INTERFACE AND INTEGRITY FACILITY
DMERT/UNIX® RTR OPERATING SYSTEM
AT&T 3B20D COMPUTER

1. GENERAL INFORMATION

1.001 This addendum supplements AT&T Practice 254-341-120, Issue 3. Place this pink sheet ahead of Page 1 of the practice.

1.002 This addendum is being issued to cover enhancements made to the UNIX Level Automatic Restart Process (ULARP) by the craft interface recovery mechanism feature for UNIX RTR operating system, Release 6.

2. CHANGES

2.001 On Page 16, change paragraph 2.77 to read: If ULARP dies and is restarted, SIM does not terminate craft processes unless the restart is in response to a craft initialization request. SIM sends ULARP a startup message USTARTUP with a value USTART. USTART results in ULARP attempting to adopt its former child processes. If an adopt call returns failure, ULARP will attempt to restart that process.

2.002 On Page 16, add major heading E and the following paragraphs after paragraph 2.77:

E. UNIX Level Automatic Restart Process (UNIX RTR Release 6)

2.003 ULARP is a UNIX process whose functions are to create and monitor critical user processes and to automatically restart any monitored processes that terminate.

2.004 ***Initialization of ULARP:*** During a bootstrap procedure, SIM creates a process named /prc/unix. In the *pcreate* message sent to the process manager, the field pm_chan is set to BOOTCHAN (defined in the header file const.h), which indicates to the UNIX system *init* function that this process is a bootstrap procedure. The *init* function then executes the ULARP process. When SIM receives the acknowledgement message on the successful creation of the /prc/unix process, SIM logs the process ID and waits for an E_SENDMSG event (defined in the header file simdef.h) from ULARP. This event causes SIM to send ULARP a start-up message, with type defined as USTARTUP (defined in header file simmsg.h). This message contains SIMBOOT in the message text indicating a bootstrap procedure and other system information (type of boot) needed for a successful ULARP start-up.

2.005 The processes and run commands that ULARP must execute and monitor are defined in ULARP ECD records. ULARP executes and monitors all records that are marked process and executes and records exit status of all records marked command.

2.006 ***ULARP Failure Reports:*** Error conditions are reported via CSOP if CSOP is running. If CSOP is not running, ULARP reports errors to SIM with SINITSTAT messages, which result in the output of PRMs. All errors reported by ULARP are logged in the ULARPLOG log file and are documented on the REPT ULARP page of the Output Messages PDS manuals.

2.007 ***ULARP Termination:*** SIM receives a death-of-child message on the ULARP process if either the UNIX system "init" sequence fails to create ULARP or ULARP dies. When this message is received, SIM usually assumes that ULARP was prematurely terminated and attempts to restart it. In the event that SIM receives a death-of-child message on ULARP before it has received the E_SENDMSG event from ULARP, SIM displays a PRM to indicate the reason for ULARP failure before attempting a restart of ULARP.

2.008 ***Restarting ULARP:*** SIM will attempt to restart the ULARP process under the following three conditions:

- ULARP is prematurely terminated. If this occurs, SIM receives a death-of-child message and immediately attempts to restart ULARP.

- The maintenance operator enters the PDS/MML shell command INIT:ULARP! on the maintenance terminal. This causes the E_UINIT (defined in simdef.h) event to be sent to SIM. When SIM receives this event, it checks ULARP's status using ULARP pid. If ULARP is not running, ULARP attempts to restart ULARP. If ULARP is running, SIM sends the E_UINIT event to ULARP. When ULARP receives this event it rereads the ECD database records and attempts to start any child process which is not running and to rerun any run commands that did not complete successfully.

- The maintenance operator enters the CRAFT INIT sequence; default sequence (level 1) of CFT-INIT option (command code 15) without an application parameter set (command code 42) or the sequence in which the level of craft initialization is determined by first setting the application parameter (command code 42) and then entering the CFT-INIT option (command code 15). The CFT-INIT option causes the craft interface handler to send SIM the fault FLT-CFT. Upon receiving this fault, SIM suspends processes dependent on the initialization level, issues a request for a SURE KILL of these suspended processes, and kills ULARP if it is alive. After waiting for the I/O driver to pcreate the device processes, SIM attempts to restart ULARP and, upon receiving the E_SENDMSG event from ULARP, issues a message to ULARP indicating CFTSTART. The CFTSTART message informs ULARP that it was restarted in a craft initialization. ULARP will search the ECD records and start all processes that are not running and adopt those processes that are currently running. After completing its search through the ECD records, ULARP sends an E_CFTCOMP event to SIM indicating that the craft initialization is completed.

2.009 If ULARP dies and is restarted, SIM does not terminate craft processes unless the restart is in response to a craft initialization request. SIM sends ULARP a startup message of type USTART. USTART messages result in ULARP attempting to adopt its former child processes. If an adopt call returns failure, ULARP will attempt to kill and restart the process.

2.010 On Page 18, add the following paragraph following paragraph 2.94:

2.011 ***Special Audits:*** There is a special set of audits run by SIM. This set of audits is contained in a data structure called *Alist*. Each audit contained in *Alist* can belong to one or more sets. During a recovery action, SIM will run the set of audits in *Alist* that belong to that recovery action. For example, during craft init, SIM will run all audits in *Alist* that belong to the craft init set. There is only a craft init set of special audits in *Alist*.

2.012 On Page 19, delete the last sentence in paragraph 2.97.

AT&T — PROPRIETARY
Use pursuant to Company instructions