



s&h computer systems, inc.

TSX-Plus Version 6.01

Release Notes

These release notes describe the differences between TSX-Plus version 6.01 and version 6.0. The TSX-Plus Reference Manual and TSX-Plus System Manager's Guide incorporate the information described in earlier release notes. In case of any differences between the release notes and the manuals, the release notes take precedence.

December 5, 1985

Copyright © 1985
S&H Computer Systems, Inc.
Nashville, Tennessee USA

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.

TSX®, TSX-Plus®, COBOL-Plus®, RTSORT®, Adaptive Scheduling Algorithm, and Process Windowing are trademarks of S&H Computer Systems, Inc. DEC, RT-11, CTS-300, DIBOL, F77, VMS and PDP-11 are trademarks of Digital Equipment Corporation. DBL is a trademark of Digital Information Systems Corporation.

TSX-Plus Version 6.01 Release Notes

1. Cross-connection Function Change

A change has been made to the way the control-\ (control backslash) character causes a TT-CL cross connection to be broken. Previously, control-\ caused the cross connection to be broken without dropping the Data Terminal Ready (DTR) signal which hangs up a modem. Now control-\ does drop DTR as well as breaking the connection. If you wish to break the connection without dropping DTR, you may type control-A X (control A followed immediately by X).

2. Window print Enable/Disable

It is now possible to disable the window print function. If this is done, the control character that causes a window print operation (normally control-B) is passed through to the running program rather than being intercepted by the system. The control character is also passed through if windowing is turned off for the process. The [NO]KEYPRINT qualifier is used with the SET PRINTWINDOW command to enable (KEYPRINT) or disable (NOKEYPRINT) window printing as a result of typing the control character. The NOKEYPRINT qualifier does not affect window printing triggered by a program executing the window-print EMT. The default setting is KEYPRINT (enable window print). For example, the following command disables window printing triggered by typing a control character:

```
SET PRINTWINDOW/NOKEYPRINT
```

3. New qualifier for DISMOUNT command

A /NOWARN qualifier may now be used with the DISMOUNT command to prevent display of the warning message "Device is still mounted by other users". If used, the NOWARN qualifier must be specified immediately after the DISMOUNT keyword. For example, the following command dismounts DL1 without warning message:

```
DISMOUNT/NOWARN DL1
```

4. PRO/TSX-Plus DW handler

The TSX-Plus DW handler is compatible with RT-11 through version 5.2. If you have a later version of RT-11, you will need to insure compatibility prior to running PRO/TSX-Plus by making a DW.TSX handler from the RT-11 distribution. This can be accomplished by copying the RT-11 XM handler using the following command:

```
COPY SY:DWX.SYS SY:DW.TSX
```

5. Miscellaneous Changes

- 5.1 A job (or installed command file) with SYSPRV privilege can now perform operations which were previously only legal within start-up command files. This includes commands such as ACCESS and SET LOGOFF.
- 5.2 Device and/or file names may now be specified in TSGEN for the file used to store user-defined commands (UCL) and the IND temp file. The UCLDAT parameter controls the file spec for the UCL file; the default name is SY:TSXUCL.TSX. The INDFIL parameter controls the file spec for the IND temp file; the default name is SY:TSXIND.TSX. These file specifications can be changed either by editing TSGEN or by using TSXMOD. Note that the previous name for the IND temp file was SY:INDTMP.TSX. If you have run an earlier version of TSX-Plus, you may wish to delete the INDTMP.TSX file.
- 5.3 A change has been made to the shared-file record locking system so that if the same shared file is opened more than once by the same program, and .SAVSTATUS and .REOPEN operations are performed on the channels opened to the shared file, the channel number is used to associate the correct record locking information with the channel when it is reopened.
- 5.4 The following error message is printed if you attempt to SQUEEZE or INITIALIZE a device containing TSX system files:

Can't do this to device containing TSX system files

This message is printed if the device contains any of the following files which are being used by the system:

- | | |
|--------------------------|-----------------|
| 1. System swap file | (SY:TSXSWP.TSX) |
| 2. PLAS region swap file | (SY:TSXRSP.TSX) |
| 3. Spool file | (SY:TSXSPL.TSX) |
| 4. TSKMON.SAV | (SY:TSKMON.SAV) |
| 5. CCL.SAV | (SY:CCL.SAV) |
| 6. IND.SAV | (SY:IND.SAV) |
| 7. IND temp file | (SY:TSXIND.TSX) |

6. Corrected Problems

- 6.1 The SHOW TERMINALS command will now display the current user name if the line was generated without a NAME macro and the line is currently logged on.
- 6.2 The SET TT n START command did not check to see if the line being started was connected to a CL unit. It now prints an error message if the line is connected to a CL unit.
- 6.3 The \$NOSUB flag which could be specified within a line definition block in TSGEN did not work as documented (it had no effect at all).

TSX-Plus Version 6.01 Release Notes

- 6.4 A job would hang if a .READW EMT was executed from within a completion routine, and generalized data caching was enabled for the device from which the data was being read, and the data being read was not currently in the cache.
- 6.5 The SET LD EMPTY command failed to dismount a logical disk if the logical disk was unavailable at the time that the command was executed.
- 6.6 Syntax checking for the INSTALL command has been made more rigorous and the REMOVE keyword may be used as well as the DELETE keyword to deinstall a program or command file.
- 6.7 A security problem related to system passwords has been corrected.
- 6.8 In TSX-Plus version 6.00, if a start-up command file was not found the job could be left in a logged on state with all privileges enabled. Now, if any start-up command file is not found the job is logged off.
- 6.9 The function to transmit a break signal through the communications port or one of the quad serial line units on a Professional series computer did not work correctly.
- 6.10 The .SYNCH system service did not support job numbers greater than 31.

TSX-Plus Version 6.0

Release Notes

These release notes describe the differences between TSX-Plus version 6.0 and version 5.1C. The TSX-Plus Reference Manual and TSX-Plus System Manager's Guide incorporate the information described in earlier release notes. In case of any differences between the release notes and the manuals, the release notes take precedence.

November 18, 1985

Copyright © 1985
S&H Computer Systems, Inc.
Nashville, Tennessee USA

TSX®, TSX-Plus®, PRO/TSX-Plus™, COBOL-Plus®, PRO/COBOL-Plus™, RTSORT®, PRO/RTSORT™, Adaptive Scheduling Algorithm, and Process Windowing are trademarks of S&H Computer Systems, Inc. DEC, DIBOL, PDP, Professional 300 Series, Q-BUS, RT-11, UNIBUS, VAX, VMS and VT are trademarks of Digital Equipment Corporation. DBL is a trademark of Digital Information Systems Corporation.

CONTENTS

Support for more jobs	1
Support for more shared files	1
Support for more CL lines	1
Enhanced job privileges	2
Changes to TSAUTH program	9
INSTALL command	13
System password feature	16
Process windowing facility	17
User defined terminal keys	24
Time-sharing line / CL line cross connection	28
Change to SEND command	30
Changes to subprocesses (virtual lines) and detached jobs	30
System service call (EMT) to acquire the file context of another job	33
SUSPEND and RESUME commands	34
Support for quad serial line multiplexer on Professional	35
PRO/TSX-Plus console support	35
ENDPAGE and ENDSTRING options for CL handler	36
New reset options for CL handler	37
Terminating start-up command files	37
TSXMOD System configuration utility	38
System crash dump option	46
Control of swap file size	47
Support for named PLAS regions	47
Terminal input completion routine	48
Non-wait terminal output	49
Miscellaneous changes	50
Corrected problems	54

1. Support for more jobs

The low memory portion of TSX-Plus has been reorganized to reduce the amount of space used by the system and the amount of space required for tables for each job. As a result it is now possible to generate systems with many more jobs than before. The exact number of jobs will depend somewhat on system options but more significantly on the number of unmapped device handlers which reside in low memory. Typically, a system with no unmapped device handlers will be able to support about 40 jobs. Approximately 330 bytes of low memory space is required for each job generated into the system.

The restriction of 30 jobs which was previously in effect has been removed. The number of jobs which may be supported is now limited only by the amount of space available in low memory for job dependent tables.

Since job numbers greater than 31 are now possible, the five-bit Q.JNUM field in I/O queue elements is no longer adequate to hold the job number. Consequently a new field named Q.JOB has been added to the extended TSX-Plus I/O queue elements. This cell is a full byte wide and is located at location 30 (octal) relative to the beginning of the queue element. In order to make it possible to write device handlers which can be used with old versions of TSX-Plus as well as the new version, the job number is still placed in Q.JNUM as long as it is less than 31. If the job number is greater than or equal to 31, the value 31 is placed in Q.JNUM indicating that the job number must be obtained from the Q.JOB byte cell. Note that the job number will be 0 (zero) for I/O operations performed by the operating system itself.

2. Support for more shared files

A change has been made to the shared file record locking system to remove the 6 Kb memory size limitation on the record locking tables. With the new system there is no inherent limit on the number of shared files or shared file channels that can be gennd into the system.

3. Support for more CL lines

The maximum number of CL lines (communication lines) has been increased from 8 to 16. The units are named CL0 through CL7, and C10 through C17. That is, CL unit number 8 is named C10 rather than CL8, CL unit number 9 is C11 and so forth through CL unit number 15 which is C17. C1 device unit names are used with the SET command the same as CL units. For example, the following command associates C12 (i.e. the 10th CL unit) with line number 7:

```
SET C12/LINE=7
```


4. Enhanced job privileges

A major revision has been made to the job privilege structure in TSX-Plus. The previous system which had only two classes of users, normal and privileged, has been replaced by a set of 28 privileges which may be individually enabled or disabled on a line-by-line or user-by-user basis. A SET PROCESS command has been implemented to allow control of privileges through the use of a keyboard command.

Changes have been made to the TSAUTH and LOGON programs to allow specification of sets of privileges based on user names. The new TSAUTH and LOGON programs MUST be used with the new version of TSX-Plus.

The following list of privileges shows the privilege keyword (e.g., ALLOCATE), the privilege word (1 or 2) which contains the privilege flag bit, and the bit position corresponding to the privilege flag within the word (0 to 15).

ALLOCATE [1/14] (Allocate a device)

- o Perform ALLOCATE command or issue EMT to allocate a device.

BYPASS [1/9] (Bypass device/file access restrictions)

- o Allows access to .TSX and .SYS files.
- o Allows access to any device (that is not allocated by someone else).
- o Allows non-file-structured lookups.

DEBUG [1/13] (Use debugging facilities)

- o RUN/DEBUG a program
- o SET CTRLD DEBUG
- o SET EMT TRACE

DETACH [1/12] (Affect detached jobs)

- o Start or kill detached jobs, commands or EMT's.

GETCXT [2/7] (Get file context from another job)

- o Use of new TSX-Plus EMT to copy the file context from another job.

GROUP [2/13] (Affect jobs with same project number)

- o Perform the following operations on any job which has the same project number as the job issuing the command:
 1. Issue KILL EMT.
 2. Issue KILL command.
 3. Change another job's priority (within range allowed to changer).
 4. Suspend or resume the execution of another job.

MEMMAP [1/10] (Access memory that may be significant to system, e.g., I/O page)

- o EMT's to peek/poke, BIS/BIC into I/O page.
- o .PEEK,.POKE requests outside of RMON (low memory or I/O page)
- o RUN/IOPAGE command.
- o EMT to map to I/O page or back to RMON
- o EMT to map to physical memory.

MESSAGE [2/10] (Named message channels)

- o Use named message channels.

NFSREAD [1/2] (Non-file-structured read access)

- o Allow non-file-structured lookup on directory structured device with read-only access. (Needed to run PIP or DIR)
- o Allows use of commands: COPY, CREATE, DELETE, DIRECTORY, PRINT, RENAME, TYPE (Some options to these commands may require NFSWRITE privilege.)

NFSWRITE [1/1] (Non-file-structured access)

- o Allow non-file-structured lookup on directory structured device and reads and writes.
- o Allows use of FORMAT, INIT, SQUEEZE, COPY/DEVICE, and other DUP functions.

OPER [1/8] (Operator)

- o Set system date and time (keyboard command and EMT's).
- o Perform any SET to device handler.
- o BOOT, \$STOP, \$SHUTDOWN commands.
- o SPOOL commands (ALIGN, LOCK, FORM, SKIP, BACK, SINGLE, MULT, HOLD, NOHOLD). OPER privilege is not required for SPOOL STATUS and SPOOL DELETE commands.

PSWAPM [1/7] (Change process swap mode)

- o EMT to lock job in memory or reenale swapping
- o RUN/MEMLOCK command.

REALTIME [1/6] (Real-time EMT's)

- o .DEVICE request.
- o EMT to get exclusive system control.
- o EMT to set processor priority level.
- o EMT to attach interrupts to service or completion routines.
- o EMT to release an interrupt connection.

RLOCK [2/9] (Shared file record locking)

- o Use shared file record locking and access control.

SAME [2/12] (Affect another job with same PPN)

- o Perform the following operations on any job which has the same project and programmer number as the job issuing the command:
 1. Issue KILL EMT.
 2. Issue KILL command.
 3. Change another job's priority (within range allowed to changer).
 4. Suspend or resume the execution of another job.

SEND [1/5] (Sending messages between jobs (not named message channels))

- o SEND command or EMT to send message to another line.

SETNAME [1/4] (Change user name or password)

- o SET PROCESS/NAME="name" command.
- o EMT to change process name.
- o Change own password.

SETPRV [1/3] (Affect own privileges)

- o SET own privileges beyond those authorized.

SPFUN [1/11] (Use .SPFUN EMT)

- o Allow .SPFUN (special function) operations to directory structured devices. (Needed to initialize some disks.)

SUBPROCESS [2/11] (Use subprocesses -- virtual lines)

- o Ability to use subprocesses.

SYSGBL [2/8] (Global regions)

- o Use named global regions. This privilege is also required to use terminal display windows since they create global regions.

SYSPRV [1/0] (System privilege - System manager type operations)

- o Access .SYS and .TSX files.
- o INSTALL ADD/DELETE command.
- o SHOW INSTALL command.
- o RESET command.
- o SET CACHE, IO ABORT, CORTIM, HIPRCT, INTIOC, NUMDC, QUANx, SYSPASSWORD.
- o Use the TSAUTH program under TSX-Plus.

TERMINAL [2/15] (Terminal and CL commands)

- o Perform any SET TT to line other than your own.
- o Make permanent SET to your own line.
- o Change line speed, parity, bits-per-character.
- o Issue EMT to change line speed, parity, bits-per-character.
- o Perform any SET CL command.
- o Issue EMT to assign CL unit to a line.
- o Issue SET HOST command to cross connect a time-sharing line with a CL line.

UP1 [2/0] (User-defined privilege 1)

UP2 [2/1] (User-defined privilege 2)

UP3 [2/2] (User-defined privilege 3)

UP4 [2/3] (User-defined privilege 4)

WORLD [2/14] (Affect any job)

- o Perform the following operations on any job:
 1. Issue KILL EMT.
 2. Issue KILL command.
 3. Change another job's priority (within range allowed to changer).
 4. Suspend or resume the execution of another job.
- o Kill detached jobs started automatically by system initialization (jobs specified by DETACH command in TSGEN).

The four user-defined privileges, UP1, UP2, UP3, UP4, have no effect on system operation but are maintained by the system in the same fashion as the other privilege flags. User programs may check the setting of these privileges and base their operation on them.

At any time during the execution of a job, there are three sets of privilege flags associated with the job:

1. Authorized privileges.
2. Set privileges.
3. Current privileges.

The authorized privileges are those privileges for which the job is authorized.

The set privileges are those privileges which have been specified by use of the SET PROCESS command or the TSX-Plus EMT for setting privileges. Unless the job has SETPRV privilege, the set privileges will not exceed the authorized privileges.

The current privileges are the privileges currently in effect for the job. These privileges are selected as each program is started based on the set privileges and program dependent privilege specifications for installed programs and installed command files. A TSX-Plus EMT is available to allow running programs to change the current privileges.

The SET PROCESS keyboard command can be used to alter job privileges as well as other job characteristics. The form of this command is:

```
SET PROCESS [/PRIORITY=value] [/IDENTIFICATION=value]
           [/SUSPEND] [/RESUME]
           [/NAME="string"]
           [/PRIVILEGES=(privileges)][/AUTHORIZED]]
```

The SET PROCESS command can be used to change the set privileges for the current job (within the set of authorized privileges); to set a priority for the current job or for another job (within the range of priorities for which the current job is authorized); suspend and resume execution of a job; and to specify a new user name for the job.

The /IDENTIFICATION qualifier may be used with the /PRIORITY, /SUSPEND, and /RESUME qualifiers to specify the number of a job which is to be affected by the SET PROCESS command. If the /IDENTIFICATION qualifier is not specified, the current job is affected. The ability of a job to affect another job is controlled by the WORLD, GROUP, and SAME privileges which determine whether any job can be affected (WORLD), or only a job with the same project number (GROUP), or only a job with the same project and programmer number (SAME).

The /PRIORITY qualifier is used to specify a new job execution priority for the current job or for another job. The specified priority value must be in the range authorized for the job.

The /SUSPEND and /RESUME qualifiers can be used to suspend and resume the execution of a job. The /IDENTIFICATION qualifier should be used to specify which job is to be suspended or resumed.

The /NAME qualifier can be used to specify a new user name for the job. SETNAME privilege is required to change the job name.

The /PRIVILEGES qualifier is used to specify a list of privileges. If more than one privilege is being specified the privilege keywords are enclosed in parentheses and separated by commas. If only a single privilege keyword is specified, the parentheses may be omitted. The word "NO" may be concatenated with a privilege keyword to cause the privilege to be removed from the job. For example, the following command grants the ALLOCATE privilege and removes the SUBPROCESS privilege:

```
SET PROCESS/PRIV=(ALLOCATE,NOSUBPROCESS)
```

In addition to the privilege keywords and their NO-complements, the following special privilege keywords may be specified:

```
ALL -- All privileges
NONE -- No privileges
STANDARD -- Standard privileges (ALLOCATE, DEBUG, DETACH, SPFUN,
SEND, SETNAME, NFSREAD, NFSWRITE, SAME, SUBPROCESS, MESSAGE,
RLOCK).
```

These special keywords may be used in conjunction with other keywords to select sets of privileges. When this is done the special keyword (ALL, NONE, or STANDARD) should be specified as the first privilege followed by additional privilege keywords. For example, the following command grants all privileges except REALTIME, SETPRV, and MEMMAP:

```
SET PROC/PRIV=(ALL,NOREALTIME,NOSETPRV,NOMEMMAP)
```

The following command grants the standard privileges plus OPER privilege:

```
SET PROC/PRIV=(STANDARD,OPER)
```

The /AUTHORIZED qualifier can be used in conjunction with the /PRIVILEGE qualifier to cause the authorized privilege flags to be affected as well as the set and current privileges. If the /AUTHORIZED qualifier is not specified, only the set and current privileges are affected. The following command sets the standard privileges as the authorized, set, and current privileges for the job:

```
SET PROC/PRIV=STANDARD/AUTHORIZED
```

The SET PROCESS command can always be used to remove a privilege from the authorized, set, and current privileges for the job. Privileges can only be granted for a job if the job is authorized for the privileges or if the job has SETPRV privilege which allows the job to override the authorized privileges and set any privilege.

The SHOW PRIVILEGE keyboard command can be used to display the authorized and current privileges for the job.

The TSAUTH program has been changed to allow the specification of sets of privileges for user accounts. See the description which follows for information about the new TSAUTH program.

Existing account authorization files may be used with TSX-Plus version 6.0 without change (unless they were created by TSAUTH prior to version 5.0). Existing accounts which do not have operator privilege are granted STANDARD privilege. Existing accounts with operator privilege are granted all privileges. Virtual line and detached job privileges for existing accounts are translated into the SUBPROCESS and DETACH privileges.

When a job is initiated it is authorized for all privileges; the \$PRIV line definition sysgen flag is no longer functional. Privileges may be restricted by either of two methods: (1) the SET PROCESS/PRIV/AUTHORIZED command or (2) the LOGON program. If the LOGON program is being used, a command of the form R LOGON should appear in the start-up command file for the line. When the LOGON program executes, it will set the privileges for the job based on privileges specified for the account by the TSAUTH program. If the LOGON program is not used, the job will have full privilege unless a SET PROC/PRIV/AUTHORIZED command is placed in the start-up command file.

A TSX-Plus EMT is available to allow running programs to determine the privilege sets for the job and to change privileges. The form of the EMT is:

EMT 375

with R0 pointing to an argument block of the following form:

```
.BYTE    1,150
.BYTE    function,prvtyp
.WORD    buffer
.WORD    0
```

Where "buffer" is the address of a four word buffer which contains the privilege flags to be set or cleared or which will receive the privilege flags. Note, although only the first two words of this buffer are currently used, four words should be reserved to allow for future expansion.

"Function" indicates the type of operation being done and must be 0, 1, or 2. A value of 0 causes the privilege flags for the job to be stored into the first two words of the specified buffer. A value of 1 causes the privilege flags which are set in the buffer to be cleared in the corresponding system table. That is, this function allows the job to selectively remove privileges without affecting other privileges. A function value of 2 causes the privilege flags which are set in the buffer to be set in the corresponding system table. That is, this function allows the job to selectively set privileges. If an attempt is made to enable privileges for which the job is not authorized, and the job does not have SETPRV privilege, then only those privileges for which the job is authorized are set and an EMT error code of 1 is returned.

"Prvtyp" indicates which of the three sets of privilege flags are to be accessed by the EMT if function=0 or changed by the EMT if function=1 or 2. If prvtyp is 0, the current job privileges are selected. The current privileges for the job are reset to the set privileges when the currently executing program exits. If prvtyp is 1, the set privileges for the job are selected. If any privileges are changed, the current privileges are changed as well. If prvtyp is 2, the authorized privileges are selected. If any privileges are changed, the set and current privileges are changed as well. The job must have SETPRV privilege to set any new privileges in the authorized privilege set.

5. Changes to TSAUTH program

The TSAUTH program has been rewritten and substantially enhanced. Some of the new features include: (1) an optional new syntax for authorizing accounts which allows all parameters to be specified on a single command line; (2) a MODIFY command that allows account parameters (including the password) to be changed without having to reauthorize the account; (3) support for the new job privileges; (4) automatic assignment of project-programmer numbers.

TSAUTH commands begin with a keyword (AUTHORIZE, MODIFY, KILL, etc.) followed (in most cases) by the user name associated with the account. Commands may be continued by typing a minus sign (-) as the last character on the line to be continued and then typing the continuation of the command on the next line.

Authorizing an account

The form of the command used to authorize a new account is:

```
AUTHORIZE username [/qualifiers...]
```

where "username" is the account name which may be up to 12 characters long, and "qualifiers" include the following specifications:

/PPN=project,programmer specifies the project-programmer number to be associated with the account. The first number of the pair is the project number, the second number is the programmer number. If a PPN is specified, it must be unique. If the /PPN qualifier is not specified, TSAUTH will automatically assign a unique PPN for the account. It does this by assigning project and programmer numbers which are one larger than the current largest PPN of any authorized account. The automatically generated numbers are never smaller than 100. If you wish to specify either the project or the programmer number but allow TSAUTH to automatically assign the other number, you may specify an asterisk (*) for the number which is to be assigned by TSAUTH and a numeric value for the number you wish to specify.

/PASSWORD=string specifies the password for the account. The password string may be up to 7 characters long. If the /PASSWORD qualifier is omitted, no password will be required for the account to log on. If the /PASSWORD qualifier is specified without an equal sign or string, TSAUTH prompts for the password and accepts it without echoing it to the terminal. This is useful if you want to prevent the password from being printed on a hardcopy terminal listing.

/START=file specifies the logon start-up file specification. The file specification may be up to 15 characters long. If the /START qualifier is omitted, no start-up command file will be executed when the account logs on.

/PRIORITY=value specifies the maximum authorized execution priority. If specified, the value must be in the range 1 to 127. If this qualifier is not specified, the maximum priority for the job will be set to 50.

/PRIVILEGE=privilege-list specifies the privileges which the job is to have. If the privilege list consists of more than one keyword, the keywords must be enclosed in parentheses and separated by commas. The parentheses may be omitted if a single privilege keyword is specified. The special privilege keywords ALL, NONE, and STANDARD may be used. The standard privileges are: ALLOCATE, DEBUG, DETACH, SPFUN, SEND, SETNAME, NFSREAD, NFSWRITE, SAME, SUBPROCESS, MESSAGE, RLOCK. Initially the privileges are set to STANDARD. The privileges specified in the privilege list modify the standard privileges. So, for example, the specification /PRIVILEGE=(NOSUBPROCESS,SYSPRV) authorizes the account with the standard privileges but without the SUBPROCESS privilege and with the SYSPRV privilege. The specification PRIVILEGE=(NONE,SPFUN,NFSREAD,NFSWRITE) cancels all of the standard privileges and grants only SPFUN, NFSREAD, and NFSWRITE. If the /PRIVILEGE qualifier is omitted, the standard privileges are assigned to the account.

For example, the following command authorizes an account with the user name "ACCT-MANAGER", password "SECRET", standard privileges plus SYSPRV, and a start-up command file named "DLO:MSTR.TSX". The default maximum execution priority (50) is used and TSAUTH automatically assigns a project-programmer number.

```
AUTHORIZE ACCT-MANAGER/PASSWORD=SECRET/PRIV=SYSPRV-
/START=DLO:MSTR.TSX
```

If the AUTHORIZE command is used without specifying any qualifiers, TSAUTH will prompt for each qualifier; this operates in a very similar fashion to the AUTHORIZE command of the old version of TSAUTH. When operating in this mode, if you press return without entering anything in response to the PPN (project programmer number) prompt, TSAUTH automatically generates and assigns a project programmer number.

Modifying an account

The form of the command used to modify parameters for an account is:

```
MODIFY {username | project,programmer}/qualifiers...
```

Either a user name or a project programmer number may be specified with the MODIFY command. If a project programmer number is specified, a wildcard character (*) may be specified for either or both of the numbers causing the modification to be applied to multiple accounts. For example, the following command grants SUBPROCESS privilege to all accounts with project number 43:

```
MODIFY 43,*/PRIV=SUBPROCESS
```

The qualifiers which may be used with the MODIFY command are: /PASSWORD, /PRIVILEGE, /PRIORITY, and /START. These qualifiers have the same form as specified with the AUTHORIZE command. The /PRIVILEGE qualifier adds or removes privileges relative to the privileges the account has at the time that the MODIFY command is executed, rather than adding or removing privileges from the standard set as is the case with the AUTHORIZE command. An account must be reauthorized to change either the user name or the project programmer number.

For example, the following command modifies the account with the user name MANAGER to add BYPASS privilege and change the password to SPIRIT:

```
MODIFY MANAGER/PRIV=BYPASS/PASSWORD=SPIRIT
```

As with the AUTHORIZE command, if the /PASSWORD qualifier is specified without an equal sign and string, TSAUTH will prompt for the password and accept it without echo.

Deauthorizing an account

The KILL command is used to deauthorize an account. The form of this command is:

```
KILL {username | project,programmer}
```

Either a user name or a project programmer number may be specified. A wildcard character (*) may be substituted for either the project number, the programmer number, or both.

Listing information about an account

The LIST command is used to list information about an account. The form of this command is:

```
LIST {username | project,programmer}
```

Either a user name or a project programmer number may be specified. A wildcard character (*) may be substituted for either the project number, the programmer number, or both.

Listing account usage statistics

The USAGE command is used to list the account usage statistics which consist of the number of sessions, the connect time, and the CPU time. The form of this command is:

```
USAGE {username | project,programmer}
```

Either a user name or a project programmer number may be specified. A wildcard character (*) may be substituted for either the project number, the programmer number, or both.

Creating a charge information file

The CHARGE command causes TSAUTH to create a file of usage information. The file is named "DK:CHARGE.TSX"; it contains one record for each account. The form of the CHARGE file is the same as for previous versions of TSX-Plus. The form of the CHARGE command is:

CHARGE

Resetting account usage statistics

The 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 the command is:

RESET {username | project,programmer}

Either a user name or a project programmer number may be specified. A wildcard character (*) may be substituted for either the project number, the programmer number, or both.

Exiting from the TSAUTH program

The EXIT command (or control-C) is used to exit from the TSAUTH program. The form of the EXIT command is:

EXIT

Use of indirect files within commands

A portion of a command line may be drawn from an external file by typing @file at the position in the command where the contents of the file are to be inserted into the command line. The default extension for these files is TSX. For example, the following command uses parameters in a file named STUDNT as part of an AUTHORIZE command:

AUTHORIZE JONES @STUDNT

The following command draws the privilege list from a file named CLERK and in addition negates the SUBPROCESS privilege:

AUTHORIZE FRANK/PRIV=(@CLERK,NOSUBPROCESS)

More than one indirect file reference may occur within a command but the indirect files may not be nested.

Existing account authorization files may be used with TSX-Plus version 6.0 without change (unless they were created by TSAUTH prior to version 5.0). Existing accounts which do not have operator privilege are granted STANDARD privilege. Existing accounts with operator privilege are granted all privileges. Virtual line and detached job privileges for existing accounts are translated into the SUBPROCESS and DETACH privileges.

6. INSTALL command

Programs and command files may now be "installed" in the system to allow special attributes and privileges to be invoked when the program or command file is run. The form of the INSTALL command used to add a program or command file to the install table is:

```
INSTALL ADD program[/attribute...][/PRIV=(privileges)]
```

Where program is the file specification for the program or command file that is being installed, attribute is one or more of the attributes listed below, and privileges is a list of privilege keywords which specifies which privileges are to be temporarily granted or denied while the program is executing. If a program is already in the install table, the new specifications replace the existing ones.

The device name specified for the program (or DK by default) is translated to a physical device name at the time the INSTALL command is executed. Similarly the physical device name for a program being started is used when the install table is searched at program start-up time. Installed programs must reside on physical disks, and may not be located on logical disks unless the wildcard character (see below) is used for the device name. The default extension for program names is .SAV, so .COM (or some other extension) must be explicitly specified if a command file is being installed.

A wildcard (*) may be used as the device name for an installed program or command file. In this case, a program with a matching name which is run from any device (including a logical disk) will match the install table entry. For example, the following command installs KED with the proper attributes to allow it to be run from any device:

```
INSTALL ADD *:KED/SINGLECHAR/NOWAIT/HIGH
```

Wildcard device names should not be used with programs that are granted special privileges because any user can create a program with the same name on any disk and acquire the privileges specified for the installed program.

The INSTALL command does not open the file of the program being installed and no error will occur if installed programs are not actually present on the system.

The following programs are automatically installed when TSX-Plus is started: IND, KED, KEX, K52, LOGON, PATCH, SETUP, SYSMON, TECO, TRANSF, TSAUTH, VTCOM, IND, DUP, TSXUCL. The INSTALL command can be used to alter the standard installed programs. These programs should be located on the disk from which the system was booted. If these programs are moved to a disk other than the system disk, INSTALL commands must be used to change the device specifications for these programs. Because SYSMON is installed, it no longer requires system privilege.

One entry is reserved in the install table for each of the programs that is automatically installed by TSX-Plus. Additional entries for installed programs are specified with the TSGEN parameter NUIP.

The form of the INSTALL command used to remove a program from the install table is:

INSTALL DELETE program

where program is the file specification for the program or command file whose entry is to be removed.

The SHOW INSTALL keyboard command may be used to list information about all installed programs.

SYSPRV privilege is required to use the INSTALL ADD/DELETE command or the SHOW INSTALL command.

The following attributes may be specified for installed programs and command files:

BYPASN (Bypass logical assignments)

Bypass all logical device assignments. DK and SY will be directed to the system disk (disk from which RT-11 was booted).

DUP (DUP program attribute)

Special flag for DUP program.

HIGH (High-efficiency mode)

High-efficiency terminal mode is selected.

IND (IND program attribute)

Special flag for IND program.

IOPAGE (I/O page access)

I/O page is mapped into PAR 7 of program virtual address space.

LOCK (Lock program to job)

Program is "locked" to job so that job is logged off if program exits.

MEMLOCK (Lock program in memory)

Program is locked in low memory while it is running.

NONINTERACTIVE (non-interactive execution)

Program is run with non-interactive execution priority.

NOWAIT (Non-wait TTY input)

Program is allowed to execute non-wait .TTYIN operations.

SCCA (Suppress Control-C Abort)

The execution of the program or the command file cannot be aborted by typing two control-C characters. It is recommended that a SET ERROR NONE command be placed in command files that must run to completion.

SETUP (SETUP program attribute)

Special flag for SETUP program.

SINGLECHAR (single-character activation)

Program is enabled to perform single-character activation.

TRANSPARENT (transparent terminal output)

Terminal output is processed in "transparent" mode where control characters such as the TSX-Plus "leadin" character are not interpreted by the system.

TSXUCL (TSXUCL program attribute)

Special flag for TSXUCL program.

Attributes and privileges specified for installed command files take effect when the command file is started (either by typing @file or IND file) and remain in effect until the end of the command file, or until the command file is aborted, at which time all attributes are cleared and privileges are reset to the set privileges. Attributes and privileges for installed command files are not acquired in one case: when a command file is started from within another command file that is being executed by IND. Attributes and privileges specified for a command file are applied to all programs started within the command file. Attributes and privileges are pushed and popped as nested command files are called. Attributes and privileges are "inherited" by more deeply nested command files. Note that the SCCA attribute can be specified for installed command files to prevent control-C from aborting them while they are executing.

Attributes and privileges specified for installed programs take effect when the program is started (either by use of a R or RUN command or by chaining to the program) and remain in effect until the program exits or chains to another program at which time the program attributes and privileges are reset to those for the current command file or to the set privileges if a command file is not executing.

The /PRIVILEGE qualifier may be used to temporarily grant or deny any privilege including privileges for which the job is not authorized. For

example, the following INSTALL command installs a program named CONTRL on DL1 which is to have PAR 7 mapped to the I/O page while the program is running. MEMMAP privilege is granted while the program is running but DEBUG privilege is denied to prevent a user from triggering a breakpoint while the program is running with access to the I/O page:

```
INSTALL ADD DL1:CONTRL/IOPAGE/PRIV=(MEMMAP,NODEBUG)
```

Note in this example, MEMMAP privilege is required to run this program since the /IOPAGE attribute was specified. If the MEMMAP privilege had not been specified when the program was installed, only those users who are authorized for MEMMAP privilege would be able to run this program. By installing the program with MEMMAP privilege, users who are not authorized for MEMMAP privilege are temporarily granted the privilege to enable them to run the program.

7. System password feature

A new optional system password facility has been added to provide additional security for dial-up lines. System password checking can be enabled on a line-by-line basis. If system password checking is enable for a line, the user is prompted with an exclamation point (!) when the line is initiated (i.e., when carriage return is pressed). After the correct password is entered, the normal logon sequence begins which prints the system greeting message and runs the LOGON program. The idea is to force the user to provide a password before displaying the logon greeting which identifies the site and the nature of the system.

To enable system password checking, specify the system password using the SYSPS macro in TSGEN, and include the \$SYSPS flag with the FLAGS macro within the line definition blocks for all lines that are to perform system password checking. The system password may be up to 20 characters long and may contain spaces. The same system password is used for all lines that perform password checking.

When a line is initiated which performs system password checking, the system does autobaud speed selection (if required) and then prints an exclamation point as the prompt for the system password. The password is not echoed as it is entered. Terminate the password by pressing carriage return. If the password is correctly entered, the system will print the standard TSX-Plus greeting message and perform the normal line initiation sequence. If an incorrect password is entered, no error message is printed but the exclamation point is redisplayed. If the password is entered incorrectly a second time, the line is hung up. The password must be entered within 30 seconds after the prompt is printed or the system will hang up.

A keyboard command of the form:

```
SET SYSPASSWORD password
```

can be used to change the password for the running system. SYSPRV privilege is required to use this command; the change only remains in effect until the system is rebooted. The TSXMOD program can be used to change the system password in the TSX.SAV file.

The system password can be displayed by use of the following command:

```
SHOW SYSPASSWORD
```

SYSPRV privilege is required to use this command.

System password checking may be enabled or disabled for individual lines by use of the following command:

```
SET TERMINAL n [NO]SYSPASSWORD
```

where n is the terminal line number. TERMINAL privilege is required to issue this command. The TSXMOD program can also be used to set or reset this flag for lines.

8. Process windowing (tm) facility

TSX-Plus now offers a full-screen process windowing system that allows the system to remember the contents and status of the terminal display and to redisplay windows as you switch between subprocesses or on demand by programs.

The process windowing facility also provides a "print window" function which allows you to print the contents of a window on a printer by typing a control character. Effectively, this provides a print screen function for each terminal without requiring a printer to be attached to every terminal.

The maximum number of windows which may be remembered by the system is specified by the MAXWIN sysgen parameter. If this parameter is set to 0 (zero) the window management system overlay is not loaded into memory, saving approximately 6 Kb of extended memory space.

In addition to this memory space which is allocated when the system starts running, each window creates a non-swappable global memory region whose size (in bytes) is equal to $(24+48 \times (\text{number of columns per line}))$. These regions are created when a window is created and deleted when the window is deleted.

Each job may have up to 26 windows active at one time. A window is identified by an ID number in the range 1 to 26. Two jobs may have windows with the same ID number without conflict.

There are system service calls (EMT's) described later which allow a program to create windows, delete windows, and select the current window for the job. However, the simplest way to use a window by use of the SET WINDOW keyboard command. The form of this command is:

```
SET WINDOW [/ON] [/OFF] [/COLUMNS=n] [/DARK] [/LIGHT]
          [/WIDE] [/NARROW] [/SCROLL[=n]]
```

This command manipulates window number 1.

The SET WINDOW ON command (ON can be omitted) causes the system to create window number 1 and select it as the current window for the job. If window 1 already exists, it is deleted and a new window 1 is created. SYSGBL privilege is required to use windows since they create global memory regions. Also, windows are only supported with VT100, VT200 and VT52 terminals. Color support is not provided. The options which may be specified are listed below:

/COLUMNS=n specifies the number of columns per line which the window can hold. The value should not exceed 132.

/LIGHT specifies that the window is to be in reverse video (i.e., dark characters on light background).

/DARK specifies that the window is to be in normal (not reverse) video (i.e., light characters on dark background).

/WIDE specifies the window is to hold 132 columns (the COLUMNS=132 option is implied) and causes the terminal to be placed into 132 column mode.

/NARROW specifies the window is to hold 80 columns (the COLUMNS=80 option is implied) and causes the terminal to be placed into 80 column mode.

/SCROLL[=n] specifies the maximum number of lines which are allowed to scroll off the window while you are disconnected from it by switching to a different process. If you specify 0 (zero), no lines are allowed to be scrolled. If you specify the SCROLL qualifier without a following equal sign and value, an unlimited number of lines are allowed to scroll off the screen. The job writing to the window will be suspended, and the bell will ring at your terminal, if more than the specified number of lines are attempted to be scrolled.

The default options are ON, COLUMNS=80, DARK, SCROLL=16. Thus you can turn on window 1 with 80 column mode by simply typing:

```
SET WINDOW
```

The SET WINDOW OFF command deletes all windows for the job.

Note, be sure the terminal type (VT52, VT100, or VT200) declared to TSX-Plus either during the system generation or by use of the SET TERMINAL command, matches the actual terminal type before turning on windows.

When windowing is turned on, the system monitors all characters sent to the terminal and maintains an updated screen image in memory. Terminal attributes such as line width, reverse/normal video, application keypad mode, etc. are saved along with line attributes (double wide, double high), and character attributes. The attributes retained for each character consist of blinking, bold, underlined, reverse video, and character set information (ascii, U.K. national, DEC supplemental, or graphics).

When a window is created it is set to all blanks. Each time a window is selected as the current window, the screen is cleared and the current contents of the window is displayed.

Any time you switch to a new subprocess and a window is active for the primary process, a window is automatically created for the subprocess as if a SET WINDOW command had been issued. The window width (80 or 132 columns), light or dark video attribute, and maximum allowed scroll lines for the subprocess window are copied from the existing window definition for the primary process. Thus once windowing has been enabled, switching between subprocesses will cause the screen to be restored to the state it was in when you left the subprocess you are switching back to. The SCROLL option on the SET WINDOW command can be used to control how many lines are allowed to scroll off the screen while you are disconnected from the process by switching to a different subprocess.

Display windows remember all characters sent to the screen by either running programs or input character echoing. However, messages sent to the terminal by use of the SEND command (or equivalent EMT) are not processed by the message manager. Consequently, if a window is redrawn received messages will not be shown.

You can cause the window for the current process to be redrawn without switching to a different process by typing control-W followed by the number that selects your current primary or subprocess. For example, if you are connected to your primary process and wish to have the current window redrawn, type control-W 0.

You can cause the contents of the currently displayed window to be written to a printer by typing control-B (a different control character can be used by changing the PWCH parameter in TSGEN). Before you can use this function, you must specify the printer to which the window is to be written. Specify the printer by use of a SET PRINTWINDOW command of the following form:

```
SET PRINTWINDOW [/DEVICE=ddn] [/TYPE=type]
                [/LETTER] [/DRAFT] [/[NO]BELL] [/[NO]WIDTH]
```

/DEVICE=ddn specifies the name of the printer device. This may be a printer name such as LP or LS, a CL device name such as CL1, or the name of any other device to which the print window output is to be directed. If a logical device name is specified, it is translated to the corresponding physical device name at the time that the SET command is issued. The printer device will usually be spooled but this is not necessary. It is possible to specify a disk type device (e.g., DK) as the output device. In this case, the window contents are written to a file with a name of the form "WINjji" where "jj" is the job number and "i" is a letter which is determined by the window ID.

/TYPE=type specifies the type of the printer. The following type names are recognized: LA36, LA120, LA100, LA12, LA50, LA210, LQP02, LQP03, and LNO3. If some other printer is used, specify FOREIGN as the printer type. The printer type specification determines the type of control characters which will be sent to perform various function. The default printer type is FOREIGN.

/LETTER specifies that letter quality print mode is to be used on printers such as LA100's. This qualifier is ignored if the printer does not support letter-quality mode.

/DRAFT specifies that draft quality print mode is to be used. This is the default mode.

/BELL specifies that the bell is to be rung at the terminal that is requesting a printwindow when the WINPRT program has successfully captured the window image. When the bell is rung, a "snapshot" of the window contents has been taken so new data may be displayed without affecting the window image that will be printed. /NOBELL is the default mode.

/WIDTH specifies that control characters are to be sent to the printer to control character width spacing. This only applies to DEC terminals that support character width settings. /NOWIDTH suppresses character width control. /WIDTH is the default mode.

Note that information provided by the SET PRINTWINDOW command applies only to the job issuing the command. Thus different jobs can direct their print window output to different print devices.

It is possible to define a terminal function key to generate control-B to produce a window print; however, this is less powerful than actually typing control-B because the operation only can be used at a time when the single line editor is accepting input -- not at a time when a program is executing. The following key definition defines PF2 to print the window:

```
DEFINE KEY/NOTERMINATE PF2 "^B"
```

If this command was placed in a command file, the defined string should be "`^B`".

The WINPRT program must be executing to process a print-window request. (The WINPRT.SAV program is supplied with the TSX-Plus distribution). This program is ordinarily run as a detached job. A command file named WINPRT.TSX is provided which may be specified with a DETACH command in TSGEN to cause WINPRT to be started as a detached job when TSX-Plus is initiated.

Each time a print-window request is made, a completion routine is executed in WINPRT. This completion routine accesses the current window contents for the job issuing the print request and saves this information in a memory area that is part of the WINPRT job. If multiple print requests occur rapidly, WINPRT captures the window data and queues the print requests for processing in the order in which they occurred.

The number of print requests which can be queued by WINPRT is determined by the amount of memory space available to the WINPRT job. The space required for the program itself and one print request is 11Kb. Each additional print request requires an additional 6.4Kb. Thus by using a MEMORY command in the command file that starts the WINPRT detached job or by use of the SETSIZ program, you can control the amount of memory used by WINPRT and the number of requests it can queue.

WINPRT must execute with GETCXT privilege. It must also have access to all of the devices to which print window requests will be directed.

If you have problems getting the print window function working, check the following things:

1. Is windowing turned on (SET WINDOW)?
2. Are you using the correct control character to request the window to be printed? Control-B is the default character but it may have been changed by use of the PWCH sysgen parameter.
3. Is the WINPRT program running?
4. Does the WINPRT job have GETCXT privilege and can it access the print device you specified with the SET PRINTWINDOW/DEVICE=device command?

If you still have problems, try running WINPRT from a time-sharing line rather than as a detached job. It will print error messages if any error occurs.

The system service call (EMT) used to create a window has the form

EMT 375

with R0 pointing to an argument block of the following form:

```
.BYTE    0,161
.BYTE    window_id,perm_flag
.BYTE    window_width,max_scroll
.BYTE    copy_id,copy_job
.WORD    0
```

Where window_id is a window identification number in the range 1 to 26 which identifies the window for this job. Different jobs can have windows with the same ID number without conflict.

Perm_flag should be either 0 or 1. If it is 0 (zero), the window is a temporary window and will be automatically deleted when the program exits to the keyboard monitor (other than doing a chain). If perm_flag is 1, the window is a permanent window and is only deleted when explicitly requested (by the delete-window EMT, or by SET WINDOW OFF) or when the job logs off.

Window_width is the width of the window in columns and should not exceed 132.

Max_scroll is the maximum number of lines which are allowed to scroll off the window during a time when the window is disconnected from the terminal because a different subprocess has been selected. Specify 0 to disallow any scrolling; specify 255 to allow unlimited scrolling.

The copy_id and copy_job parameters in the EMT argument block can be used to cause the system to copy certain window characteristics from another window as it is initializing the new window. If copy_id and copy_job are both zero, no information is copied from another window and the supplied values (or default values) are used for the new window. If copy_id is non-zero, it is used as the id of the window from which the information is to be copied. If copy_job is non-zero it is the number of the job that owns the window from which the information is copied. If copy_job is zero, the current job is assumed. The following characteristics are copied:

1. The number of columns per line.
2. The maximum allowed number of lines to be scrolled while switched to a different process.
3. 80 or 132 column mode.
4. Light or Dark (normal or reverse video) mode.

The global memory regions created for windows have names of the form "WINjj*i*" where "jj" is the job number and "i" is a letter which is determined by the window ID.

The following error codes may be returned by this EMT:

- 0 Window management not included in system generation or invalid EMT argument block.
- 1 Maximum allowed number of windows are already in use. Increase value of MAXWIN sysgen parameter.
- 2 Unable to create global memory region for window. Job may not have SYSGBL privilege or you may need to increase the value of the NGR sysgen parameter, or there may be insufficient memory space available.

Creating a window allocates a window control block and a global memory region for the window. The window contents are set to all blank.

The EMT argument block used to select a window as the current window has the form:

```
.BYTE    1,161
.BYTE    window_id,0
.WORD    0
```

where window_id is the window identification number as specified when the window was created. When this EMT is executed, the terminal screen is cleared and the selected window is drawn on the screen. A window id value of 0 (zero) has a special meaning: it causes windowing to be disabled for the job but the contents of all existing windows are retained and may be reselected later.

The following error codes can be returned by this EMT:

- 0 Window support not generated into system.
- 3 Unable to locate window with specified window_id.

The EMT argument block used to delete a window has the following form:

```
.BYTE    2,161
.BYTE    window_id,0
```

where window_id is the identification number of the window to be deleted. There are two special window id values which may be used with this EMT: a value of 0 (zero) causes all temporary windows for the job to be deleted; a value of 255 (decimal) causes all temporary and permanent windows for the job to be deleted. The global memory region used by a window is freed when the window is deleted.

Window processing may be suspended to allow characters to be passed to the terminal which are not processed by the window management system. The EMT argument block used to suspend window processing has the following form:

```
.BYTE      3,161
```

The following EMT argument block can be used to resume window processing after it has been suspended:

```
.BYTE      4,161
```

The suspend/resume functions affect the currently selected window. If the job does not have an active window, these EMT's perform no operation.

The following EMT argument block can be used to cause the contents of a window to be printed:

```
.BYTE      5,161
.BYTE      window_id,0
```

where "window_id" is the window id number.

The following error codes can be returned:

Code	Meaning
-----	-----
0	Windowing not available
3	Invalid window ID
4	WINPRT program not running

9. User defined terminal keys

It is now possible to specify character strings which will be substituted for a character when that character is typed. Most commonly this feature is used to associate frequently typed commands with some of the terminal function keys, but it may be used to associate a character string with any key.

The DEFINE KEY keyboard command is used to specify a key substitution. The form of this command is:

```
DEFINE KEY[/qualifiers] key "string"
```

where key is the name of the key that is associated with the string. For example, the following command associates the SHOW TIME command with the PF2 key:

```
DEFINE KEY PF2 "SHOW TIME"
```

If the associated string contains no leading or trailing spaces, no at-signs (@), and no control characters, the quotation marks may be omitted. Thus the command may be specified as:

```
DEFINE KEY PF2 SHOW TIME
```

The substitution string may be up to 64 bytes long.

If the key being defined is a control character, enclose the key in quotation marks and place a caret (^) in front of the letter that is to be converted to a control character. For example, the following command causes control-T to display the current time of day:

```
DEFINE KEY/LETTER/NOECHO "^T" SHOW TIME
```

A key definition is removed by using a DEFINE KEY command with no substitution string. Thus the definition for PF2 can be removed by use of the following command:

```
DEFINE KEY PF2
```

The following key names may be used:

Key name	Actual key
PF1,PF2,PF3,PF4	PF1,PF2,PF3,PF4
KPO,KP1,...,KP9	digits 0 to 9 on numeric keypad
PERIOD	period key (.) on numeric keypad
COMMA	comma key (,) on numeric keypad
MINUS	minus key (-) on numeric keypad
ENTER	ENTER key on numeric keypad
UP	up-arrow key
DOWN	down-arrow key
LEFT	left-arrow key
RIGHT	right-arrow key
FIND	"Find" key on VT200
INSERT	"Insert here" key on VT200
REMOVE	"Remove" key on VT200
SELECT	"Select" key on VT200
PREV	"Prev screen" key on VT200
NEXT	"Next screen" key on VT200
HELP	"Help" key on VT200
DO	"Do" key on VT200
F6,F7,...,F20	Function keys on top row of VT200

The key substitutions are performed by the single line editor; hence, substitutions are only done on terminal input processed by the single line editor. The following command can be used to enable/disable key substitutions:

```
SET SL [NO]SUBSTITUTE
```

The default mode is SUBSTITUTE.

The single line editor must be in KED mode (SET SL KED) if you define substitutions for any of the keys of the numeric keypad other than the PF keys.

You may specify a key definition for a key which is normally used to perform some single line editor function (e.g., up arrow). When this is done, the key definition takes precedence over the normal function.

The following qualifiers can be used with the DEFINE KEY command:

/[NO]ECHO determines whether the substitution string will be printed at the terminal at the time the substitution takes place. The default mode is /ECHO.

/[NO]TERMINATE determines whether a carriage-return and line-feed is appended to the end of the substitution string when it is substituted. Use /NOTERMINATE with a definition that specifies a part of a command. You may not specify /NOECHO and /NOTERMINATE together. The default mode is /TERMINATE. For example, the following definition associates the word HELP followed by one space, without termination, for the PF2 key:

```
DEFINE KEY/NOTERMINATE PF2 "HELP "
```

/[NO]GOLD determines whether the specified key must be prefixed by the "gold" (PF1) key. If /GOLD is specified then the substitution only takes place if you type the sequence PF1 followed by the specified key. If /NOGOLD is specified (the default case) then the substitution takes place when the specified key is pressed without being prefixed by the PF1 key. For example, the following two definitions associate the SHOW TIME with the PF2 key in the case where it is not prefixed by the PF1 key, and the SHOW DATE command when PF2 is prefixed by PF1:

```
DEFINE KEY PF2 SHOW TIME  
DEFINE KEY/GOLD PF2 SHOW DATE
```

/LETTER specifies that the key being defined is not a function key, but is a key such as a letter, digit, or punctuation mark which is part of the typewriter portion of the keyboard. If /LETTER is specified, then key must be a single letter from the non-function-key portion of the keyboard. The /GOLD qualifier may be used in conjunction with /LETTER to cause the substitution to only take place if the specified letter is prefixed by the PF1 key. The default mode is /NOLETTER. For example, the following

commands cause an at sign (@) to be substituted for the accent mark (`) and the SHOW JOBS command to be substituted for the gold accent character:

```
DEFINE KEY/LETTER/NOTERMINATE ` "@"
DEFINE KEY/LETTER/GOLD ` SHOW JOBS
```

It is a little tricky deleting or redefining a substitution for a letter since the letter itself must be specified in the DEFINE command and the substitution for the letter takes place as the command is being typed. If you need to delete or redefine a letter substitution, invoke the SET SL NOSUBSTITUTE command to turn off substitutions, then use the DEFINE command, then type SET SL SUBSTITUTE.

You may define a key to invoke a command file. If you do this, be sure to enclose the string containing an at-sign (@) in quotation marks so that the command processor will not try to invoke the command file when it processes the DEFINE KEY command. For example, the following command defines the PF2 key to invoke a command file named RUNACT:

```
DEFINE KEY PF2 "@RUNACT"
```

It is possible to include control characters in the key substitution string. To do this, enclose the substitution string in quotation marks and type a caret (^) in front of the letter that is to be a control character. For example, the following command defines numeric keypad key 0 (zero) as control-U:

```
DEF/KEY/NOTERMINATE KPO "^U"
```

Control characters count as one character in the string even though two characters are used to specify them. If the DEFINE command is placed within a command file, two caret characters must be specified to denote each control character because command file processing translates two consecutive caret characters into a single caret. Thus if the previous DEFINE command were placed in a command file, it would be written as:

```
DEFINE KEY/NOTERMINATE KPO "^ ^U"
```

It is possible to define a key with a control-W digit sequence to cause a switch to a subprocess; however, this is less powerful than actually typing control-W digit because the defined key method can only be used at a time when the single line editor is accepting input -- not at a time when a program is executing. The following key definition defines PF2 to switch to subprocess 1:

```
DEFINE KEY/NOTERMINATE PF2 "^W1"
```

If this command was placed in a command file, the defined string should be "^ ^W1".

Key definitions are passed from the primary process to a subprocess when the subprocess is initialized.

A SHOW KEYS command is available to display the current key definitions.

The maximum number of keys that may be defined by each job is controlled by the KEYMAX sysgen parameter. The maximum value that may be assigned to KEYMAX is 60.

The key definitions are stored in a local PLAS region named "KEYDEF". This region is created when the first key definition is made and is deleted when the last key definition is deleted or the job logs off. The size of the KEYDEF region (in bytes) is equal to KEYMAX*68.

Note, the PI handler which controls the console terminal on a Professional personal computer normally provides VT100 support for the console terminal -- not VT200. Hence the function keys on the top row and the "Do", "Help", "Find", etc. keys may not be defined. However, it is possible to enable the Pro console for VT200 compatible operation by sending it the following control sequence: "<ESC>[?39h". The Do, Help, Find, and other keys will be usable after issuing this control sequence.

10. Time-sharing line / CL line cross connection

It is now possible to cross connect a time-sharing line with a CL (communication line) line in such a fashion that all characters received from the time-sharing line are transmitted directly to the CL line and all characters received from the CL line are transmitted directly to the time-sharing line. This is useful to allow a time-sharing line on one TSX-Plus system to be used as a terminal on another system connected through a CL line.

This function is similar to using VTCOM to communicate through a CL line but has the advantage that there is much less overhead because the cross connection is made at a low level within TSX-Plus such that characters do not have to be passed to a running application program. Of course the internal cross-connection feature does not provide the file transfer capabilities of VTCOM.

The keyboard command used to establish a cross connection has the form:

SET HOST/PORT=ddn

where "ddn" is the name of a CL or Cl device to which your terminal is to be cross connected. For example, the following commands would connect CL unit 1 with terminal line 4 at 9600 baud and then cross connect the current terminal with the CL unit:

```
SET CL1 LINE=4,SPEED=9600
SET HOST/PORT=CL1
```

A logical name that has been assigned to a CL or Cl unit may be used as the port name. For example:

```
ASSIGN CL3 VAX
SET CL3/LINE=21/SPEED=9600
SET HOST/PORT=VAX
```

The program name "\$HOST\$" is shown for the line while a cross connection is in effect.

TERMINAL privilege is required to use the SET HOST command. Once the cross connection is established, characters typed at your terminal are transmitted to the CL line. Control characters such as control-C, control-W, etc. are not interpreted by the local system. There are only four control characters which are interpreted by the local system: control-S (XOFF - stop terminal output); control-Q (XON - resume terminal output); control-A (special command); and control-\ (control backslash - release connection).

The cross connection can be released at any time by typing control-\ (control backslash). When this is done, the message "Cross connection released" is printed and control returns to the keyboard monitor. A character other than control-\ can be chosen as the connection-release character by use of the CCXTRM parameter in TSGEN.

Typing control-A signals the cross connection controller that a special character follows. The following characters, when typed immediately after control-A, perform the specified functions:

- B Send a 0.5 second break signal.
- D Raise DTR signal.
- H Drop DTR signal (hangup).
- R Reset flag that says an XOFF (control-S) character has been received, and transmit an XON (control-Q) character.

If any other character is typed after control-A, the character is sent to the CL line without interpretation. This provides a way to transmit the control-A and control-\ characters. If you type control-A twice, one control-A character will be transmitted to the CL line. Typing control-A control-\ will cause a control-\ character to be transmitted rather than releasing the cross connection. A character other than control-A can be chosen as the special-function character by use of the CCXCTL parameter in TSGEN.

The DTR signal is automatically raised when a SET HOST command is issued. The "D" and "H" function characters can be used to control the DTR signal after the connection is made. The DTR signal is not automatically dropped when the cross connection is broken.

11. Change to SEND command

A change has been made to the SEND command which is used to send messages to terminals. Previously, if no number was specified, the message was sent to all lines. Now, a line number must be specified with the command. The form of the command is:

SEND,linenum message

where "linenum" is the line number specification and "message" is the message to be sent.

In addition to a numeric line number, the following special words may be used for the line number:

ALL = Send message to all lines.
OPERATOR = Send message to operator's console.
PARENT = Send message to parent job for this job. The job which initiates the execution of a detached job is the "parent" job of the detached job. Hence SEND,PARENT... can be used to send a message from a detached job to the job which initiated it.

12. Changes to subprocesses (virtual) and detached jobs

At the time that they are started, subprocesses (virtual lines) and detached jobs now "inherit" the following information from the primary process:

1. The authorized and set privileges.
2. The current and maximum authorized priority.
3. ASSIGN commands.
4. Logical disk mounts.
5. Mounts of disks to enable caching.
6. File access controls specified with ACCESS command.
7. User defined commands (i.e., UCL commands).
8. User defined terminal keys (see DEFINE KEY command).

9. Project-programmer numbers and job name.
10. The keyboard monitor command prompt string.
11. Parameters specified with the SET PRINTWINDOW command.
12. Single Line Editor status.
13. User Command Interface (UCI) information.

Subprocesses (virtual lines) have been changed so that they no longer normally execute the startup command file for the line that started them. If you wish to specify a startup command file for subprocesses, you may do so by placing a command of the following form in the startup command file for the primary process:

SET SUBPROCESS/FILE=file

where file is the file specification for the startup command file to be executed when any subprocesses are initiated. The SET SUBPROCESS/FILE command is only legal within a startup command file.

The DETACH command, and the EMT for starting a detached job, now accept up to 80 characters of information to be passed to the detached job. The first item must be the name of a command file to be started; parameters for the command file may be specified following the command file name. Parameters may be passed to a detached command file in the same way as they are passed to ordinary command files under TSX-Plus (e.g., @COMFIL parm1 parm2 ...). For example, the following command would start a detached job to execute a command file named BCOMP with the parameter ARFILE:

DETACH BCOMP ARFILE

Up to 80 characters may also be specified for the command file and parameters for system detached jobs specified in TSGEN with the DETACH macro.

The job which initiates a detached job is known as the "parent" of the detached job. No privilege is required for a parent job to kill a detached job which is its own "child". SAME, GROUP, or WORLD privilege is required to kill detached jobs which are children of other jobs. WORLD privilege is required to kill a detached job that is started automatically because of a DETACH macro in TSGEN.

The ability of a detached job to acquire the context of the job which started it simplifies the process of creating detached jobs to perform low priority "batch" tasks. For added flexibility, the RT-11 IND control file processor may be invoked by calling it from the command file which is detached. (An IND control file cannot be detached directly.) In addition, parameters may be passed through the DETACH command, through the inter-

mediate detached command file, into the IND local symbol 'COMMAN'. The following example IND control file demonstrates the use of the IND control file processor from a detached job to perform a "batch" assembly of a designated file. The string parsing used to build the file specification may vary in detail among different versions of IND. The SEND,PARENT command is used to report the outcome of the assembly to the parent job. Even though detached jobs are not directly associated with any time-sharing line (terminal) and terminal output from detached jobs is eventually discarded, it may be captured first in a log file. This allows the terminal output logging facility to collect any errors reported by MACRO.

```

.; BMACI -- IND control file to assemble a file as a
.; detached process and report assembly results
.; to parent.
.;
.; Invoke as: DETACH SY:BMAC <filename>
.;
.; Calls SY:BMAC.COM which contains only
.;
.; IND SY:BMACI ^1
.;
.; Invoking this control file to do the work.
.;
.ENABLE SUBSTITUTION
.SETS BELL "" .;CONTROL-G BELL
.;Make sure filename was specified
.IF COMMAN <> "SY:BMACI" .GOTO CONTIN
SEND,PARENT ?BMAC-F-No file specification 'BELL'
.EXIT
.CONTIN:
.;Get name of file to be assembled
;COMMAN = 'COMMAN'
.PARSE COMMAN " " P1 P0
.;Strip device name into P1
.PARSE P0 ":" P1 P2
.;If no device specified, make it "DK"
.IF P2 <> "" .GOTO 1
.SETS P2 P1
.SETS P1 "DK"
.1:
.;Strip extension into P3
.PARSE P2 "." P2 P3
.IF P3 = "" .SETS P3 "MAC"
SET PROC/NAME='P2'/PRIO=10

```

```

    .;See if we can find file
    .SETS P0 P1+":"+"P2+"+"P3
    .TESTFILE 'P0'
    .IF <FILERR> = <SUCCES> .GOTO OLDFIL
SEND,PARENT ?BMAC-F-Can't find file >'P0'< 'BELL'
    .EXIT
.OLDFIL:
    .;Start log file
    .SETS P4 P1+":"+"P2+"+"LOG"
SET LOG FILE='P4'
    .;Assemble as requested
MAC 'P0'
    .;Branch on error
    .IF <EXSTAT> <> <SUCCES> .GOTO OOPS
SEND,PARENT Assembly of 'P0' finished successfully
SET LOG CLOSE
DELETE 'P4'
    .EXIT
.OOPS:
SET LOG CLOSE
SEND,PARENT 'BELL' 'BELL' Assembly of 'P0' had errors
SEND,PARENT See log file 'P4'
    .EXIT

```

13. System service call (EMT) to acquire the file context of another job

A new system service call (EMT) has been implemented to allow one job to acquire the file context of another job. The form of this EMT is:

```
EMT      375
```

with R0 pointing to an argument block of the following form:

```

.BYTE    0,160
.WORD    jobnum
.WORD    0

```

where "jobnum" is the number of the job whose file context is to be acquired.

The following error codes may be returned by this EMT:

Error 0 = Job is not privileged to use this EMT.

Error 1 = Specified job number (jobnum) is not valid or job is not logged on.

GETCXT privilege is required for a job to use this EMT except in the case where the job being accessed is the parent job of the job executing the EMT.

When this EMT is executed, the following actions occur:

1. All channels for the job which are opened to files on logical disks are purged.
2. All devices mounted by the job are dismounted.
3. The following items are copied from the job being accessed, replacing the previous information for the job executing the EMT:
 - a) ASSIGN commands.
 - b) ACCESS command restrictions.
 - c) Logical disk information.
 - d) Mounted device information.

This EMT allows one job (typically a detached job) to perform file service operations for other jobs by taking on the file context of the job it is servicing.

RTSORT version 2.0 can use this feature when running in message communication mode to allow it to sort files on logical disks and using logical device names. To invoke this feature with RTSORT version 2.0, specify the /GETCXT RTSORT command switch following the /MESSAGE switch. The recommended start-up command file to initiate RTSORT as a detached job when the system starts is as follows:

```
SET PROC/PRIV=(STANDARD,GETCXT)
R RTSORT
/MESSAGE/GETCXT
```

14. SUSPEND and RESUME commands

SUSPEND and RESUME keyboard commands have been implemented in a fashion analogous to the RT-11 SUSPEND and RESUME commands. The SUSPEND command is used to suspend the execution of a job, the RESUME command resumes execution of a suspended job. The form of these commands is:

```
SUSPEND job_number
```

```
RESUME job_number
```

where job_number is the number of the job to be suspended or resumed. SAME privilege is required to suspend or resume a job with the same project-programmer number as the job issuing the command. GROUP privilege is required if the affected job has the same project number but a different programmer number. WORLD privilege is required if the affected job has a different project number.

15. Support for quad serial line multiplexer on Professional

Support is now provided for the quad serial line multiplexer on the Professional-350/380 personal computers. By use of this hardware option, TSX-Plus can support four time-sharing lines in addition to the console, printer port, and communications port.

The quad line controller (QLC) may be installed in any free option slot. The cable that connects the controller card to the terminal connector box plugs into the top of the controller card through an opening in the top of the card cage. The cable should be routed down the back of the card cage. The cable exits through an opening in the back of the Pro box made by removing a plastic tab that is immediately above the "NET 1" connector.

There are two configurations of the QLC determined by which plug inside the terminal connector box the connecting cable is joined to. If the cable is connected to J1, ports 0 and 2 have modem control but lines 1 and 3 do not. If the cable is connected to J6, no lines have full modem control. It is not possible to have full modem control on all of the ports. It is possible to determine which configuration has been selected without disassembling the connector box. If the configuration that supports modem control on lines 0 and 2 has been selected, the 12-pin side of the 25-pin connectors are closest to the cable.

Port 0 is the connector with the 12-pin side closest to the end of the box; port 3 is the connector with the 13-pin side closest to the other end of the box.

If the QLC is installed, a time-sharing line connected to each of the four ports will automatically become available when TSX-Plus is started. The SHOW TERMINALS command displays "QP" (quad port) as the device type for the lines.

16. PRO/TSX-Plus console support

PRO/TSX-Plus does not support the Professional 380 high-resolution graphics; the national replacement character (NCR) language compose key or foreign keyboards; or the GIDIS graphic commands (special functions 370 and 371). Some options of the SETUP program may be used under PRO/TSX-Plus to control the operation of the Professional. However, the following SETUP commands options are not functional and should not be used: 480INTERLACE, COMPOSE, DATA, INTERLACE, LANG, SAVE, RESET, RETAIN, and TYPE. The SETUP SAVE function should not be used as it may damage the PI.SYS file.

17. ENDPAGE and ENDSTRING options for CL handler

Two new SET options are now available for the CL (Communications Line) handler to control end-of-file output processing. The ENDPAGE option specifies the number of form-feed characters which are appended to an output file when the CL channel is closed. The ENDSTRING option specifies a character string which is appended to the output file when the CL channel is closed. The form of these options are:

```
SET CLn ENDPAGE=number
SET CLn ENDSTRING="string"
```

The number specified with the ENDPAGE option determines how many form-feed characters will be added to the end of each file. The default value is zero. The string provided with the ENDSTRING option specifies a character string of up to seven characters in length which is appended to the end of each output file.

The ENDSTRING string may contain control characters. A control character is specified by prefixing the normal character by a caret (^). For example, the following specification causes "EOF" to be printed followed by carriage-return, line-feed, control-Z:

```
SET CL ENDSTRING="EOF^M^J^Z"
```

Control characters count as one character in the string even though two characters are used to specify them. If the SET command is placed within a command file, two caret characters must be specified to denote each control character because command file processing translates two consecutive caret characters into a single caret. Thus if the previous SET command were placed in a command file, it would be written as:

```
SET CL ENDSTRING="EOF^^M^^J^^Z"
```

The ENDSTRING string may be enclosed by apostrophe (') or quote (") marks.

The ENDPAGE and ENDSTRING options take effect when an I/O channel is closed to a CL unit to which a write operation has been performed. That is, you must write some data to a CL unit in order for the ENDPAGE and ENDSTRING options to do anything on the close. The ENDPAGE and ENDSTRING options may be used with spooled and nonspooled CL units. If both ENDPAGE and ENDSTRING are specified, the ENDPAGE option is performed first.

A .SPFUN is available to allow a program to specify ENDPAGE and ENDSTRING parameters for a CL unit. The .SPFUN function number is 264 (octal). The data is provided in a buffer pointed to by the .SPFUN. The first word of the buffer contains the value to set for the ENDPAGE parameter. The second and subsequent words of the buffer contain the ENDSTRING string in ASCIZ form; any characters beyond the first 7 are ignored.

18. New reset options for CL handler

A new special function (.SPFUN) has been added to reset the status of a CL unit. A SET command is also available to invoke this function. The reset function has .SPFUN code 265 and performs the following operations:

1. Empty the input ring buffer of received characters.
2. Empty the output ring buffer of transmitted characters.
3. Stop sending a break if one is currently being sent.
4. Clear XOFF (control-S) received flag.
5. Send an XON if we have previously sent an XOFF.
6. Clear end-of-file status and reset line and column numbers.

The SET CL RESET command now invokes this function rather than function 201. The new SET CL XON command can be used if you wish to invoke special function 201 which clears the XOFF status of the CL unit without performing the other functions of the RESET option.

19. Terminating start-up command files

A SET command has been added which can be placed within a start-up command file to mark the end of the portion of the command file that is to execute with start-up privilege. Normally start-up command files execute with special privilege: commands such as ACCESS, SET LOGOFF, and SET SUBPROCESS/FILE are only legal within start-up command files and start-up command files cannot be aborted by typing control-C. This special privilege normally continues until the end of the start-up command file is reached or a program within the start-up command file performs terminal input. A command of the form:

```
SET ENDSTARTUP
```

can be placed within a start-up command file to terminate the special start-up privilege. Following the execution of this command the start-up command file is executing as a normal command file -- it can be aborted by typing control-C and command such as ACCESS are not allowed.

This command can be used within a start-up command file to separate the portion of the file that contains restricted commands (such as ACCESS) from a non-privileged start-up command file which can be edited by non-privileged users. For example, the following start-up command file provides access to a limited set of devices and then invokes a non-privileged user start-up command file:

```
ACCESS DLO,DL1
SET ENDSTARTUP
@DL1:STRTUP.COM
```

20. TSXMOD System configuration utility

TSXMOD is a specialized patching utility which allows reconfiguration of TSX-Plus without performing a system generation. The principal application of TSXMOD is to PRO/TSX-Plus which does not include system generation capability. Although regenerating TSX-Plus is relatively simple (typically requiring only a few minutes), sometimes the minimal desired changes do not seem to warrant even a simple re-generation. TSXMOD alters the disk image of TSX.SAV so that when TSX-Plus is next restarted the modified information is used.

TSX-Plus is distributed as a collection of files, including several object modules and a configuration file (TSGEN.MAC). System generation involves editing parameters in TSGEN.MAC to reflect site specific hardware characteristics and operating preferences. This is then assembled and linked with the object modules provided to create the executable image of the operating system, TSX.SAV. Because TSXMOD alters specific relative locations in TSX.SAV, it must be linked with the symbol table for TSX.SAV. This is done automatically by the TSXLNK command file when rebuilding TSX. If you wish to re-generate TSX, but preserve the old file (TSX.SAV) and be able to use TSXMOD to patch it at a later time, then you must also save the TSXMOD which was linked with the TSX.SAV. The four files which are dependent on changes in TSGEN are: TSX.SAV, TSKMON.SAV, SYSMON.SAV and TSXMOD.SAV. They must always be saved and restored together.

When TSX is started, it uses the configuration information from TSGEN to select most operating characteristics. Although many of these characteristics may be dynamically modified by keyboard commands while TSX is executing, some of the information is used during initialization and cannot be changed during execution. It is modification of this information for which TSXMOD is most appropriate. Most, but not all, of the parameters in the user-modifiable portion of TSGEN can be configured by TSXMOD. Those parameters which affect the amount of memory reserved in TSGEN are generally not alterable without revising TSGEN.MAC and rebuilding TSX.

Note: TSXMOD modifies the TSX.SAV file on disk, not the not the currently executing TSX image. Therefore it is necessary to restart TSX to have a change made by TSXMOD take effect.

20.1 Running TSXMOD

Since TSX does not refer to its disk image during execution, TSXMOD may be used to modify TSX.SAV while running either RT-11 or TSX-Plus. TSXMOD requires approximately 30 Kb of memory.

When TSXMOD is started, it first tries to load DK:TSX.SAV. A file is only loaded successfully if it can be opened and read and if it was linked with the appropriate symbol table. If DK:TSX.SAV cannot be loaded, an informational message is displayed and SY:TSX.SAV is tried next. If

neither can be loaded, then there is nothing loaded to be patched and the only valid commands are: READ or QUIT. Whenever either DK: or SY:TSX.SAV is loaded into memory, it becomes the "current file". If a READ or WRITE command is issued without a file name, I/O is to the "current file". If either the READ or WRITE command specifies a file name, then that file becomes the "current file". The current parameters are rewritten to the "current file" when TSXMOD is exited if any changes have been made since the last WRITE.

20.2 TSXMOD Commands

There are two categories of commands to TSXMOD. The first type is used to configure TSX, and the other is used to control TSXMOD. The format of configuration commands is identical to the format for the same commands in TSGEN.MAC. (Note: the TSXMOD prompt is an exclamation point "!") Some configuration command examples are:

```
!SWDBLK: .RAD50 /SY TSXSWPTSX/
!HIMEM = 64
!BUSTYP = QBUS
```

The current values of all parameters and macro calls may be displayed with the SHOW command followed by the name of the parameter or macro call. For example:

```
!SHOW QUANO
QUANO = 2
!SHOW NRMFLG
NRMFLG = $DEFER+$LC+$PAGE+$ECHO
```

Commands used to control TSXMOD are used to READ and WRITE files, exit TSXMOD with or without saving current data, or SHOW such information as current values or an abbreviated link map. For example:

```
!SHOW FILE
!Current file is DK :TSX .SAV
!QUIT
```

20.3 Creating TSXMOD command files

Commands to TSXMOD may be accepted either from the terminal or from a command file. This allows the output from a previous session of TSXMOD, which was recorded using the terminal logging facility of TSX-Plus, to be edited and serve as a means of restoring a given configuration without rebuilding TSX or maintaining full copies of TSX.SAV or even the shorter copies preserved with a TSXMOD WRITE command.

We also recommend creating a log file then running TSXMOD and issuing the SHOW ALL command in order to maintain a record of your current configuration. In fact, when communicating a problem report or inquiry

regarding operation of TSX-Plus, it is useful to include the information from a monitor SHOW ALL command and another from within TSXMOD. The following command file demonstrates this:

```
SET LOG FILE=CONFIG !OPEN LOG FILE
SHOW ALL           !SHOW RUNNING CONFIGURATION
R TSXMOD           !RUN TSXMOD
SHOW ALL           !SHOW CURRENT TSGEN CONFIGURATION
QUIT               !QUIT TSXMOD
SET LOG CLOSE      !CLOSE LOG FILE
PRINT CONFIG.LOG   !PRINT THE LOG FILE
```

In order to use a log file as a command file to reconfigure TSX: while running TSX-Plus, start a log file, run TSXMOD, issue the SHOW ALL command, QUIT TSXMOD, and close the log file. With some minor editing of this log file, it is simple to reconfigure TSX or restore a known configuration by invoking the log file as a command file. In the following example, the underlined portions indicate those commands or characters which should be removed by editing along with any other desired parameter changes.

```
.SET LOG FILE=MOD.COM !THIS LINE WON'T BE IN LOG FILE
.RUN TSXMOD
!TSXMOD-I-Input file -- DK :TSX .SAV
```

```
!SH ALL
SWDBLK: .RAD50 /SY TSXSWPTSX/
SPLBLK: .RAD50 /SY TSXSPLTSX/
RSFBLK: .RAD50 /SY TSXRSFTSX/
HIMEM = 64.
.
.
.
!TSDHIO 050354
!TSCLR 052360
!TSINIT 052606
! INITOP 100324
!QUIT
!TSXMOD-I-Session not saved.
.SET LOG CLOSE
EXIT !ADD THIS LINE
```

This use of a logged and edited TSXMOD session as a command file to reconfigure TSX is also the reason why the exclamation point is used as the TSXMOD prompt character and prefixes most of its text lines. When accepting input from a command file, TSXMOD ignores everything on a command line after an exclamation point.

20.4 Commands used to control TSXMOD

The following commands are used to control the operation of TSXMOD, such as selecting the file to be modified or exiting the program.

Command	Explanation
DELETE	Prefixes the DEVDEF, RTDEF or DETACH configuration commands in order to remove an entry and replace it with an unused entry.
EXIT	Exits TSXMOD after writing memory to the current file. (Does not automatically write unless changes have been made.)
READ [<u>file</u>]	Loads memory from the current file or optionally from the specified file. If a file is specified, it becomes the current file.
SET TERMINAL <u>n</u>	Selects the current time-sharing line-definition block. (The SHOW TT <u>n</u> and SHOW LINDEF <u>n</u> commands also set the current line number.) Time-sharing lines are numbered in the order in which they are defined in TSGEN.MAC.
SHOW <u>keyword</u>	Displays the current value of a system parameter. The value in memory is the current value and will not reflect the value in TSX.SAV if it has been changed until memory is rewritten to the disk file with a WRITE command or upon EXIT. If no keyword is specified with the SHOW command, the current value of all system parameters is displayed.
SHOW FILE	Displays the current file name.
SHOW MAP	Displays TSX module entry point addresses. (The most useful information from the link map.)
QUIT	Terminates TSXMOD without rewriting to a disk file.
WRITE [<u>file</u>]	Causes memory to be rewritten to the current file or optionally to the file specified. If a file is specified, it becomes the current file.

20.5 TSXMOD Configuration Commands

Parameters which are normally configured in TSGEN by an equate statement (e.g., QUANO = 1) or by a macro call (e.g., RTDEF <SY CBR060SHR>,R,1) may be configured by entering the same type of statement as a command to TSXMOD. (We will refer to these types of equate statements and macro calls in the user-modifiable section of TSGEN.MAC as TSGEN statements.) Current values of all TSGEN statements may be displayed with SHOW commands.

20.5.1 Simple numeric parameters

The most common type of parameter is simply equated to a numerical value. (Note that decimal is the default radix for all input.) For example:

```
!SHOW DFLMEM
DFLMEM      = 56.
!DFLMEM = 64.
!SHOW DFLMEM
DFLMEM      = 64.
```

20.5.2 Simple non-numeric parameters

Some parameters are equated to symbolic named values. For example:

```
!SHOW UCLORD
UCLORD      = MIDDLE
!UCLORD = LAST
!SHOW UCLORD
UCLORD      = LAST
```

20.5.3 RAD50 parameters

Some parameters which declare file names use the MACRO syntax for declaring a .RAD50 value. For example:

```
!SHOW SWDBLK
SWDBLK:      .RAD50 /SY TSXSWPTSX/
!SWDBLK: .RAD50 /DLOSWPFILTSX/
!SHOW SWDBLK
SWDBLK:      .RAD50 /DLOSWPFILTSX/
```

20.5.4 Device definitions:

Device handlers to be loaded by TSX-Plus are declared with the DEVDEF macro. The 2-character device name must be enclosed by angle brackets and may be followed by several optional symbolic parameters. For example:

```
!SHOW DEVDEF
DEVDEF <DL >
DEVDEF <DX >
DEVDEF <LP >,MAPH
DEVDEF <NL >
```

If unused device definition slots are available, new devices may be added simply by declaring a new DEVDEF. For example:

```
!DEVDEF <RK>
!SHOW DEVDEF
DEVDEF <DL >
DEVDEF <DX >
DEVDEF <LP >,MAPH
DEVDEF <NL >
DEVDEF <RK>
```

Device definitions may also be removed with the DELETE command. For example:

```
!DELETE DEVDEF <RK>
!SHOW DEVDEF
DEVDEF <DL >
DEVDEF <DX >
DEVDEF <LP >,MAPH
DEVDEF <NL >
```

20.5.5 SPOOL

Devices (usually printers) which are to be spooled by TSX-Plus are declared by the SPOOL macro. This macro is not changeable by TSXMOD. However, the spool disk file size (the fourth parameter to the SPOOL macro) may be changed by adjusting a parameter named SPLSIZ. For example:

```
!SHOW SPOOL
!SPOOL      2.,20.,2.,2000.,<LP CL2>,0,5.
SPLSIZ = 500
!SHOW SPOOL
!SPOOL      2.,20.,2.,500.,<LP CL2>,0,5.
```

20.5.6 Shared run-times (RTDEF)

Shared run-time areas are declared using the RTDEF macro. The name of the file to be loaded is specified in angle brackets, followed by "R" or "RW" if the shared run-time memory is read-only or read-write access, and then the number of blocks to skip when loading the disk file into memory. RTDEFs may be deleted with the delete command. If free RTDEF slots are available, new RTDEFs may also be declared. For example:

```

!SHOW RTDEF
RTDEF      <SY CBRO60SHR>,R,1.
!RTDEF     <DLOSHAREDRTS>,RW,1
!SHOW RTDEF
RTDEF      <SY CBRO60SHR>,R,1.
RTDEF      <DLOSHAREDRTS>,RW,1.
!DELETE RTDEF <SY CBRO60SHR>,R,1
!SHOW RTDEF
RTDEF      <DLOSHAREDRTS>,RW,1.
!DELETE RTDEF <DLOSHAREDRTS>,RW,1
!SHOW RTDEF
!No shared run-times (RTDEF) declared.

```

20.5.7 Unchangeable parameters

The following TSGEN parameters may not be changed by TSXMOD: EXTMCH; MAXSEC; MAXALC; MXSPAC; WILDFL; SPOOL (although spool disk file size may be changed by adjusting SPLSIZ); RTVECT; PMSIZE; default time-sharing line parameters (DINSPC, DOTSPC, OTRASZ, NCSILO, NCXOFF, NCXON, CLXTRA, CLORSZ, NRMFLG); TBLDEF; LINDEF; LINEND; MUXDEF; DHDEF; DHVDEF; DZDEF; MUXEND; CLDEF; CLEND. For example:

```

!EXTMCH = 0
!This parameter not adjustable without rebuilding TSX.
EXTMCH   = 1.

```

20.5.8 Time-sharing line definitions

Neither terminal line definitions nor multiplexer definitions may be changed by TSXMOD, but the optional macro calls within each line definition block may be changed. Current terminal line characteristics may be displayed with the SHOW LINDEF [n] command, or an abbreviated display may be obtained with the SHOW TERMINAL [n] command. The optional number [n] indicates which time-sharing line is to be current. If n is zero (0), then all lines will be displayed. For example:

```

!SHOW TERMINAL 0
!Line  Type      Vector      CSR      Terminal  Speed      Line Name
!-----
!1      OPER      DL -    060    177560    VT200      9600 8N    Console
!2      Local     DH -0   330    160060    VT200      Auto  8N    Charles G. Br
!3      Local     DH -1   330    160060    VT100      9600 8N    Sir N Dippity
!4      Local     DH -4   330    160060    VT100      Auto  8N    VAX link
!5      Local     DH -5   330    160060    VT100      9600 8N

```

20.5.9 Time-sharing line options

In order to change line options, you must select a current line with one of the commands: SHOW LINDEF n; SHOW TERMINAL n; SET TT n; or SET TERMINAL n. You may then set any line option by issuing the appropriate TSGEN statement. For example:

```

!SHOW LINDEF 2
SET TT 2.
! LINDEF 0.
  CMDFIL LINE1.TSX
  FLAGS NRMFLG+$AUTO+$FORM+$TAB
  NAME <Charles G. Br>
  SPEED S19200,8.,NONE
  TRMTYP VT200
! LINEND
!TRMTYP VT100
!SHOW TRMTYP
  TRMTYP VT100

```

The syntax for the FLAGS macro differs slightly from that used in TSGEN.MAC where the exclamation point is used to separate individual flags. In TSXMOD the "+" or "-" symbols separate the individual flags. For example:

```

!FLAGS NRMFLG+$SCOPE+$PAGE
!SHOW FLAGS
  FLAGS NRMFLG+$SCOPE
!SHOW NRMFLG
NRMFLG = $DEFER+$LC+$PAGE+$ECHO

```

In addition to declaring a completely new set of flags for a line, you may also turn on or off individual flags. If the first flag is not preceded by a "+" or a "-", then all flags are cleared, each desired flag must be included, and flags must be separated by a "+" sign. To make individual flag changes, simply use a "+" or "-" before the first flag. The existing flags are then not cleared, flags preceded by a "+" sign will be added, and flags preceded by a "-" sign will be deleted. For example:

```

!FLAGS -$SCOPE+$FORM
!SHOW FLAGS
  FLAGS NRMFLG+$FORM

```

20.5.10 DETACH

Command files which are to be automatically executed on detached lines whenever TSX-Plus is started from RT-11 are declared with the DETACH macro. The total number of detached jobs which may be running at any given time is declared with the TBLDEF macro, which may not be changed by TSXMOD. Start-up detached jobs may be declared with the DETACH command by TSXMOD if free start-up detached job slots are available. For example:

```
!DETACH SY:STRUP.TSX
```

To remove an automatically started detached job, use the DELETE command. Only the minimum number of characters needed to uniquely identify the detached job are necessary. Lower case characters are not automatically converted to upper case. For example:

```
!SHOW DETACH
DETACH <SY:DETACH.TSX>
DETACH <SY:STRUP.TSX>
!DELETE SY:DE
!SHOW DETACH
DETACH <SY:STRUP.TSX>
```

If parameters (up to 80 characters) are to be passed to the start-up detached job, the command file name and the parameters must be enclosed in angle brackets. For example:

```
!DETACH <SY:NEWDET.COM Tuesday 13:00>
```

21. System crash dump option

It is now possible to cause the system to print a dump of internal information about the status of the system if a system crash occurs. This is primarily useful to the organization that provides your system support in helping you to diagnose system difficulties. The dump option requires approximately 3.2 Kb of memory space in the mapped portion of the system and is only loaded into memory if you request it during system generation. It is recommended that you not include the system dump facility unless you are experiencing system crashes.

There are three sysgen parameters associated with the crash dump facility:

SYSDMP controls whether the crash dump facility is loaded into memory and whether a dump is produced if a system crash occurs. Set SYSDMP to 1 (one) to include the crash dump facility, or 0 (zero) to exclude it.

DMPTCR specifies the address of the transmitter control register for the device to which the dump is to be written. The dump listing device may be any DL-11 type serial communications port (including the console terminal) or it may be a parallel printer port. XON/XOFF support is provided for serial ports. Specify 177564 to cause the dump to be written to the console terminal; specify 177514 to cause the dump to be written to a parallel printer connected to the standard LP address.

DMPKTP controls whether a system crash occurs on all kernel mode traps. If DMPKTP is 0 (zero; the recommended case) kernel mode traps that occur during EMT processing abort the currently executing job but do not halt the system. If DMPKTP is 1 (one), a system crash (and dump if selected) will happen if any trap occurs in kernel mode.

It is possible to force a system crash to generate a dump for a hung system by halting the processor and causing it to resume execution from location 0 (zero) which contains a jump to a crash generation routine. If you do this, be sure to write down the address of the program counter before you transfer control to zero.

22. Control of swap file size

A new sysgen parameter "SWPSLT" has been added which allows you to control the size of the job swap file. If SWPSLT is set to 0 (zero), the system operates as it did previously and assigns enough slots in the swap file for all lines that are generated into the system. If SWPSLT is assigned a value in the range 1 up to the number of jobs, then only that number of job swap slots is allocated for the swap file.

The SWPSLT parameter may be used to reduce the size of the job swap file (TSXSWP.TSX) in the case where more jobs are generated into the system than are actually logged on at any one time. It also may be useful in systems which only rarely swap jobs. Note however that a system crash will occur if the system needs to swap a job and no free space is available in the swap file. In this case the following error message is printed:

```
?TSX-F-Fatal system error
SFO-Swap file overflow
```

23. Support for named PLAS regions

TSX-Plus now supports named PLAS memory regions in a fashion compatible with RT-11. Named PLAS regions are areas of memory which may be allocated by jobs and shared between multiple jobs. Unlike unnamed PLAS regions, named regions are not necessarily deallocated when the program which created the region terminates execution. Named regions may be used to communicate between programs, to hold common code executed by multiple users (e.g., shared run-times), and to pass information from one program to another program - possibly run at a later time.

In addition to supporting named global PLAS regions in a fashion compatible with RT-11, TSX-Plus also provides an additional facility known as local named regions. Local named regions are regions which can only be accessed by the job that created them. Their names are private to the creating job and more than one job may create (different) local regions with the same name. Local named regions are deallocated when the creating job specifies that they are to be eliminated or when the creating job logs off.

Local named regions are allocated memory space associated with the creating job and are swapped in and out of memory with the job. Global named regions are allocated memory space at the top of the area of memory used for jobs. They are never swapped out of memory and are only deallocated when a job eliminates the region. Thus global named regions may continue to occupy memory after the job that created the region logs off. Caution should be exercised when creating global regions since it is possible to lock jobs out of memory by creating large global regions.

Local named regions are distinguished from global named regions at creation time by setting bit 0 (mask 000001) in the status word (the third word - R.GSTS) of the region definition block used with the .CRRG EMT.

A SHOW REGIONS keyboard command may be used to display information about local and global named regions accessible by the job. The form of the display produced by this command is shown below:

Region	Size	Type	Job	Use	Shared	AGE
-----	-----	-----	---	---	-----	---
REGN1	2.0	Local	15	0	No	No
REGN2	4.0	Global	15	0	Yes	Yes

The "Region" column displays the name of the region; "Size" shows the region size in units of K-bytes (and tenths); "Type" indicates whether the region is local or global; "Job" shows the number of the job that created the region; "Use" shows the number of jobs that are currently attached to the region; "Shared" is "Yes" if the region may be accessed by jobs other than the creating job and "No" if the region is reserved for exclusive use by the creating job (note that local named regions will always be "No"); and "AGE" is "Yes" if the Automatic Global Elimination option was specified for the region.

The REMOVE keyboard command may be used to eliminate a local or global named region. The form of this command is:

REMOVE region

For example, the following command eliminates a region named REGN1:

REMOVE REGN1

24. Terminal input completion routine

It is now possible to establish a completion routine which will be executed as each character is received from the terminal. The form of the EMT argument block used to do this is:

```
.BYTE      1,133
.WORD      complRoutine
```

where "compl routine" is the address of the completion routine that is to be entered for each input character.

On entry to the completion routine, the received character is in R0. The EMT to connect the completion routine must be reexecuted each time a character is received; usually this is done at the end of the completion routine after processing the last received character. The program should be running with single character activation and with bit 12 set in the job status word.

The following program illustrates how this EMT can be used to read and echo characters.

```

        .MCALL .TTYOUT,.SPND
;
; EMT argument block used to connect completion
; routine (CPLRTN) with terminal character input:
;
CPLEMT: .BYTE 1,133
        .WORD CPLRTN
;
; Set special-TT-mode flag in Job status word
;
START:  BIS      #010000,@#44      ;Special-TT-mode to JSW
;
; Connect completion routine to receive first character
;
        MOV      #CPLEMT,R0        ;Set compl routine for TT input
        EMT      375
;
; Now suspend mainline program execution.
; Completion routine will do all character processing.
;
1$:     .SPND
        BR       1$
;
; Completion routine entered as each character is received.
; On entry, R0 contains the received character.
;
CPLRTN: .TTYOUT                      ;Echo the received character
        MOV      #CPLEMT,R0          ;Reconnect compl rtn for next char
        EMT      375
        RETURN                      ;Return from completion routine
        .END      START

```

25. Non-wait terminal output

It is now possible to use the .TTOUTR EMT to transmit characters to the terminal with the carry-flag being set on return if the terminal output buffer is full. The following conditions must be true to enable non-wait terminal output:

1. The .TTOUTR EMT must be used rather than .TTYOUT.
2. The instruction following .TTOUTR must not be [BCS .-2].
3. Bit 6 must be set in the job status word.
4. The program must be run in single-character activation mode or with the NOWAIT attribute set.

26. Miscellaneous changes

- 26.1 In previous versions of TSX-Plus, full access to all files was provided during the execution of a start-up command file. This access bypassed the effect of ACCESS commands which were in the start-up command file. With version 6.0, file access restrictions apply within start-up command files. If, within the start-up command file, you wish to access some files which are later to be access restricted, place the ACCESS commands after the part of the command file which performs the file accesses.
- 26.2 The maximum number of devices that may be specified with ACCESS commands has been increased from 25 to 30.
- 26.3 Previously you were allowed to list the directory of a device if you had access to any files on the device. Now you must have full access to the device in order to list its directory.
- 26.4 A new column of information has been added to the SHOW DEVICES display. This column, labeled "Active I/O", shows the number of I/O operations which have been queued to the device handler but not yet completed. This count is the total for all units supported by the device handler. The count can be useful for checking for "hung" devices.
- 26.5 Two new options have been added to the SET CLn command. The GRAPH option is equivalent to WIDTH=0; NOGRAPH is equivalent to WIDTH=132.
- 26.6 The period character (.) can now be used with program debugger commands to denote the current value of the program counter. For example, the command:

.;1B

would set breakpoint 1 at the current program counter location.

- 26.7 The processing of .DEVICE reset lists for real-time programs has been changed. If a cell modified by the .DEVICE list corresponds to an interrupt vector that is connected to the job, the system no longer connects that vector to the usual unexpected interrupt routine during job cleanup.

- 26.8 The spooler form mount message ("Mount 'xxxxxxx' form on ddn") has been changed to include the device unit number with the device name.
- 26.9 A SET LD EMPTY command may be used to dismount all logical disks for the current job. All logical device names that are assigned to logical disks are also deassigned.
- 26.10 A /TRANSPARENT option may be specified with the RUN or R commands to cause the program being started to be executed without its terminal output in "transparent" mode. This prevents the system from interpreting output control characters such as the TSX-Plus "leadin" character.
- 26.11 The terminal input activation mode which activates on receipt of escape sequences from the terminal (turned on by the leadin-B program control sequence), now correctly handles VT200 multi-character control sequences.
- 26.12 A change has been made to the way a job is scheduled for execution when a shared-file locked block is unlocked and a job is waiting for the block. The waiting job is now given a priority boost to give it a chance to lock the block before the unlocking job has a chance to lock the block again. This reduces the tendency of one job which is rapidly locking and unlocking blocks from unduly hindering the execution of other jobs that need the same blocks.
- 26.13 The values programs return in location 53 to indicate error severity levels have been changed for the SEVERE and FATAL error classes. The new values are as shown below:
- | | |
|---------------|----|
| Success | 1 |
| Warning | 2 |
| Error | 4 |
| Severe | 10 |
| Fatal | 10 |
| Unconditional | 20 |
- 26.14 Processing of control-U has been substantially speeded up in the case where SL is set OFF and a large number of characters are in the terminal input buffer.
- 26.15 In order to implement end-of-file processing for the ENDPAGE option, the LS and LP handlers now set the non-RT-11-directory-structured bit in the .DSTATUS status word (bit 12 of word 1). This could require a change to some programs which require a file name to be specified as well as a device name if the device is directory structured (the system itself does not require a file name). The CL handler implements the ENDPAGE and ENDSTRING options without setting status bit 12.

- 26.16 A SHOW SL keyboard command has been added to display the options in effect for the single-line editor.
- 26.17 It is now possible to determine if program input is coming from a command file or directly from the terminal. If the program is being run from a command file, bits 8, 12, and 15 will be set in fixed monitor offset location 366. This status word can be accessed with an instruction of the form:
- .GVAL #AREA, #366
- If the program is not being run from a command file, all of these bits will be reset (0).
- 26.18 TSX-Plus now checks during initialization to make sure the computer line time clock is running. If the clock is not running, the initialization is aborted with the message: "Computer line time clock (50 or 60 Hz) is not working".
- 26.19 TSX-Plus will now buffer I/O request above 248 Kb for devices which use mapped I/O buffers (MAPIO attribute specified for their TSGEN DEVDEF declaration). I/O requests were previously mapped only when the read or write buffer was above 256 Kb.
- 26.20 You must now specify a device name with the INITIALIZE and SQUEEZE commands. You may not omit the device name and have the system prompt you for it as it did before.
- 26.21 The keyboard command scanner now recognizes quoted strings (apostrophe or quote characters can enclose the string). Characters such as "@" and "!" which normally have significance within command strings are treated as ordinary characters if they are within a quoted string.
- 26.22 A character sequence of "@@" (two consecutive at-signs) within a command line argument list is now translated to a single at-sign rather than being interpreted as a within-line command file reference.
- 26.23 Device handlers provided on the distribution are RT-11 version 5.02 compatible. Two DM handlers are provided. DM22 performs 22-bit QBUS I/O for the Dilog DQ215 and the Emulex SC02C. DM18 performs 18-bit DMA I/O for DEC and other type controllers. A standard delay following each subsystem clear operation is included for some non-DEC controllers. All device handler SLP files have been reviewed and revised as necessary.
- 26.24 The default attributes for the MS handler are DMA, MAPH, and HANBUF. Thus, by default, the MS handler will now load as a mapped device handler with the normal exceptions for a handler which has an internal buffer (i.e., extended UNIBUS machines and in combination with the MAPIO option).

- 26.25 The SET SL terminal-type commands are no longer implemented. The current terminal type setting is assumed.
- 26.26 Because the named message channel system and data regions are constrained to an 8 Kb region, the message system may be truncated if generated very large. SHOW commands are now available to determine the effective values for MAXMSG, MAXMC and MAXMRB.

Corrected problems

27. The following corrections are included in TSX-Plus version 6.0:
- 27.1 When a .CLOSE is performed to an I/O channel that is opened to a CL device, or a SET CL LINE=*n* command or EMT is executed, a check is made to see if any output is currently pending in the output ring buffer for the CL unit. If there is pending output, the job is suspended until the output is completed or until a maximum of 20 seconds of time have elapsed. This prevents truncation of CL output issued just prior to disconnecting the CL unit from a line.
- 27.2 TSX-Plus would lose the first character typed after initiating a logon after the terminal had been turned off or a break character was transmitted to the system. This problem only occurred on lines which were not generated with autobaud speed setting.
- 27.3 A job would lose the memory mapping to PLAS regions when a shared run-time system was disassociated from the job.
- 27.4 TSX-Plus would generate XON/XOFF flow control characters on lines connected to DHV11 multiplexers even if the line was in use as a CL line with the BININ/BINOUT flags set.
- 27.5 Parity checking and generation could not be enabled for lines connected to DHV11 multiplexers.
- 27.6 The .CSIGEN EMT has been corrected. Previously, if the command line was provided by the program (rather than being prompted for by .CSIGEN), and an error was detected such as a specified device or file not existing, the .CSIGEN EMT purged all channels before returning the error code. Now the .CSIGEN does not purge the channels, so you can tell which file was in error by checking the channel status. This change makes the TSX-Plus .CSIGEN processing comply with that done by RT-11.
- 27.7 The SET LOG CLEAN command now correctly resets the highest-block-written information for the log file.
- 27.8 The keyboard monitor information message "Device is still mounted by other users" was displayed inappropriately when dismounting a logical disk.
- 27.9 The highest-block-written value for a file was incorrectly set if an invalid block number was specified with a write operation and the word count was zero. A write with a word count of zero now performs no operation.

- 27.10 The open of the spool file used to fail during TSX-Plus startup if the RT-11 device handler was not loaded. Now the initialization code fetches the handler before trying to open the spool file.
- 27.11 Register 0 (R0) was destroyed if a BPT instruction was used in a program to cause the TSX-Plus debugger to be started. This problem only occurred when the BPT instruction was executed before the debugger had been initialized (i.e., it didn't occur when BPT was executed as a result of having set breakpoints with the debugger).
- 27.12 A system crash could occur if data caching was enabled for a device at the time that an I/O request for the device was waiting for a free UNIBUS map register. This problem could only occur on UNIBUS systems with more than 256 Kb of memory.
- 27.13 A .CHAIN EMT no longer calls device handler abort entry points.
- 27.14 A problem in TSAUTH has been corrected which prevented the KILL command from removing all selected accounts when a wildcard was specified.
- 27.15 The .SPCPS EMT did not work.
- 27.16 A user defined command of the form:

XXX:==R DIR DK:\R DIR DK:

only executed DIR one time (the second part of the command was ignored).
- 27.17 An indirect command file reference within a command line was not expanded if UCL was set FIRST.
- 27.18 The EMT to send a message to another line did not correctly truncate the message if it was too long.
- 27.19 A system hang could occur if a terminal read timeout occurred at the same time that a character was received from the terminal.
- 27.20 The .CSISPC EMT to parse a file specification allowed two dots in the file name.
- 27.21 CCL version 3.00L corrects several problems. The /DELETE and /REMOVE library command switches now generate an output file (with the same name as the input file) when only one file is specified on the command line. Several command switches (/LIST, /MAP, /PRINT, etc.) no longer interfere with the positioning of other switches (/BEGIN, /PASS, etc.) in the expanded command line. MACRO/CROSSREFERENCE (or /CREF) switch does not insert a cross reference file name unless one is specified. The /NOQUERY switch on the INITIALIZE and SQUEEZE command suppresses the warning message "CCL-W-This command may interfere with other users." The EXECUTE command will not expand if one of the files specified cannot be located.

TSX-Plus Version 6.0 Release Notes

- 27.22 A correction was made to prevent a log file from continuing to write onto the device when a USR error occurred during closing of the log file.
- 27.23 Read and write operations with word counts of 0 (zero) are now ignored (treated as NOP's) rather than returning error codes.
- 27.24 The DL and DM handlers contain corrections for bad block replacement operations.
- 27.25 Mapped device handlers larger than 8 Kb will abort system startup.
- 27.26 Device vectors for the modem control portion of a DH11 multiplexer (DM11) are no longer required to be addressed as a multiple of 10 (octal).
- 27.27 Hardware flow control for DHV11 multiplexers is no longer enabled. BININ and BINOUT modes now behave the same on DH, DHV and DZ type multiplexers.
- 27.28 SYSMON changes.
 - 27.28.1 SYSMON was altered to display all three digits of the job priority in the job execution status display. The PRIV flag was removed from the terminal status display, in accordance with the new privilege structure.
 - 27.28.2 The SET SL SUBSTITUTE option was added to the terminal display screen.
 - 27.28.3 A zero length message no longer causes a trap in the message display screen.
 - 27.28.4 The message display screen now requires either OPER or SYSPRV privilege to run (in addition to MEMMAP). You may install SYSMON with OPER or SYSPRV in addition to MEMMAP to bypass this restriction if desired.
 - 27.28.5 The CL line display has been modified to display up to 16 lines, CL0 - CL7 and C10 - C17.
 - 27.28.6 The third terminal status line is now correctly cleared in the terminal status display.
 - 27.28.7 The CL line display now correctly displays parity. Even and odd parity were reversed.

INDEX

- .DEVICE, 50
- .DSTATUS, 51
- .TTOUTR, 49
- @-sign
 - Double, 52
- ACCESS command, 50
 - Directories, 50
 - Number of devices, 50
- Account authorization, 9
- At-sign
 - Double, 52
- CL
 - .SPFUN to reset, 37
 - 16 units, 1
 - Cross-connection, 28
- CL units
 - RESET option, 37
- Command file
 - Determining if in, 52
 - Startup, 9
 - Subprocess startup, 31
- Communication lines
 - 16 units, 1
- Completion routine
 - Terminal input, 48
- Compose key
 - Professional, 35
- Configuration utility
 - TSXMOD, 38
- Context
 - File EMT to obtain, 33
- Control-B
 - Printwindow, 20
- Crash dump, 46
- Cross-connection, 28
 - Releasing, 29
 - Special characters, 29
- Debugger
 - Program counter, 50
- DEFINE KEY command, 24
 - Qualifiers, 26
- Defined keys, 24
 - Command files, 27
 - Control characters, 25, 27
 - Deleting definition, 25
 - Key names, 25
 - Redefining letter, 27
 - Showing, 28
 - Subprocess switch, 27
- DETACH command
 - Parameters, 31
- Detached jobs
 - Inheritance, 30
- Device handlers
 - Version, 52
- Directories
 - ACCESS command, 50
- DM device handler, 52
- DTR
 - Cross-connection, 29
- ENDPAGE
 - .SPFUN to specify, 36
 - CL SET option, 36
- ENDSTARTUP SET option, 37
- ENDSTRING
 - .SPFUN to specify, 36
 - CL SET option, 36
- Error levels, 51
- Escape sequences
 - VT200, 51
- File context
 - EMT to obtain, 33
- Forms request
 - Unit number, 51
- GIDIS
 - Professional console, 35
- Global named regions, 47
- GRAPH
 - SET CL command option, 50
- Graphics
 - Professional, 35
- Greeting message, 16
- Handlers
 - Device, version, 52
- I/O Queue element
 - Job number, 1
- Inheritance
 - Detached jobs, 30
 - Subprocesses, 30
- INITIALIZE command, 52
- INSTALL command
 - ADD, 13
 - DELETE, 14
 - Device names, 13
 - Wildcards, 13
- Installed command files, 13
- Installed programs, 13
 - Attributes, 15
 - Privileges, 15
 - SHOW command, 14
- Interconnect
 - CL, See cross-connection7

- Job name
 - Setting, 6
- Jobs
 - Number of, 1
- Keyboard commands
 - Quoted strings, 52
- Keys
 - User defined, 24
- Laser printer
 - SET CL ENDPAGE, 36
- Line number
 - SEND command, 30
- Line time clock, 52
- Local named regions, 47
- Logon
 - Pre-, password, 16
- LOGON program, 2
- LP handler, 51
- LS handler, 51
- MAPIO, 52
- Message channels
 - Effective size, 53
- MS handler, 52
- Name
 - Setting job, 6
- Named regions
 - Global, 47
 - Local, 47
 - Removing, 48
 - Showing, 48
- NOGRAPH
 - SET CL command option, 50
- Parent
 - Detached job, 31
 - Subprocess, 31
- Password
 - Pre-logon, 16
 - System, 16
 - User, 9
- Patcher
 - TSXMOD, 38
- PLAS regions
 - Named, 47
- PPN, 9
- Pre-logon password, 16
- Priority
 - Maximum authorized, 9
 - Setting, 6
- Privileges, 2
 - Authorizing, 7, 10
 - Displaying authorized, 7
 - Displaying current, 7
 - EMT to change, 8
 - Existing accounts, 7, 13

- Setting, 6
- Process name
 - Setting, 6
- Process Windowing, 17
- Professional
 - Quad-SLU, 35
 - Professional console, 35
 - VT100 emulation, 28
- Program debugger
 - Program counter, 50
- Project, programmer number, 9
- Q.JNUM, 1
- Q.JOB, 1
- Quad Serial Line Unit, 35
- Queue element
 - Job number, 1
- Quoted strings
 - Keyboard commands, 52
- Record locking, 1
- REMOVE command, 48
- Resetting CL units, 37
- RESUME command, 34
- Resuming jobs, 6
- RTSORT
 - /GETCXT switch, 34
 - As detached job, 34
 - File context, 34
- R[UN] command
 - /TRANSPARENT switch, 51
- SEND command
 - Changes, 30
 - Window interaction, 19
- SET CL command
 - ENDPAGE, 36
 - ENDSTRING, 36
 - RESET, 37
 - [NO]GRAPH, 50
- SET ENDSTARTUP command, 37
- SET HOST command, 28
- SET LD command
 - EMPTY, 51
- SET PRINTWINDOW command, 20
- SET PROCESS command, 6
- SET SL command
 - Terminal type, 53
- SET SUBPROCESS command, 31
- SET WINDOW command, 18
- SETUP
 - Professional console, 35
- Shared files, 1
- SHOW command
 - Message channels, 53
- SHOW DEVICE
 - Active I/O, 50

- SHOW INSTALL command, 14
- SHOW KEYS command, 28
- SHOW PRIVILEGE command, 7
- SHOW REGION command, 48
- SHOW SL command, 52
- Single line editor
 - Defined keys, 26
 - Substitution, 26
- SQUEEZE command, 52
- Startup command file, 9, 31
 - Access, 50
 - Ending, 37
- Subprocess
 - Defined key switch, 27
 - Defined keys, inheriting, 28
 - Inheritance, 30
 - Startup command file, 31
 - Window, 19
- SUSPEND command, 34
- Suspending jobs, 6
- Swap file size, 47
- System crash dump option, 46
- System I/O mapping, 52
- System password, 16
 - Setting, 16, 17
- System size, 1
- Terminal input
 - Completion routine, 48
- Terminal output
 - Non-wait, 49
- TSAUTH, 2
 - Account statistics, 11
 - Authorizing account, 9
 - Changes, 9
 - Charge file, 12
 - Indirect files, 12
 - Killing account, 11
 - Listing account, 11
 - Modifying account, 10
 - New program, 7
 - Resetting statistics, 12
- TSGEN
 - DETACH macro parameters, 31
 - DMPKTP parameter, 46
 - DMPTCR parameter, 46
 - FLAG macro, 16
 - KEYMAX parameter, 28
 - MAXWIN parameter, 17
 - NUIP parameter, 14
 - SWPSLT parameter, 47
 - SYSDMP parameter, 46
 - SYSPS macro, 16
- TSXMOD, 38
- TTOUTR request, 49
- Unmapped size of TSX, 1
- User defined keys, 24
- Virtual lines
 - Changes, 30
 - See subprocess
- VT200 escape sequences, 51
- Window
 - Clearing, 19
 - EMT to create, 22
 - EMT to delete, 23
 - EMT to print, 24
 - EMT to resume, 24
 - EMT to select, 23
 - EMT to suspend, 24
 - Printing, 19
 - Refreshing, 19
 - SEND messages, 19
 - Subprocess, 19
 - WINPRT program, 21

1. SET TERMINAL START command

A command of the form:

SET TERMINAL n START

can be used to initiate a time-sharing line. OPER and TERMINAL privilege are required to use this command. If the line being started is set for autobaud speed recognition, it defaults to 9600 baud as it is started. If the terminal is known to be set to another speed, combine the speed setting with the START option to avoid speed mismatch errors. For example:

SET TERMINAL 7 SPEED=19200,START

2. SET [NO]SHUTDOWN command

A SET [NO]SHUTDOWN command has been added to start or cancel a system shutdown. OPER privilege is required to use this command. The SET SHUTDOWN command is equivalent to the old \$SHUTDOWN command. The SET NOSHUTDOWN command cancels a previous SET SHUTDOWN or \$SHUTDOWN command.

3. .GVAL to get system version number

A .GVAL EMT with an offset value of -32 (decimal) can be used to get the TSX-Plus system version number. The version number is expressed as the version times 100, thus version 6.0 returns a version number of 600 (decimal).

4. SHOW VERSION command

A SHOW VERSION keyboard command has been added to display the current TSX-Plus system version number.

5. Access to .SYS and .TSX files

No privilege is now required to access files with the extensions .SYS and .TSX unless the files are located on the system (SY, boot) device. SYSPRV or BYPASS privilege is required to access .SYS and .TSX files on the system device.

* Software Problem Report *

Company name:
Address:

Mail to: S&H Computer Systems, Inc.
1027 17th. Avenue, South
Nashville, Tennessee 37212-2299
USA

Submitted by:

Attn: Technical Support

Phone:

Phone: (615) 327-3670

Distributor:

Telex: 786577 S AND H UD

Date:

Easylink: 62-365020

TSX-Plus version and license number

Cobol-Plus version and license number

Other product name, version and license number

System disk handler, controller make and type

RT-11 Version

Report classification:

☐ Severe Error
☐ Moderate Error
☐ Minor Error
☐ Documentation Problem

Suggestion

☐ Inquiry

☐ Comment

☐ Other

Problem description / Comment:

All submissions become the property of S&H Computer Systems, Inc.

S&H Response: Received: _____ Verified: _____ Resolved: _____