject to change without notice and should not be construed as a commitment by S & H Computer Systems Inc. S & H assumes no responsibility for any errors that may appear in this document.

NOTE: TSX-Plus, COBOL-Plus, TSX and RTSORT are proprietary products owned and developed by S&H Computer Systems, Inc., Nashville, Tennessee, USA. The use of these products is governed by a licensing agreement that prohibits the licensing or distribution of these products except by authorized dealers. Unless otherwise noted in the licensing agreement, each copy of these products may be used only with a single computer at a single site. S&H will seek legal redress for any unauthorized use of these products.

Questions regarding the licensing arrangements for these products should be addressed to S&H Computer Systems, Inc., 1027 17th Ave. South, Nashville, Tennessee 37212, (615)-327-3670.

Contents

READ THIS FIRST

The TSX-Plus distribution package you have received should contain the following items: a) a copy of the TSX-Plus Reference Manual that describes the features of TSX-Plus; b) a copy of the TSX-Plus System Manager's Guide that provides information needed by the system administrator such as how to generate a system; and c) a disk containing at least the following files:

TSX1.OBJ	Object module library for building TSX-Plus SAV file.
TSX2.OBJ	Object module library for building TSKMON.
TSGEN.MAC	Macro source file of TSX-Plus parameter module.
CCL.SAV	SAV file of DCL command processor.
LOGON.SAV	SAV file of TSX-Plus logon program.
TSAUTH.SAV	TSX-Plus account authorization
TSODT.OBJ	Object file for TSX-Plus ODT debugging program.
TSODT.REL	Relocatable copy of TSODT debugging program.
TSXPM.SAV	TSX-Plus performance monitor reporting program.
TSXLNK.COM	Command file used to link TSX-Plus.
TSXDB.SAV	Program used by us to debug TSX-Plus.

The TSX-Plus distribution disk may also contain some device handlers which have the extension ".TSX" (e.g., LP.TSX, RK.TSX).

The process of generating a TSX-Plus system is not long or difficult. If you understand what you are doing you can probably generate the system in about 15 to 30 minutes. However, before you begin the system generation process you should do two things: First, you should read the TSX-Plus Reference Manual. There are a number of features provided by TSX-Plus that are not available in standard RT-11 (deferred character echoing, virtual lines, detached jobs and device spooling, to name a few). It is necessary to understand the function of these features before you can perform a system generation. Secondly, you should find out the device status register and interrupt vector addresses of the communication equipment that will be used by TSX-Plus. Once you have done this you can proceed with the TSX-Plus system generation as described in the system manager's manual.

1. TSX-Plus System and File Access Security

TSX-Plus provides a number of system security options that allow the site manager to control access to the system by timesharing users. By selecting the appropriate combination of options the system manager can control who can log onto the system, which files or devices each user can access and can also lock users to application programs. There are five facilities that can be used to control system access:

1. Start-up command files.
2. The RUN/LOCK switch.
3. The ACCESS command.
4. Operator command privilege.
5. The LOGON and account authorization programs.

In addition to these security features, TSX-Plus also provides a use accounting facility that keeps track of the number of timesharing sessions and the total connect time that each user uses.

1.1 Start-up Command Files

When TSX-Plus is generated, the system manager may specify for each timesharing line the name of a command file that is to be executed each time the line is started. The command file name is specified by using a "CMDFIL" macro within the line definition block in TSGEN. This is explained in detail in the section on system generation. Different command files may be specified for each line and any or all lines can be generated without having start-up command files.

If a line has a start-up command file, that command file is initiated each time the line is initialized, i.e., when a user presses carriage return on an inactive line. Start-up command files have a unique characteristic compared to other command files. They cannot be aborted by typing control-C. In fact, all typing at the terminal is ignored during the execution of a start-up command file. This allows the system manager to place any desired commands in a start-up command file to be executed to completion regardless of the actions of the timesharing user.

A start-up command file can contain any keyboard command and can run one or more programs. Terminal input is completely ignored until the start-up command file is terminated or a program initiated by it requests input from the terminal. It is suggested that start-up command files be given the extension "TSX" to prevent their being tampered with by users who do not have operator command privilege (see below). Note, if "TSX" is used as the file extension, it must be specified with the file name in the CMDFIL macro as the default extension is "COM". The default device is "DK:".

When generating TSX-Plus, the "$QTSET" flag is used to control whether command files are initially listed or not. This flag has the same effect as the "SET TT (NO)QUIET" command. The listing of start-up command files is controlled by the setting of this flag. If this flag is set the startup command file will not be listed. Note, however, that the start-up command file may contain a "SET TT (NO)QUIET" command to set the listing control for later command files.

1.2 The RUN/LOCK Switch

The "R" and "RUN" commands accept a "/LOCK" switch that causes the program being run to be "locked" to the timesharing line. A locked program executes in the normal fashion, and may chain to other programs (which become locked). However, if a locked program exits or is aborted by typing control-C the line is automatically logged off. Note that one can prevent an ongoing program from being aborted by control-C by doing an .SCCA EMT or by using the TSX-Plus "D" program controlled terminal option (see the TSX-Plus Reference Manual).

In a situation in which a timesharing line is to be automatically locked to a program when the line is started, simply build a start-up command file for the line and include as the last entry in the file a "RUN/LOCK program" command.

1.3 The ACCESS Command

The ACCESS keyboard command is used to limit access to devices and files. The ACCESS command is unique in that it is only legal if executed as part of a start-up command file.

The form of the ACCESS command is:

```
ACCESS dev:file.ext/switch,dev:file.ext/switch,...,dev:file.ext
```

Up to 20 "dev:file.ext" expressions may be specified.

If no ACCESS command is executed, the timesharing user is allowed to access all devices and files on the system (with the exception of SYS and TSX-Plus files--see below). If an ACCESS command is executed, the user is restricted to accessing only the devices and files that are specified with the command.

The "dev:file.ext" expression has three items: the device name, the file name and the extension. The "*" (wildcard) character may be substituted for any or all of these three items. In this case the wildcard will allow access to any name that occurs in the wildcarded position. For example, "RK1:*.ABC" will allow access to any file on RK1 that has the extension "ABC". Consider the following ACCESS command:

```
     ACCESS RK0:*.ABC,RK0:*.BAK,RK1:*.*,LP:*.*
```

Will not accept wildcard specs. except *.*

This allows access to any files on RK0 that have the extension "ABC" or "BAK"; it also allows access to all files on RK1 and LP. Note that the LP specification is needed if the user is to be allowed to access the spooled line printer. Access privilege is needed to read, create, delete, or rename a file. A device can only be initialized (directory zeroed) if full access to the device is granted.

The ACCESS facility works by matching the user-specified device, file and extension names with those that were specified on the ACCESS command. This matching is done after any ASSIGNS of logical to physical device names are carried out.

The "/READ" switch may be specified with a device-file name to restrict access to the device-file to be read-only. For example, the following command allows full access to RK1 but read-only access to Rk0.

```
     ACCESS RK1:*.*,RK0:*.*/READ
```

The DIRECTORY command can be used to list the names of all files on a device if the user has access to <u>any</u> file on the device. Thus, if the ACCESS command specified:

```
     ACCESS RK1:TEST.FOR,RK2:*.DAT
```

the user could list the directories of RK1 and RK2, but not RK3.

A timesharing user can always access device "TT".

A timesharing user is <u>always</u> prohibited from accessing any file with the extension "SYS" or "TSX" unless the timesharing user has operator privilege (see below).

1.4 Operator Command Privilege

Certain TSX-Plus keyboard commands can only be used by users who are granted "operator command privilege" by the system manager. The commands controlled by this feature are $STOP, BOOT, and $SHUTDOWN. If a user who does not have operator privilege tries to use one of these commands, TSX-Plus displays the message "?KMON-F-You're not privileged for that command".

Operator command privilege is also required for two other actions. Files that have the extension "SYS" or "TSX" cannot be accessed unless the user has operator command privilege. Also, the TSX-Plus account authorization program (TSAUTH) cannot be run under TSX-Plus unless the user has operator command privilege.

Operator command privilege can be granted in either of two ways. If the LOGON program is used, individual project programmer numbers can be granted or denied this privilege (see the section on account authorization). If the LOGON program is not used, this privilege is specified on a line-by-line basis during TSX-Plus system generation by setting the appropriate control flags using the FLAGS macro (see the section on system generation).

1.5 Use of the LOGON facility

The TSX-Plus LOGON facility provides access security to the system by requiring users to enter a valid project programmer number and password before granting access to the system. In addition, the LOGON facility allows the system to grant different privileges to each user and provides system use accounting on a user by user basis.

To use the LOGON facility the system manager must first use the account authorization program (see section 2) to create an account authorization file. This file specifies the valid project programmer numbers, passwords and privileges. He must then generate a TSX-Plus system and specify a start-up command file to be executed for each line that is to be forced to logon. The suggested name for this start-up command file is "SY:LOGON.TSX". This command file may contain any desired keyboard commands but must end with the command:

```
R/LOCK LOGON
```

This command causes the LOGON program to be started and "locked" to the line so that the user cannot run any other program until the logon has been successfully completed. Note that the logon program (LOGON.SAV) must be present on the system device.

To prevent listing the start-up command file, the character sequence "^(" can be placed in front of the first command in the file. Thus, the logon start-up file might contain:

```
^(R/LOCK LOGON
```

R TSAUTH

2. The TSX-Plus Account Authorization Program

The TSX-Plus account authorization program is called "TSAUTH". It is used to authorize project programmer numbers for access to the system when the LOGON facility is used. It is also used to display the use accounting statistics that are collected when the LOGON facility is used.

A user must have operator command privilege to be allowed to run TSAUTH under TSX-Plus. However, TSAUTH may also be run directly under RT-11 without TSX-Plus. TSAUTH creates a file on SY named "ACCESS.TSX". Note that operator command privilege is required to access any file with the extension "TSX".

Whenever TSAUTH is started it checks to see if an account authorization file already exists. If not it prints the message:

```
Cannot open account authorization file "SY:ACCESS.TSX"
Do you want to initialize a new authorization file?
```

If you respond "YES" (or "Y") to this question it will ask you how many project programmer numbers (PPN's) you want to reserve room for in the file. Respond by entering the maximum number of PPN's that you anticipate ever needing to have authorized at any one time. As old PPN's are deauthorized file space is freed that can be used for new PPN's. Note however that the only way to enlarge the ACCESS file is to delete it and build a new larger one from scratch.

Once the authorization file is found or built TSAUTH prints an asterisk indicating it is waiting for a command. Commands are entered as a single letter followed by a space and (for most commands) a project programmer number typed with a comma separating the project number from the programmer number. The L(IST), K(ILL), U(SE) and R(ESET) commands allow a wildcard asterisk character to be used in place of the project number, programmer number or both. This allows sets of project programmer numbers to be dealt with as a group.

TSX-Plus gives no special significance to the grouping of accounts by project and programmer numbers. However, for convenience in using the TSAUTH program, it is suggested that a common project number be used for all PPN's associated with the same project or class and the programmer number be used to identify an individual.

2.1 Authorizing a project-programmer number

The "A" (Authorize) command is used to add a new PPN to the authorization file. The form of this command is:

```
A proj,prog
```

A wildcard ("*") may not be substituted for the project or programmer number. A project programmer number must be deauthorized by the use of the kill command before it can be reauthorized. Project and programmer numbers must be decimal values in the range 1 to 65535.

TSAUTH responds to the "A" command by asking a series of questions as follows:

a) Password:

Enter a 1 to 7 character alphanumeric string that is the password to be associated with the PPN.

b) Start-up file:

Enter the name (dev:file.ext) of the start-up command file to be executed whenever the user with this PPN loggs on. The default device is DK and the default extension is COM.

c) Virtual lines:

Respond with "Y" for yes or "N" for no, indicating whether the PPN is to be allowed to use the TSX-Plus virtual line facility.

d) Detached jobs:

Respond Yes/No indicating whether the PPN is to be allowed to use detached jobs.

e) Operator commands:

Respond Yes/No indicating whether the PPN has operator command privilege. Generally, this should be restricted to a small set of users.

Example:

In the following example a PPN 107,423 is authorized with the password "SECRET" and the start-up command file named "SY:SU107.TSX" is specified.

```
*A 107,423
Password:SECRET
Start-up file:SY:SU107.TSX
Virtual lines:Y
Detached jobs:N
Operator commands:N
*
```

2.2 Deauthorizing Accounts

The "K" (Kill) command is used to deauthorize project programmer numbers. The form of this command is:

```
K proj,prog
```

where a wildcard character ("*") may be substituted for the project number, programmer number or both.

Examples:

1. Deauthorize PPN 107,423

```
*K 107,423
```

2. Deauthorize all PPN's with the project number 237.

```
*K 237,*
```

2.3 Listing account status

The "L" (List) command is used to list the current authorization status of the PPN's. The form of this command is:

```
L proj,prog
```

The wildcard character may be substituted for the project and programmer numbers. The information listed includes all of the items that were specified when the account was authorized.

Example:

List the current status of PPN 107,423

```
*L 107,423
Password:SECRET
Start-up file:SY:SU107.TSX-Plus
Virtual lines:Y
Detached jobs:N
Operator commands:N
*
```

2.4 Listing Account Usage Statistics

The "U" (Usage) command can be used to display the account usage statistics which consist of the number of sessions, the connect time, and the CPU time. The form of this command is:

```
U proj,prog
```

The wildcard character may be substituted for the project number, programmer number, or both.

Example:

List the usage statistics for all PPN's with the programmer number 423.

```
*U *,423

PPN:107,423     #sessions:14    connect time 01:23:00 CPU=00:03:07.4
PPN:413,423     #sessions:5     connect time 14:02:00 CPU=01:13:02.5
PPN:21,423      #sessions:10    connect time 01:11:00 CPU=00:49:18.9
```

2.5 Creating a Charge Information File

The "C" (Charge) command causes TSAUTH to create a file of usage information. The file is named "DK:CHARGE.TSX"; it contains one record for each PPN.
The format of a charge record is as follows:

Columns	Contents
1	(blank)
2 - 6	Project number
7	(blank)
8 - 12	Programmer number
13	(blank)
14 - 18	Number of logons
19	(blank)
20 - 24	Number of minutes of connect time
25	(blank)
26 - 33	CPU time used (0.1 second units)

2.6 Resetting Account Usage Statistics

The "R" (Reset) command resets the account usage statistics (number of sessions, connect time, and CPU time) to zero for all or a selected set of accounts. The form of this command is:

```
R proj,prog
```

where the wildcard character may be substituted for the project number, the programmer number, or both.

Examples:

1. Reset all PPN's with the project number 21.

```
*R 21,*
```

2. Reset all PPN's

```
*R *,*
```

2.7 Exiting from the Account Authorization Program

The "E" (Exit) command is used to exit from the TSAUTH program to the keyboard monitor. The form of this command is:

```
E
```

3. TSX-Plus System Generation

The process of generating a TSX-Plus system tailored to the needs of a particular installation consists of 4 steps:

1. Assembling the TSGEN module with listing.
2. Editing parameters in the TSGEN module.
3. Assembling the TSGEN module.
4. Linking the TSX-Plus object modules to form the TSX and TSKMON save images.

3.1 Assembling TSGEN module

The first step in building a TSX-Plus system is to use the MACRO assembler to assemble TSGEN.MAC with a listing. The command to do this is shown below.

```
.MACRO/LIST TSGEN
```

The file "SYSMAC.SML" must be present on the system device ("SY") during the assembly. The SYSMAC.SML file is supplied by DEC with RT-11.

The TSGEN module of TSX-Plus is always supplied in source form. TSGEN contains no executable code, but rather contains the definitions of parameters and tables that are used by TSX-Plus. In building a TSX-Plus system, the RT-11 EDIT, TECO or KED program is used to set appropriate values for parameters in TSGEN. This module is then assembled and linked with the other TSX-Plus object modules. Each of the parameters found in TSGEN is described below. Remember that numeric values are assumed to be octal unless the number is terminated with a decimal point. Note when using EDIT or TECO, the TSGEN module is divided into several "pages" by form-feed characters.

3.2 Setting Parameters in TSGEN

Once a listing of TSGEN is available, save a copy of the original TSGEN.MAC file, then use EDIT or TECO to set the appropriate values of parameters for the system being generated.

3.2.1 General Parameters

parameter	meaning
SWDBLK	This is the name of the file that will be used to hold programs swapped out of core by TSX-Plus. The default name is "TSXSWP.TSX", and it is placed on the system device "SY". The first three characters of the file name may be changed to direct the swap file to some other device.

SPLBLK This is the name of the file that holds output directed to spooled devices. The file name must be supplied even if there are no spooled devices. The default file name is "TSXSPL.TSX", and it is placed on the system device. Note that it is quite feasible to place the swap and spool files on separate devices.

HIMEM This parameter is used to specify the maximum amount of memory that can be used by any job. The value is specified in terms of k-bytes. The maximum value that may be specified is 56 (Kb). The value of this parameter does not affect the size of the generated TSX-Plus system; however, it does affect the size of the TSX-Plus swap file whose size is approximately equal to ((Total number of lines)*(HIMEM + 3))*2.

DFLMEM This parameter specifies the default memory size to be allocated to a line when it logs on. Specify the value as number of k-bytes. After a line is logged on, the "MEMORY" command may be used to alter the number of bytes of memory allocated to the job.

SWAPFL This parameter controls whether TSX-Plus is allowed to swap jobs to disk if insufficient memory is available to hold all active users. The normal case (SWAPFL=1) allows TSX-Plus to do job swapping. SWAPFL can be set to 0 (zero) in special situations such as when a small number of lines are being supported on a floppy disk based system that does not have room for a swap file. If SWAPFL is set to zero the following actions occur:

1. No disk swap file is created.
2. A line will not be allowed to log on if there is insufficient free memory space to support it.
3. Each job is allocated a memory size equal to DFLMEM (default job memory size).
4. The MEMORY command cannot be used to change the job size.

DINSPC This parameter specifies the default number of characters that will be reserved for the input ring buffer for each line. This value is used for all virtual lines and for actual lines that do not have any other value specified. It must be large enough to hold an entire line of input plus any characters that are typed ahead.

DOTSPC This parameter specifies the default number of characters that will be reserved for the output ring buffer for each line. This value is used for all virtual lines and for actual lines that do not specify any other value. A running program will be suspended when its output buffer is filled.

QUAN1 This parameter specifies the length of time a job will run in a high-priority state after receiving an activation character from the terminal. Specify the value in 0.1 second units.

QUAN1A This parameter specifies the length of time a job will run in a high-priority state after being restarted following an I/O wait.

QUAN2 This is the time-slice given to compute-bound jobs. A compute-bound job is allowed to run this long if there are no high-priority tasks that need service. If there are high-priority tasks, the job is suspended after QUAN1 or QUAN1A length of time. Specify the value in 0.1 second units.

QUAN3 Time-slice given to low-priority CPU-bound jobs if no high-priority or normal-priority jobs want to run. This time-slice controls round-robin scheduling of low-priority jobs. Note: the only low-priority jobs are jobs on virtual lines that are not connected to a terminal.

OTRASZ A user's execution is suspended and the user may be swapped out of core when that user's character output buffer is filled. As the output buffer is emptied the user is reactivated when the number of characters remaining in the buffer equals OTRASZ. The idea is to get the user running again before all of the available output is exhausted. OTRASZ should be increased as the speed of terminals is increased.

MAXSEC This is the maximum number of virtual lines that a single user may own at any given time.

MAXFIL Maximum file size (number of blocks) that will be returned in response to a .ENTER request that specifies a file size of 0 blocks. This parameter does not limit the space that will be allocated to .ENTER requests that specifiy a fixed size -- it only affects .ENTER requests that specify a file size of 0.

MAXCSH The MAXCSH and NMFCSH parameters relate to the cache of file directory entries maintained by TSX-Plus. This cache is used to reducee the number of disk accesses required to do lookups on frequently accessed files. The system disk directory is always cached. Other devices are only cached if they are introduced to the system by use of the "MOUNT" command. File directory caching can have a dramatic affect on the speed of lookups of commonly used files. It does not affect the time taken to do .ENTER, .DELETE and .RENAME requests. The MAXCSH parameter is used to specify the maximum number of device units whose directories may be cached. System space required is 2 bytes per unit.

NMFCSH This parameter specifies the maximum number of file entries that can be held in the file directory cache. This number is the total number of file entries that will be cached for all users on the system (the cache is common to all users). System space required is 18 bytes per file entry.

TIMOUT This parameter is only used for lines connected to dial-up telephone equipment. It is the time between the reception of the ring signal and the user's logging on that will be allowed before the telephone connection will be dropped. Specify in 0.5 second units.

TSLICH This is the "lead-in" character that tells TSX-Plus that the following character, which is being output by the program, is to be interpreted by TSX-Plus as a special command (for example, defining a new activation character).

MXSPAC TSX-Plus allows running programs to dynamically define activation characters. (An activation character is a character that, when received, causes TSX-Plus to swap a program into memory and continue its execution-- such as carriage return.) MXSPAC specifies the maximum number of user defined activation characters that each line may define.

EDITOR This parameter specificies the default system editor. The "SET EDIT..." command can be used to select a different editor. The allowable editors are EDIT, TECO, KED and K52.

WILDFL This flag sets the system default for implicit or explicit wildcards in file names. The "SET WILD..." command can be used to alter this setting for a line. Specify 0 (zero) for explicit wildcards or 1 for implicit wildcards.

AUTHAN This specifies those devices for which device handlers are to be automatically loaded when TSX-Plus is started. All devices to be accessed by the running system must be specified in the AUTHAN device list. Note in particular that the system disk device name must be included. The only device which should not be listed is "TT". See section 3.5 for information about TSX-Plus device handlers.

3.2.2 Device Spooling Parameters

Device spooling is an optional feature of TSX-Plus. If any spooled devices are wanted, give appropriate values to the parameters of the SPOOL macro. If spooling is not wanted, specify 0 (zero) as the first parameter to the SPOOL macro. The basic spooling feature adds 1000 words to the size of TSX-Plus.

SPOOL The SPOOL macro is used to define information about devices that are to be spooled by TSX-Plus. The SPOOL macro has 7 parameters:

1. The number of devices that are to be spooled by TSX-Plus. Specify 0 (zero) if there are none.

2. The number of spool files that may be open to all users. A spool file entry is required for each spool file that is being printed, waiting to be printed, or is in the process of being generated by a running program. Each spool file entry defined here uses 10 words of memory.

3. The number of 256 word buffers that are to be used by the spooled devices. If two buffers are available for each active device, the I/O will be "double-buffered" to achieve maximum speed. However, one buffer per device is generally satisfactory and is the recommended number. If fewer buffers are available than active devices, the devices will operate in bursts and share the buffers. Space for these buffers is reserved within TSX-Plus.

4. The number of disk blocks to be allocated within the spool disk file. All spool files share this space in the common disk spool file. If the file fills up, running programs are suspended until space is freed as blocks are printed and released One bit of memory space in TSX-Plus is required for each spool file block allocated here.

5. The names of those devices that are to be spooled. Specify exactly three characters per name. The spooled devices must be non-file structured output devices such as line printers, card punches, or plotters. Note that spooled devices should also be specified in the AUTHAN (auto handler load) list.

6. Specify 1 for the sixth parameter if the default mode for the spooler is to be 'HOLD' (see SPOOL command description in the TSX-Plus reference manual). Specify 0 (zero) for 'NOHOLD' mode. In HOLD mode, spooled output will not be processed until the spool file is completely created and the I/O channel associated with the file is closed. In NOHOLD mode, a spool file may begin to be copied to the spooled device as the spool file is being created.

7. The seventh parameter specifies the number of spool blocks that TSX-Plus will back up in response to the "SPOOL BACK" command. One word of memory in TSX-Plus is allocated for each backup block for each spooled device.

Example:

The following SPOOL macro declares that there is 1 spooled device; there may be up to 10 active spool files; one 512 byte buffer is to be used for spooling I/O; the spool file is to be 500 blocks large; the spooled device is "LP"; and the SPOOL BACK command is to backup 10 blocks.

```
SPOOL 1., 10., 1., 500., <LP >,1,10.
```

3.2.3 Record Locking Parameters

If the shared file record locking feature of TSX-Plus is wanted, the three parameters MAXSF, MAXSFC, and MXLBLK must be given appropriate values. If the shared file record locking feature is not wanted, set MAXSF, MAXSFC, and MXLBLK to 0 (zero). The basic shared file record locking feature adds 475 words to the size of TSX-Plus.

MAXSF MAXSF specifies how many shared files may be open simultaneously. Note that several users accessing the same shared file count as one. Five words of memory space are allocated for each shared file declared here.

MAXSFC Maximum number of I/O channels that all users may simultaneously have open to shared files. Note that this is the total number of channels for all users not for each user. Five words of memory space in TSX-Plus are allocated for each shared file channel specified with the MAXSFC parameter.

MXLBLK Maximum number of file blocks that may be simultaneously held locked by any channel. A file block contains 512 characters. A minimum of 2 blocks must be specified. The number of words of memory space required in TSX-Plus based on this parameter is (MXLBLK*MAXSFC).

3.2.4 Message Communication Parameters

If the message communication feature is not wanted, the three parameters MAXMC, MSCHRS and MAXMSG should be set to zero. If the message communication feature is wanted, assign appropriate values to the three parameters. The basic message communication facility adds 380 words to the size of TSX-Plus.

MAXMC Maximum number of message communication channels that may be simultaneously active. A messsage channel is active if any messages are pending on it or if any users are waiting for messages to come through it. Ten words of memory space in TSX-Plus are required for each defined message channel.

MSCHRS Maximum length of messages; specify in bytes.

MAXMSG Maximum number of messages that may be simultaneously held in message queues for all channels. Note, this is the maximum number of messages that can be queued on all channels, not each channel. The number of bytes of memory space in TSX-Plus required for the message buffers is (MSCHRS*(MAXMSG+1)).

3.2.5 Performance Monitor Parameter

TSX-Plus includes a performance monitor facility that allows you to monitor the execution of an application program running under TSX-Plus and produce a histogram showing the amount of time spent in various regions of the program. The use of the performance monitor facility is completely described in the TSX-Plus reference manual.

There is one parameter in TSGEN that is associated with the performance monitor feature. This parameter, PMSIZE, specifies the number of bytes of memory to set aside for use in accumulating histogram values during a performance analysis run. Memory space equal to the size specified with PMSIZE is permanently set aside for the performance analysis buffer. If you do not intend to use the performance analysis feature, set PMSIZE to 0 (zero) to avoid tying up any memory space. The maximum value that may be given to PMSIZE is 8192.

3.2.6 Timesharing Line Definitions

Each line that is to be used as a TSX-Plus timesharing line must be declared in TSGEN. The total number of timesharing lines is first declared by setting the proper values as arguments to the TBLDEF macro. The TBLDEF macro has three arguments. The first is the number of real (physical) timesharing lines. The second argument is the number of virtual timesharing lines. The third argument is the number of job slots to allocate for the execution of detached jobs.

The memory space required in TSX-Plus for each real timesharing line is 170 words plus the space required for the input and output character ring buffers. The memory space required for each virtual line is 155 words plus the ring buffer space. Detached job slots require 155 words each; detached jobs do not have ring buffers. A typical size for ring buffers is 75 words (150 characters) for each input buffer and 150 words (300 characters) for each output ring buffer. Then the total size for each real line would be 395 words and the size of each virtual line would be 360 words. The actual space allocated for ring buffers is specified by use of the BUFSIZ macro (see below) or the setting of the DINSPC and DOTSPC parameters.

Refer to the example at the end of this section and to the examples in the supplied TSGEN module as you read the following explanation.

The actual line definitons follow the invocation of the TBLDEF macro. Each line definition is specified by creating a Line Definition Block (LDB). There must be exactly as many LDB's as there are physical lines. Virtual lines and detached jobs are not described by LDB's.

A Line Definition Block begins by calling the LINDEF macro and ends by calling the LINEND macro. Each LDB must have matching calls to LINDEF and LINEND. Other optional macros may be called between LINDEF and LINEND to specify parameters for the line.

TSX-Plus supports lines connected to DL11 and DLV11 (LSI-11) serial communication cards and lines connected to DZ11 multiplexors. TSX-Plus will support a mixture of DL11 and DZ11 dial-up and hardwired lines.

When generating a TSX-Plus system, each real line must be declared with a line definition block (LDB) that begins with a LINDEF macro call and ends with a LINEND macro call. This is true for both DL11 and DZ11 lines.

The line definition blocks for lines connected to a DZ11 multiplexor are enclosed within a Multiplexor Definition Block (MDB). An MDB begins with a MUXDEF macro call, contains the LDB's for all lines connected to the multiplexor, and ends with a MUXEND macro call.

The MUXDEF macro requires two parameters: the first is the address of the MUX receiver interrupt vector and the second is the address of the MUX Control and Status Register (CSR).

The Line Definition Block macros require different parameters depending on whether they describe DL11 type lines or DZ11 multiplexor lines. In the case of DL11 lines, two parameters are required by the LINDEF macro. The first is the memory address of the input (receiver) interrupt vector for the line. The second argument is the address of the receiver status register for the line.

A LINDEF macro for a DZ11 MUX line requires only a single argument; it is the number of the line on the DZ11 multiplexor. Note that DZ11 lines are numbered 0 to 7.

TSX-Plus will support up to four DZ11 multiplexors (but not more than 30 total lines). If there is more than one, each DZ11 must be defined using a separate multiplexor definition block.

Note that different model DL11 cards have different ranges of addresses for the status register. DL11-A and B cards generally start at 176500, while DL11-C, D, and E cards start at 175610. The addresses increase by 10(octal) per line. Note that 16-bit device addresses are specified in TSGEN. The receive interrupt locations for DL11 cards normally start at 300 and increase by 10 (octal) per line.

The receiver status register and interrupt vector addresses for all devices are normally written on a card that is attached to the top cover of the CPU drawer (not true for 11/03). This card can be seen by sliding out the CPU. If you are unsure of the addresses of your communication cards, call DEC field service and ask for assistance. The most common problem in getting started with TSX-Plus is specifying incorrect addresses for the communication cards.

The LINDEF macro also accepts a third (second for DZ11 lines) optional parameter. One terminal may be declared to be the operator's console that receives system control messages such as requests for special form mounts if spooling is used. The terminal to be the operator's console is signified by specifying "OPER" as the third argument. Only one terminal may be declared to be the operator's console.

Optional macros may be invoked between the LINDEF and LINEND calls to set parameters for a line. The available macros are listed below.

macro	meaning
FLAGS	This macro is used to set a variety of control flags for the line. The single argument to FLAGS must be the logical sum of those flags that are to be set for the line. The legal flags are listed below.

Flag Meaning when set

$SCOPE Terminal is a CRT type terminal and rubout is to echo as backspace-space-backspace.

$ECHO Echo characters to the terminal.

$START If this flag is set the line will be automatically initiated when TSX-Plus is started. If the flag is not set, the line will not be initiated until carriage-return is pressed at the terminal.

$NODET If this flag is set the line is prevented from using the DETACH keyboard command which controls detached jobs.

$TAB Do not simulate tabs by inserting spaces. Use with terminals whose hardware responds to tab characters.

$FORM Do not simulate form feed by inserting line feed characters. Use with terminals whose hardware responds to form feed characters.

$HOLD This parameter may be used with VT50 and VT52 terminals to cause the terminal to be placed in "hold screen" mode where the SCROLL key is used to control output. Generally, the TSX-Plus automatic page rolling feature that is in effect for all scope type terminals, is more convenient than VT52 hold mode. It is recommended that $HOLD not be set. Hold mode may also be used with other types of terminals. When used with terminals other than VT50's, hold mode causes output to be suspended each time a form feed character is transmitted. Output is restarted by typing ctrl-Q.

$PAGE Allows ctrl-S to mean suspend output and ctrl-Q to mean restart output. If $PAGE is not set, ctrl-S and ctrl-Q are not interpreted by TSX-Plus and are passed directly to the user's program. $PAGE is usually set on.

$LC Enables lower case input from the terminal. Note that bit 14 of the job status word must also be 1 to enable lower case input.

$NOVLN If this flag is specified, the line will not be allowed to use the TSX-Plus virtual line facilities. That is, the line will always be connected to its primary line.

$DEFER If this flag is set, "deferred" character echoing will be enabled. If the flag is not set, "immediate" character echoing will be used. See the description of the DEFER option to the SET command in the TSX-Plus reference manual for an explanation of deferred echoing. It is recommended that deferred echoing mode be used.

$QTSET If this flag is set the line will be initialized as if a "SET TT QUIET" command had been executed. This prevents the listing of command files.

$PRIV If this flag is set the line will be authorized for "operator command privilege". See earlier section of this manual for an explanation of this privilege.

$PHONE This flag should be set if the line is connected to a dial-up telephone modem. If this flag is set, TSX-Plus will perform modem control such as answering the phone when the ring signal occurs and hanging up when carrier is lost. It is important that this flag not be specified for a line unless that line is actually connected to a line with hardware modem control facilities (DL11-E, DZ-11, etc.)

The NRMFLG parameter should be set to the appropriate combination of these flags for use as default flags. If all lines require the same set of flags, simply set NRMFLG to this combination and do not use the FLAGS macro.

Note that the "SET" keyboard command may be used to alter flag settings for a line. Thus, the command "SET TT LC" would enable lower case input, and "SET TT NOSCOPE" would say that the terminal is not a CRT device.

LINPRM The LINPRM macro is used to define parameters for DZ11 MUX lines. It must be used only within LDB's for DZ11 lines. The LINPRM macro requires three arguments. The first argument is a code that specifies the line's operating baud rate. The following codes may be used:

speed code	baud rate	speed code	baud rate
0	50	10	1800
1	75	11	2000
2	110	12	2400
3	134.5	13	3600
4	150	14	4800
5	300	15	7200
6	600	16	9600
7	1200		

The second parameter specifies whether even (0) or odd (1) parity is to be generated for transmitted characters. Parity is ignored on received characters. The third parameter specifies the number of stop bits to be sent with each transmitted character. The value may be 1 or 2. Two stop bits should be used for 110 baud mechanical teletype terminals. One stop bit should be used for all other types of terminals. If a DZ11 line definition block does not contain a LINPRM macro, the last set of parameters defined for an earlier line will be used. The first DZ11 LDB must contain a LINPRM macro.

TRMTYP

The TRMTYP macro is used to declare what type of terminal will be used with the line. The valid choices are listed below:

Name	Meaning
VT100	DEC VT100 terminal
VT52	DEC VT52 terminal
LA36	DEC LA36 terminal
LA120	DEC LA120 terminal
HAZEL	Hazeltine brand terminals
ADM3A	Lear Siegler ADM3A terminal
DIABLO	Diablo-1620 terminal
QUME	Qume Sprintwriter

DELAY

The DELAY macro is used to set the number of delay (filler) characters that are required by various terminals. DELAY requires three arguments. The first argument is the number of filler characters needed. The second argument is the character after which filler characters are to be sent. The third argument is the character to send as a filler. The appropriate values for various terminals are listed below.

terminal	DELAY parameters
LA36	= none required
LA30 @ 300 baud	= 8., 15, 0
LA30 @ 150 baud	= 4., 15, 0
LA30 @ 110 baud	= 2., 15, 0
VT05 @ 2400 baud	= 4., 12, 0
VT05 @ 1200 baud	= 2., 12, 0
VT05 @ 600 baud	= 1., 12, 0
VT50	= none required
VT52	= none required

Note that the "SET TT FILLER" keyboard command may also be used to set the terminal delay requirements.

PAGE The PAGE macro is used to set the number of printable lines on a page. It requires one argument that is the number of lines. On scope type terminals, TSX-Plus pauses at the bottom of the page until the user types ctrl-Q. On non-scope terminals TSX-Plus spaces to the top of the next page when form feed is encountered. Note that the "SET TT LENGTH" keyboard command may also be used to set page size.

BUFSIZ The BUFSIZ macro is used to specify the number of characters to reserve for the line's input character ring buffer (argument 1) and the output character ring buffer (argument 2). If a BUFSIZ macro is not used in a Line Definition Block, the default sizes as specified for DINSPC and DOTSPC will be used.

CMDFIL The CMDFIL macro is used to specify the name of a start-up command file to be executed when the line is initialized. This macro has one argument that is the name of the command file (dev:file.ext).

This ends the description of macros that can be used within Line Definition Blocks.

3.2.7 Defining Start-up Files for Detached Jobs

The DETACH macro may be used to specify the names of start-up command files to be initiated as detached jobs when the TSX-Plus system is started. The use of the DETACH macro should follow the last line definition block. The DETACH macro requires one argument which is the name of the command file to be initiated as a detached job. There may be one use of the DETACH macro for each detached job slot specified with the TBLDEF macro. If there are more detached job slots defined than there are invocations of the DETACH macro, the excess job slots are left idle when the system is started and detached jobs may be started in them by use of the DETACH keyboard command.

3.2.8 Line Definition Example

The following example shows the definition of two DL11 lines and three DZ11 lines. Two virtual lines are also declared (LDB's are not used for virtual lines). Note that the exclamation mark is used to perform the logical OR (sum) operation when combining flags.

```
NRMFLG    =  $ECHO!$PAGE!$DEFER
     TBLDEF    5.,2.,1.       ;5 real, 2 virtual lines, 1 detached job
;define DL11 line #1
     LINDEF    300,175610     ;DL11-E line
     PAGE 24.
     TRMTYP    VT100
     FLAGS     NRMFLG!$SCOPE!$VT50
     BUFSIZ    120.,300.
     CMDFIL    SY:START.COM
     LINEND
;define DL11 line #2
     LINDEF    60,177560,OPER      ;console terminal (operator's terminal)
     TRMTYP    VT100
     FLAGS     NRMFLG!$PRIV
     LINEND
;define DZ11 lines
     MUXDEF    310,177620
;MUX line #0
     LINDEF    0
     LINRPM    7,1,1          ;1200 baud, odd, 1 stop
     DELAY     2.,12,0
     FLAGS     NRMFLG!$SCOPE
     CMDFIL    SY:LOGON.TSX
     LINEND
;MUX line #1
     LINDEF    1
     LINPRM    5,1,1          ;300 baud, odd, 1 stop
     TRMTYP    LA36
     CMDFIL    SY:LOGON.TSX
     LINEND
;MUX line #6   (use same LINPRM as #1)
     LINDEF    6
     LINEND
;End of MUX lines
     MUXEND
;
;Define start-up command files for detached job slots
;
     DETACH    SY:INITDJ.COM
```

3.3 Assembling the modified TSGEN module

Once the TSGEN module has been modified using EDIT or TECO to contain the desired parameter settings, it must be assembled using MACRO. The command to do this is:

```
.MACRO TSGEN
```

If you want to get a listing of the modified TSGEN (a good idea), use the command:

```
.MACRO/LIST TSGEN
```

No errors should occur during the assembly.

3.4 Linking TSX-Plus

The final stage of building TSX-Plus is to link the component parts together. A command file to this is provided on the TSX-Plus distribution disk with the name "TSXLNK.COM". The system command to execute this command file is

```
.@TSXLNK
```

This command file creates two save files: TSX.SAV and TSKMON.SAV. Both of these files must be on the system disk (SY:) before TSX-Plus can be started. Note that both TSX-Plus and TSKMON must be rebuilt if any parameters are changed in TSGEN. Warning: Do not relink TSKMON onto the system device or copy it there while running under TSX-Plus. The position of the TSKMON file on the system disk must not change while TSX-Plus is running.

3.5 Device Handlers for TSX-Plus

TSX-Plus uses standard RT-11 device handlers (with a couple of exceptions). Handlers for TSX-Plus must be assembled with the memory management option enabled (MMG$T=1) and all other options (error logging, etc.) turned off. The handlers must be linked with the extension "TSX" rather than "SYS". Note that this means that the TSX-Plus handlers can be on the same system disk as your standard RT-11 handlers without conflict. Handlers for all devices, including the system disk, must be on the system disk when TSX-Plus is started. All handlers used by TSX-Plus must be INSTALLed in RT-11.

Any handler that directly accesses the user's buffer by using PAR1 memory address region must be changed to use PAR6. (Addresses in the I/O queue entry passes to the handler by TSX-Plus are in the PAR6 virtual address range). The only DEC handlers that are affected by this are DX, DM and DD. Copies of these handlers which have been modified for use with TSX-Plus are included on the TSX-Plus distribution disk.

3.6 Starting TSX-Plus

Before starting TSX-Plus you should make sure some programs needed by it are on the system disk (SY:). The programs TSX.SAV, TSKMON.SAV and CCL.SAV are always needed to run TSX-Plus and must be present on the system device. LOGON.SAV and TSAUTH.SAV are needed if the TSX-Plus logon facility is to be used. You may wish to run TSAUTH under RT-11 to create an initial account authorization file before starting TSX-Plus. Any required start-up command files (such as LOGON.TSX) must also be on the system disk before TSX-Plus is started.

Once you have determined that all required files are on the system disk, you can start TSX-Plus by typing

```
.R TSX
```

After this is typed, time-sharing lines that were generated to be automatically starting ($START flag) should start up and print the TSX-Plus greeting message. Other lines will be initiated when carriage return is pressed at the terminal or when the phone rings for dial-up lines.

The message "CANNOT OPEN SWAP FILE" almost always means that there is not a large enough contiguous file space to contain the TSX-Plus swap file. The size of this file is a function of the number of timesharing lines and the user program space size; it can easily reach several hundred disk blocks. The solution is to do a compression operation (i.e., "SQUEEZE SY:") and then try to run TSX-Plus again.

During its initialization TSX-Plus performs a test to make sure the physical lines defined in TSGEN actually exist. It does this by trying to access the receiver status register for each line. If a trap occurs, TSX-Plus displays the message:

```
?TSX-Plus-F-Hardware trap while accessing line
        with address = xxxx
```

If this occurs you must edit in the correct line address in the TSGEN module. TSX-Plus does not check the interrupt vector addresses for the lines, so if the system dies when a line is started, check to see if its interrupt vector address is correctly specified.

When TSX-Plus is running, the system line frequency clock must be operating at all times. This is true even if only a single job is being run under TSX-Plus.

If TSX-Plus does not start properly, carefully review the parameter settings in TSGEN. Check especially the values provided for DL11 interrupt vectors and receiver status registers. Different models of DL11 cards use different addresses.

In the case where RT-11 runs successfully on a system but TSX-Plus does not, look carefully at the memory installed on the machine above 56Kb. If it is not functional or improperly configured, TSX-Plus will not run. If you are using non-DEC peripherals, check with the peripheral vendor to make sure the device can support extended memory addressing.

If all else fails, get a listing of TSGEN and contact us.

Appendix A Modification to the PATCH Program for TSX-Plus

The following patch must be made to the PATCH program if it is to be used under TSX-Plus.

Patch to version 3B PATCH.

```
    1024/      101004  240
    1034/      5000 12700
    1036/      5740 177777
```

Patch to version 4 PATCH.

```
    1024/      101006  240
    1040/      5000 12700
    1042/      5740 177777
```

With these changes to PATCH, it will run under either RT-11 or TSX-Plus. When running under TSX-Plus the /SINGLE (single character activation) switch must be used with the RUN command because of the type of character activation done by PATCH.