TSX–Plus Version 6.50

Release Notes

These release notes describe the differences between TSX–Plus version 6.50 and version 6.40. The *TSX–Plus Documentation Set* incorporates the information described in earlier release notes. In case of any differences between the release notes and the manuals, the release notes take precedence.

27 Mar, 1991

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.

1 New Features and Changes

1.1 RT-11 V05.05 Compatibility

Most new features available in RT-11 V05.05 which affect TSX-Plus sites are supported in TSX-Plus V6.50. By category:

- Keyboard Commands

  The BACKUP command has been extended to support the new options /DIRECTORY, /INITIALIZE, and /SAVESET.

  The EDIT command has been extended to support initialization files and journaling with the new KED editor. The new options are /COMMAND, /JOURNAL, and /RECOVER.

  The FORTRAN command has *not* been updated to support F77 specific options.

- Utilities

  The new KED editor is supported, including initialization files and journal (and restore) files.

- Device Handlers

  Device handlers were not updated. The base version of RT-11 from which device handlers were derived remains RT-11 V05.04C. Most significantly, this means that extended unit support for DU is not included. For sites with MSCP devices larger than 256Mb, contact S&H for information on "pass through" or "front end" handlers which allow you to fully utilize large MSCP devices.

  The AT and UB device handlers and the extended memory services $JBREL, $MPMEM, and XDEALC were not implemented since extended Unibus support already existed in TSX-Plus.

- System Services

  The .CLOSZ system service was implemented in TSX-Plus V6.40. The .GFDAT, .GFINF, .GFSTA, .SFINF, and .SFSTA services are included in this version. The .CALLK service was not implemented.

  The fixed offsets in simulated RMON are now implemented as high as offset 504(8). As with previously implemented offsets, not all values are meaningful and may not be supported. The most significant new offsets are: $PROGD which tracks the SET EDIT command, $PROGF which always indicates FORTRAN IV, $SLOT2 which indicates the total number of device handlers allowed, and the bit in $CNFG3 which indicates if any devices use DMA.

- Miscellaneous Features

  Extended Unibus support has existed in TSX-Plus since V4.1, so the new AT and UB handlers are not required.

  Dates after 2003 are now supported. See the description of the DATE command below for more information on extended dates. Note that some utilities, such as DIR, do not yet recognize dates past 2003.

1.2 Keyboard Commands

1.2.1 DATE Command

The DATE command has been enhanced to accept and display dates up to the year 2099. When supplying a value to set the date, the year may either be specified as a two-digit number from 72 to 99, or as a 4-digit number from 1972 to 2099. Similarly, when the keyboard monitor displays the date, it displays dates through 1999 as a two-digit year (e.g. 26-Sep-90) and displays dates after 1999 as a four-digit year (e.g. 4-Jul-2076).

System services to set and retrieve dates (.SDTTM and .DATE) and automatic date rollover (at midnight) now support dates through 2099.

1.2.2 SET TT or CL NAME Command

A new SET TT NAME command has been added to change the name of a terminal line. The form of the command is:

```
SET TT n NAME=name-string
```

where *n* is the terminal line number and *name-string* is the name to which the terminal name is to be changed.

A new SET CL*n* NAME command has been added to change the name of a communication line. The form of the command is:

```
SET CLn NAME=name-string
```

where *n* is the communication line number and *name-string* is the name to which the CL unit name is to be changed.

1.3 System Services (EMTs)

1.3.1 User PAR control

This is an unsupported feature!

An EMT has been added which allows PAR and PDR values that have been modified by a user job to be re-established after a context switch. User manipulation of memory management registers is highly discouraged and we explicitly disclaim responsibility for any and all damage to the system which may result from such actions. One possible rationale for using such techniques is the need to update a large bank of shared or dual–ported memory (such as a graphics board) in the fastest possible way. Jobs mapped to the I/O page have always been able to do this, but previously job context switching had to be disabled so that the system did not re-establish its own idea of the appropriate contents of the PAR and PDR registers on returning to execution of the job which managed its own PAR registers. Now, if this request has been executed, then on switching context away from the critical job, the current contents of the hardware user memory management registers is saved in the job context block and restored from it on switching back to it.

The form of the EMT to allow user memory management control is:

```
        EMT     375
```

with R0 pointing to an argument block of the form:

```
        .BYTE   10,143
```

The pre-requisites to issue this request are that the job be locked in memory (either in place or in low memory) and that the job be mapped to the I/O page. After this request has been issued, subsequently unlocking the job in memory or unmapping the I/O page will disable user PAR control. Any request which causes the system to re-calculate memory mapping for the job will invalidate any current user changes to the mapping registers. Examples of requests which will cause such side effects are: re-sizing the job (EMT 375, function 0,141), and unmapping a PLAS window.

The errors returned by this EMT are:

Error Code	*Meaning*
0	Job is not locked in memory
1	Job is not mapped to the I/O page

1.3.2 I- and D-space with Overlays

In the previous version, when a job had enabled separate I– and D–space, all I/O was directed to D–space. This prevented the use of overlayed programs with separate I– and D–space since program overlays need to be mapped into I–space. A new EMT has been added which enables separate I– and D–space, but which sets a flag that allows overlays to be read into I–space. To be specific, when separate I– and D–space is enabled in this mode, then all .READ operations on channel 17 (octal) are directed to I–space. I/O operations on any other channel, writes to channel 17, or .SPFUN operations on channel 17 are not affected and are still directed to D–space.

The form of the EMT which enables separate I– and D–space, except that reads on channel 17 are directed to I–space, is:

```
        EMT     375
```

with R0 pointing to an argument block of the form:

```
.BYTE   7,143
```

As with the similar 4,143 function, separate I– and D–space remains in effect until explicitly disabled (with function 5,143) or until the program chains or exits. Errors are the same as for function 4,143.

This solves the disk input problem so that overlays are read into the correct address space, but it does leave an efficiency problem. Before reading a new overlay from disk, the overlay handler first checks the first word of the overlay region to determine if the needed overlay is already in memory. When it does this, the memory reference is automatically directed by the CPU to D-space, which is incorrect when checking a code region. To alleviate this problem, a new EMT has been added which allows a program to obtain a word value from I-space. This allows the overlay handler to be modified and correctly determine if the desired overlay segment is already in memory.

The form of the EMT to obtain a word value from I-space is:

```
EMT     375
```

with R0 pointing to an argument block of the form:

```
.BYTE   0,165
.WORD   <I-space address>
```

The value at the specified I-space address is returned in R0.

For efficiency sake: this EMT is implemented in the primary TSX-Plus EMT processor rather than as part of the I- and D-space series (n,143); no validity checking is done on the address value specified; and no errors are returned. If the specified I-space address is not currently mapped, then the program will abort with an error message similar to:

```
?MON-F-Kernel mode trap within TSX-Plus
Abort location = 130370   Overlay = EM2
```

The exact form of the error message and the abort location may vary in future versions.

Details of modifying the overlay handler and substituting the modified overlay code into your programs are left to the user.

1.3.3 EMT to get PSW

A real-time EMT has been added to return the current job processor status word (PSW). Because of the variety of ways access to the PSW has been implemented in different PDP–11 processors, the current value is normally obtained with the .MFPS macro, which calls a subroutine in the resident monitor to obtain the value. Under TSX-Plus this monitor routine always returns the value 0 (current mode *user* , previous mode *user* , general register set 0, priority 0, and all condition code and trace bits clear), which would be the normal case for user jobs. Note that .MFPS will fail with a trap to 4 when the job is not currently mapped to simulated RMON (e.g. virtual jobs and jobs which have mapped PAR 7 to the I/O page or to a shared run-time). Sometimes, it is desireable to obtain the actual PSW value (for example, when experimenting with supervisor mode). The following request may be used to obtain the current PSW value:

```
        EMT     375
```

On entry, R0 should point to an argument block of the form:

```
        .BYTE   23,140
```

The current value of the PSW, as pushed onto the stack by the EMT instruction, is returned in R0. This request returns no errors and will work even if the job is not mapped to RMON.

1.3.4 EMT to get License Number

An additional function has been added to the site information EMT to return the full TSX–Plus site license number as a string. Unlike the sub–functions to return the incremental license number and the site name string, this sub–function does not require inclusion of printer spooling during system generation.

The form of the EMT to return the site license number string is:

```
        EMT     375
```

with R0 pointing to an argument block of the form:

```
        .BYTE   2,124
        .WORD   buff-ptr
```

where *buff-ptr* is the address of a **word-aligned** buffer to hold the returned license number string. The buffer should be at least 26 bytes long. The actual license number string length is returned in R0. If carry is set on return from this EMT, then the buffer address was invalid.

1.3.5 EMT to set Line Names

An EMT has been added to allow the user to get or change the name of a terminal line from within a program. The form of the EMT is as follows:

```
        EMT     375
```

with R0 pointing to an argument block of the form:

```
.word  func,147
.word  buff-addr
.word  line-num
```

where *func* is 4 to get the terminal name and 5 to set the terminal name, *buff-addr* is the address of the buffer to contain the terminal name and *line-num* is the terminal number.

The following error code can be returned from this EMT

Error Code	*Meaning*
3	The name string was too long

1.3.6 EMT to Suspend Output

A new EMT has been added which emulates the behavior of pressing a Control–O (^O). This EMT flushes the terminal output buffer and suppresses any further terminal output until the .RCTRLO (reset CTRL–O) EMT is issued. The form of the EMT is:

```
EMT     375
```

with R0 pointing to an argument block of the form:

```
.BYTE   2,120
```

No errors are returned by this EMT. Terminal output suppression remains in effect for the job until the .RCTRLO EMT is issued or until CTRL–O is received from the terminal.

1.4 Device Handlers

1.4.1 DM Handler

The DM handler has been modified to allow the Emulex SC02/C controller to work on faster processors (such as the 11/93). This controller, which is popular on Q-Bus systems because it supports 22–bit addressing, appears to have a latency in its registers after receiving certain controller commands. This controller register latency results in a "Fatal System Error, KTP–Kernel mode trap" with the "Arg. value" pointing to an address within the DM handler. This problem first appeared when 11/23 processors were replaced with (faster) 11/73's. At that time, a patch was developed for the DM handler which inserted some "No OPeration" (NOP) instructions after loading certain commands into the controller registers. With the advent of even faster processors, such as the 11/83 and 11/93, the delay after loading these controller registers needs to be lengthened even more.

To compensate for the variety of processor speeds, the delaying sequence of NOP's has been replaced by a call to a delay loop. The number of iterations of this loop can be controlled with a handler SET command. The handler option is named **DELAY** and accepts values from 0 to 127. For 11/23 processors, a value of 0 is appropriate. For 11/73 processors, a value of 1 is sufficient. The appropriate values for 11/83 and 11/93 processors have not yet been determined, but a value of 4 should be adequate. The default value is 1. The following example would set the number of iterations of the delay loop to 4 for use on a 11/93 processor:

```
SET DM DELAY=4
$STOP
```

Note that the SET DM DELAY=n command must be issued while running TSX and then the system should be rebooted. Also, note that device handler SET commands should *not* be issued in "start–up" command files. Handler SET commands only need to be issued one time and the value is recorded in the disk copy of the handler. Unless the device handler is replaced, the SET remains in effect and does not need to be reissued.

1.5 Miscellaneous Features

1.5.1 Size of unmapped TSX

In order to facilitate system generation of large TSX systems, when the TSGEN parameter INIABT is set to 1, the unmapped size of TSX is now displayed during system start-up just prior to loading unmapped device handlers. The highest address used by the unmapped portion of TSX is displayed at that point as an octal address. After adding to this the sizes of all unmapped device handlers, the result must be less than 120000(8), the 40Kb limit. (Note that a few additional words may be allocated in low memory after loading handlers, depending on selected system features such as handlers which are mapped high and those which use I/O mapping, so the high limit after loading low handlers may need to be slightly smaller than 120000(8).)

1.5.2 DUP Installed with SYSPRV Privilege

Starting with RT-11 V05.05, the DUP utility attempts to obtain information about logical subset disks by opening the SY:LD.SYS device handler image. Access to SY:*.TSX and SY:*.SYS files is restricted under TSX-Plus unless the user has SYSPRV or BYPASS privilege. This prevents non-privileged users from issuing otherwise legitimate commands, such as:

```
COPY/DEVICE LDO: DYO:
```

even if there are no other ACCESS restrictions. In order to allow this type of operation, the DUP utility is by default now installed with SYSPRV privilege.

If this is a problem for any site, the new default installed attributes can be overridden in a system start-up command file (specified by the DETACH macro in TSGEN). To return the DUP utility to its previous attributes, include the command:

```
    INSTALL ADD SY:DUP.SAV/DUP/PRIV=(NODEBUG)
```

1.5.3 SET RECALL REVERSE Inheritance

The current state of the SET RECALL [NO]REVERSE command is now inherited by a newly created subprocess.

1.5.4 National Language Support

Process Windowing has been enhanced to support National Replacement Character sets on the VT220 (and later) series terminals. A **/LANGUAGE=** switch has been added to the **SET WINDOW** command. This option sets the initial National Replacement Character set for the window, selects it as hard character set G0, and lock–shifts G0 into GL. The keyboard monitor accomplishes this by sending the appropriate terminal control sequences after the window has been created. Default language specification is *not* an option to the system service which creates a window (EMT 375, function 161, subfunction 0). Note that only a single national replacement character set is available at a time and this must correspond to the current keyboard setup selection; the language specific keyboard layout must be selected on the "Set–Up Directory" setup menu and "National" must be selected on the "General Set–Up" menu. Note that language and other character mappings are not preserved when initiating a new window, even when other characteristics are copied from an existing window; the appropriate terminal control sequence must be issued to initialize the character set mapping for a new window. When switching to a new subprocess by typing a "control–W n" sequence, language initialization can readily be handled with the appropriate **DISPLAY** command in a subprocess initialization file (see the **SET SUBPROCESS** command).

The following language designators are accepted for the **SET WINDOW/LANGUAGE=** command: **ASCII**, **BRITISH**, **DANISH**, **DUTCH**, **FINNISH**, **FRENCH**, **FRENCHCANADIAN**, **GERMAN**, **ISOLATIN1**, **ITALIAN**, **NORWEGIAN**, **SPANISH**, **SWEDISH**, **SWISS**, **UK**, and **US**. The minimum unique abbreviation is accepted. Note that **UK** is a synonym for **BRITISH** and **US** is a synonym for **ASCII**. **ISOLATIN1** is only available on VT320 and later terminals.

1.5.5 Autobauding at 2400

There is a conflict between the automatic baud rate determination for time-sharing lines at 2400 and 19200 baud. In previous versions the conflict was resolved in favor of the 19200 speed, making it very difficult to "autobaud" 2400 baud modems. However, with the increased availability of 2400 baud modems it is now more commonly desired to "autobaud" the modem. Therefore, the automatic rate determination is now resolved in favor of 2400 baud. This makes it difficult to "autobaud" terminals at 19200. Since terminal rates are rarely changed, time-sharing lines which are to be used with terminals set for 19200 baud should be explicitly defined as 19200 baud using the SPEED macro in TSGEN rather than "autobauded".

1.5.6 TSAUTH Password Control

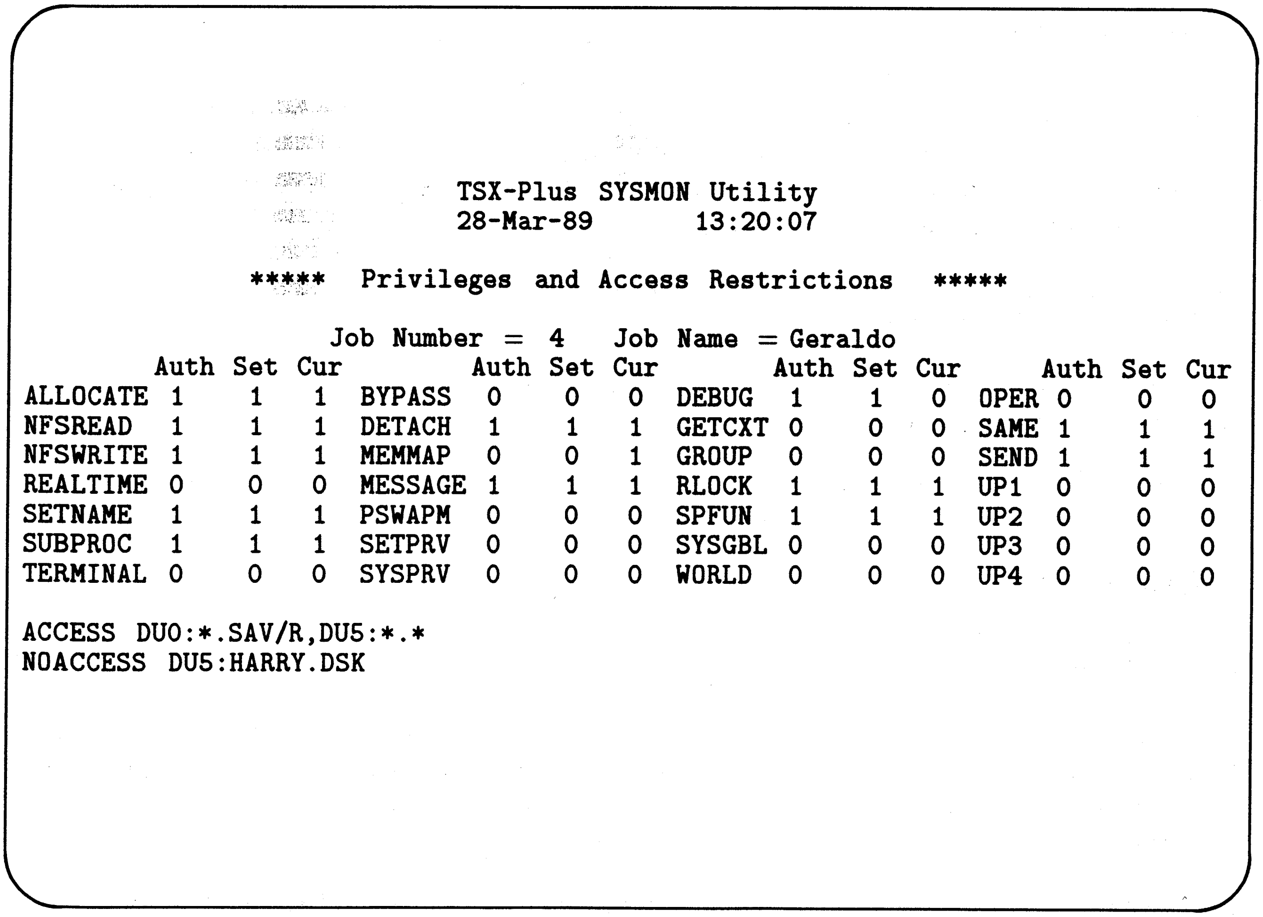
A new SETPASSWORD privilege has been added to TSAUTH. This privilege is internal to TSAUTH and LOGON. It is set in the same manner as any other privilege is set in TSAUTH. It is also displayed along with the other privileges when LISTing an account in TSAUTH. However, it is not displayed when the SHOW PRIVILEGES keyboard monitor command is issued.

1.5.7 Locked Systems

The ability to "lock" a TSX-Plus distribution to a single machine is available. This was implemented at the request of some European distributors in locations where copyright laws are only regarded casually. Distributors can contact S&H for more details on locked systems.

1.6 SYSMON

1.6.1 SYSMON Privilege Display



```
                    TSX-Plus SYSMON Utility
                    28-Mar-89       13:20:07

          *****  Privileges and Access Restrictions   *****

                Job Number =  4   Job Name = Geraldo
        Auth Set  Cur         Auth Set  Cur         Auth  Set Cur        Auth Set Cur
ALLOCATE 1    1    1  BYPASS   0    0    0  DEBUG   1     1    0  OPER  0    0   0
NFSREAD  1    1    1  DETACH   1    1    1  GETCXT  0     0    0  SAME  1    1   1
NFSWRITE 1    1    1  MEMMAP   0    0    1  GROUP   0     0    0  SEND  1    1   1
REALTIME 0    0    0  MESSAGE  1    1    1  RLOCK   1     1    1  UP1   0    0   0
SETNAME  1    1    1  PSWAPM   0    0    0  SPFUN   1     1    1  UP2   0    0   0
SUBPROC  1    1    1  SETPRV   0    0    0  SYSGBL  0     0    0  UP3   0    0   0
TERMINAL 0    0    0  SYSPRV   0    0    0  WORLD   0     0    0  UP4   0    0   0

ACCESS DU0:*.SAV/R,DU5:*.*
NOACCESS DU5:HARRY.DSK
```

The SYSMON system monitoring utility has been enhanced by the addition of a privileges and access restrictions display. SYSMON can now be used to determine what privileges and access restrictions a particular job has.

1.6.2 SYSMON Miscellaneous Changes

- A SYSMON screen may now be pre-selected from the command line. For example:

  ```
  R SYSMON 9,5
  ```

  would cause screen 9 (the generalized data cache display) to be displayed with a 5 second refresh interval.

  If the selected display requires any input, this should also be entered on the command line. For example:

  ```
  R SYSMON 3,5,1
  ```

  would pre-select the terminal display with a 5 second refresh interval and terminal line 1 as the terminal for which the information is to be displayed.

  If no parameters are specified on the command line, SYSMON will function in *menu–mode* as before. If an invalid screen number is specified, an error message indicating this will be displayed. If the time (second) parameter is invalid or missing, the default (5 seconds) will be used. If the third parameter is invalid or missing, the default for that screen will be used.

- The SYSMON directory cache display now displays the values of the TSGEN parameters MAXCSH and NMFCSH.

- The SYSMON window status display now displays whether or not flag pages and date/time stamp are printed for a printwindow.

- The SYSMON window status display now displays job name for a particular window.

2 Corrected problems

2.1 Keyboard monitor

1. A problem was corrected which caused the system to hang when first started if certain terminals were specified both to be automatically started when the system was brought up and to require system password. The problem was unique to version 6.40 and did not occur in earlier versions.

2. The `MACRO` command improperly handled arguments to the `/CROSSREFERENCE` option.

3. The KILL command incorrectly required GROUP privilege to kill a job with the same PPN.

2.2 System service calls (EMTs)

1. A fatal system error SOF–Stack overflow with Arg. value = 3 could occur while opening a shared file if the TSGEN parameter MXLBLK was too large (more than about 35). Note, that this parameter indicates the maximum number of 512-byte blocks that may be locked within a single file by each program. It does not determine how many blocks can be locked in the file by several users. This parameter should normally indicate the size of the largest record in the file. For example, if the largest record to be locked in any file on the system contained 630 bytes, then this could potentially span 3 blocks (a few bytes at the end of one block, 512 bytes in a middle block, and the remainder of the 630 bytes in a third block). In this example the appropriate value for MXLBLK would be 3. Depending on the programming style, MXLBLK may need to be somewhat larger. For example, COBOL-Plus requires one additional block to be locked (the data file home block) besides enough to cover the largest record. Other coding styles may require MXLBLK to be twice as large as required to cover the largest record, for example, if a second record is to be locked before releasing the first.

2. The system service to kill a job (EMT 375, function 132, subfunction 2) incorrectly required DETACH privilege to kill the last primary job.

2.3 System internals

1. There was a problem with re-mapping of local PLAS regions during an in–swap operation. This was most commonly noticed as a "Trap-to-4" in the KEX editor after VTCOM was run on another line. Actually, it was possible for access to be disallowed to various portions of PLAS regions after re–positioning due to swapping.

2. It was possible to have a system crash in overlay CLO if a CL unit was disassociated from a time-sharing line while there was active input from the line.

3. The WINPRT program generated an incorrect time stamp on machines with a 50Hz line time clock.

4. Under some conditions, when the sequence of characters to log on a line was typed too rapidly, the echoed characters would get "out of sync" with the typed input. That is, typed input was not echoed until terminated and was not passed to the current program until a subsequent field was terminated. If the condition was noticed, it could be cleared by typing several control–C's. Occurrence of the problem was dependent on system activity, processor and terminal speed, and rate at which the logon sequence was entered. If KED was run while in this "out of sync" state, then it was very difficult to properly issue a command (especially EXIT). If the SET SL ON command was issued while in this state, system performance was dramatically reduced.

5. A problem was corrected which incorrectly granted access to .SYS and .TSX files on the system device to non-privileged jobs under certain circumstances.

6. When separate I– and D–space was enabled and symbolic code was displayed from a region for which the I– and D–space mapping were different, then immediate operands and offsets were incorrectly displayed from the corresponding D–space address.

2.4 CCL

1. The EDIT command has been corrected so that the /ALLOCATE switch now generates the proper /A:nnn CSI switch rather than specifying the output file size in square brackets.

3 Documentation Changes

1. The CL version number is assigned during system initialization, based on the booted RT-11 version, according to an internal table of RT-11 and the corresponding XL handler version numbers. This is done to insure correct operation of the VTCOM program. VTCOM checks the XL (or CL) handler version number by matching the high byte of the word returned by .SPFUN 204. This is assigned in XL.MAC by the $$$VER symbol. If TSX is started from a version of RT-11 which is not in its table of known versions, then the CL version number may not be correct and VTCOM will report an error. This can be easily remedied with the SET CL VERSION command. The quickest way to determine the correct version is to inspect the XL.MAC file for the $$$VER symbol. Alternatively, try changing the CL version number and using VTCOM. When the correct version is identified, put the appropriate SET CL VERSION command near the beginning of the first system start-up command file specified with the DETACH macro in TSGEN.

2. Shared files and locked records are not preserved across a chain, although the channels do remain open.

3. When controlling file access with the `NOACCESS` command, if no `ACCESS` commands have been issued and a logical subset disk is mounted, then most file accesses are denied. This can be avoided by explicitly adding a single `ACCESS` command to allow access to everything not otherwise restricted. For example if a startup command file did not contain any `ACCESS` commands, but did contain the following `NOACCESS` commands:

   ```
   NOACCESS DU1:PAYROL.DSK
   NOACCESS DU1:PRSNLL.DSK
   ```

   Then the job would have normal access to everything on the system except to the PAYROL and PRSNLL logical subset disks on DU1: and any files therein. This is normal and correct behavior. However, if the job mounts a logical disk to which it should have access, then any file which is not in that logical disk will be incorrectly forbidden.

   Proper behavior can be ensured by explicitly granting access to everything except the restricted files, as follows:

```
ACCESS *.*
NOACCESS DU1:PAYROL.DSK
NOACCESS DU1:PRSNLL.DSK
```