

**RECENT CHANGE PROCEDURE AND DATA BASE  
GENERATION FOR E911 PERIPHERAL  
DATA STORAGE PROCESSOR (PDSP)  
2-WIRE NO. 1 ELECTRONIC SWITCHING SYSTEM**

CONTENTS	PAGE	CONTENTS	PAGE
1. GENERAL . . . . .	2	E. RCCNTL Mode to Limit Table Growth . . . . .	16
2. OVERALL DESCRIPTION . . . . .	2	F. RCCNTL Mode to Override ESN Check . . . . .	16
A. Function . . . . .	2	G. Recommended Set of RCCNTL Modes . . . . .	16
B. Hardware Perspective . . . . .	2	H. RCCNTL Message Description . . . . .	17
C. Software Perspective . . . . .	3	6. PROMATS TAPE OF RC:TNESN MESSAGES . . . . .	18
D. User Perspective . . . . .	3	A. Process Overview . . . . .	18
3. DATA BASE DESCRIPTION . . . . .	3	B. Process Control Structure . . . . .	19
A. File Description . . . . .	3	C. Process Control Commands . . . . .	20
B. Storage Allocation . . . . .	3	D. PROMATS Tape Format . . . . .	21
C. Data Base Backup . . . . .	3	E. Process Control TTY Commands . . . . .	22
4. RC MESSAGE INFORMATION . . . . .	4	F. Output Messages . . . . .	25
A. Definitions . . . . .	4	7. TTY INTERACTIVE DEVICE . . . . .	27
B. RC Identification and Action Field . . . . .	6	8. OUTPUT MESSAGES . . . . .	27
5. 911 RECENT CHANGE MESSAGES . . . . .	12	A. OP:TNESN Messages . . . . .	27
A. RC:TNESN Message Description . . . . .	12	9. VERIFICATION AND STATISTICS MESSAGES . . . . .	29
B. RCCNTL Mode Message . . . . .	13	A. TNESN Verify . . . . .	29
C. RCCNTL Modes for PROMATS . . . . .	14		
D. RCCNTL Modes Used During Comparison Runs . . . . .	15		

**NOTICE**

Not for use or disclosure outside the  
Bell System except under written agreement

CONTENTS	PAGE
B. RCCNTL Verify . . . . .	30
C. ESN Statistics Message . . . . .	30
D. MAXESNS Statistics Message . . . . .	31
E. NOG Statistics Message . . . . .	32
10. DISTRIBUTING DATA BASE . . . . .	33
11. SYSTEM AUDITS . . . . .	34
12. GLOSSARY . . . . .	35

**1. GENERAL**

**1.01** This section covers the recent change (RC) data base generation procedures and data base description associated with a peripheral data storage processor (PDSP) to be used with expanded 911 system (E911). The E911 system can be used with 1E5 and later generics.

**1.02** This section is reissued to include the 2A generic, RC:TNESN, RC:PDTABL, and OP:TNESN Error Tables.

**1.03** The programmable magnetic tape system (PROMATS) mass input device is used for RCs; however, other input devices may be needed such as the Dataspeed 40. This section covers both types of RC inputs. The present speed of PROMATS RC input is about 16,000-18,000 RCs per hour.

**1.04** Abbreviations used in this section are as follows:

- BCD — Binary Coded Decimal
- DMS — Data Management System
- DS40 — Dataspeed 40 (input output device which consists of a high speed printer and CRT display)
- EDDL-ESS — Data Definition Language
- EOS — Extended Operating System
- EOT — End of Tape

- EPL-ESS — Programming Language
- EPSCS — Enhanced Private Switched Communication Service
- ESB — Emergency Service Bureau
- ESN — Emergency Services Number
- FCLD — Flash the Calling Line Display
- NPA — Numbering Plan Area or Area Code
- NPD — Numbering Plan Digit
- PDSP — Peripheral Data Storage Processor
- PIU — Processor Interface Unit
- PROMATS — Programmable Magnetic Tape System
- TDC — Tape DATA Cartridge.

**2. OVERALL DESCRIPTION**

**A. Function**

**2.01** The selective routing option of an expanded 911 system routes 911 calls to the proper answering point based on the calling telephone number (TN). One system may serve over one million customers. The data base system used allows a PDSP to be coupled to a No. 1 ESS.

**B. Hardware Perspective**

**2.02** The PDSP is connected to the No. 1 ESS via the PIU (Processor Interface Unit). The PDSP consists of a duplex configuration of the 3ACC, PIU, Tape Data Cartridge (TDC), and all associated peripherals, and a simplex configuration of the PROMATS (Programmable Magnetic Tape System) and Dataspeed 40 (DS40). The I/O channel used by the DS40 has 4 ports; any port can be used with a data set to accommodate a remote location. This I/O channel operates in what is sometimes called an ECHOPLEX mode, so that inputs from any source are displayed at all terminals. The initial PDSP hardware arrangement will be equivalent to the EPSCS application except for the addition of PROMATS. Also, the data link units are dormant.

### C. Software Perspective

**2.03** The software design in an E911 data base management system using the PDSP has all of the capabilities of the No. 1 ESS version (although exact message formats may differ slightly plus some additional administrative support tools). The Extended Operating System (EOS), with its full complement of maintenance routines, plus the PIU communication and maintenance routines developed for the EPSCS project, is used as a programming base.

**2.04** The PROMATS (Programmable Magnetic Tape System) is used for RC input; the PROMATS is not used for data base backup or as an output device. All RC messages input via the PROMATS can also be entered via the Dataspeed 40 (DS40). These PROMATS tapes will be generated by a TELCO owned and operated 911 Data Management System (DMS) that monitors service orders to generate 911 assignments. When recent changes are input via the PROMATS and a message has an error, the Dataspeed 40 will print the message and an error description.

**2.05** The software design and implementation uses top-down methodology. The programming language is EPL with EDDL data structure specification which provides portability and modifiability. Listing documentation includes a drawing or the equivalent of each structure chart block as it is associated with the code.

### D. User Perspective

**2.06** The only interface the user has with the 911 data base is through recent change and verify messages. The user also updates the TDC data base backup as described in part 3 of this section. Initializing a data base involves entering four parameters directly via the DS40. All output messages contain enough heading and prose to provide for alignment and interpretation of the data fields. The system provides traffic counts and troubleshooting messages to ensure each function is being executed as expected.

## 3. DATA BASE DESCRIPTION

### A. File Description

**3.01** The E911 data base consists primarily of one large TN file (which does not necessarily

mean one large memory table). The TN file will map the Numbering Plan Digit (NPD) and a seven-digit TN to an Emergency Services Number (ESN). The NPD, of value 0-3, specifies the NPA or area code. If nonzero, it is also the first digit displayed on a calling line display at the Emergency Service Bureau (ESB). The ESN is a number that designates the set of emergency services a customer has (eg, police and fire). Each ESN may also specify whether the calling line display is to be flashed at the ESB. For example, the Flash Calling Line Display (FCLD) option might be applied to FX lines that terminate outside of the 911 serving area. The 911 TN file is very straightforward, and its important characteristics are its size (ie, 100s of thousand TNs) and its changeableness (ie, 100s of changes per day). The memory used to store the TN-ESN file and all associated tables of the data base are write protected. Only RC programs, after issuing a series of unlock messages, are able to change this memory.

### B. Storage Allocation

**3.02** The memory required for the E911 TN file is critically dependent upon the number of ESNs required per thousands number group and must be carefully considered. For example, suppose a county encompasses 110 office codes on a total of 1,100,000 TNs (assigned and unassigned). Perhaps 50 to 60 percent of these TNs represent assigned customer lines. If a direct access storage technique is required (more than 32 ESNs needed for every thousands number group), then this county could not be served by a single No. 1 ESS with PDSP E911 controlled office since the memory is exhausted at approximately 650K total TNs (assigned and unassigned). Alternately, if 9 to 16 ESNs are required for every TNG, one control office would prove to be sufficient since the memory will be exhausted at approximately 2.6 million total TNs (assigned and unassigned) for this level of translation.

### C. Data Base Backup

**3.03** A data base memory backup is always stored on duplicate TDC tapes. These four-track cartridge tapes hold 345 file blocks per track. Each file block contains 832 16-bit words of which four are control words. Track 0 is a nonwritable track and stores the bootstrap and all resident generic programs. Track 1 contains a duplicate of the bootstrap program, checksum file, a file directory, and the start of the data base. Tracks 2 and 3

are generally used by applications for the data base and possibly an RC file, but not for E911. The TDC tapes are very seldom needed during system recovery because all constant memory (eg, program instructions and office data tables) is write protected and all variable memory (eg, buffers and call registers) is reinitialized by emergency recovery programs. Thus, except for a few programs that can unlock the write protect mechanism (eg, recent change), memory is not susceptible to normal software faults.

**3.04** RC messages will not overwrite the data base stored on the TDC. After a set of RCs is entered, whether via the DS40 or the PROMATS, the current set of TDC tapes must be shelved, and then the data base must be copied onto two other previously shelved backup tapes. The recording mechanism is similar to a simple cassette tape recorder, and the recording process, referred to as TDC update, will take about 10 minutes per cartridge. If the RCs are later found to cause system problems, the shelved TDC tapes will be used to restore the data base to its previous state.

#### **TN-ESN File**

**3.05** The TN-ESN file translates the NPD and TN to an ESN. The TN-ESN file is made up of 1000s group pages. A 1000s group page is a relocatable memory block that contains 1 to 8 1000s group tables depending on ESN distribution. Figure 1 illustrates how the NPD and first four

digits of the TN select either an ESN or a 1000s group page.

**3.06** The 1000s group page may contain one full access 1000s group block or multiple abbreviated 1000s group blocks. The full access 1000s group block is simply a 1000-word block indexed by the last three digits of the TN to obtain the ESN. (The full access block will rarely, if ever, be used.) An abbreviated 1000s group block, as shown in Fig. 2, has three sections. The backpointer section contains the NPD and first four digits that select a particular 1000s group block. The abbreviated section is indexed by the last three digits of the TN to obtain the AESN (abbreviated ESN). The full section is indexed by the AESN to obtain the actual ESN. The size of the AESN determines the number of different ESNs a 1000s group serves without advancing to a larger abbreviated 1000s group block (Fig. 3). For example, a 3-bit AESN will accommodate 8 different ESNs in a 1000s group. If more ESNs are required, the 1000s group must use an abbreviated 1000s group block with a 4-bit AESN.

**3.07** The 1000s group page is 1104 words long and will contain combinations of tables with 2-, 3-, 4-, and 5-bit AESNs and full access blocks (Fig. 4, 5, 6, 7, and 8). The memory management program will automatically select the smallest abbreviated 1000s group applicable to a particular 1000s group. The figures show all blocks the same size; however, block sizes may vary.

## **4. RC MESSAGE INFORMATION**

### **A. Definitions**

**4.01** The following terms are used to describe RC messages.

**Keyword:** A symbol consisting of two to eight alphanumeric characters with the first character always a letter.

**Keyword Unit:** Both keyword and, when needed, accompanying data in the form of one or more variables. Variables are numbers or alphanumeric symbols, depending on the keyword. When there is more than one variable, the variables are separated by commas. The keyword must be separated from its data by a space.

Examples: NPD 0

Examples: RNG 9999

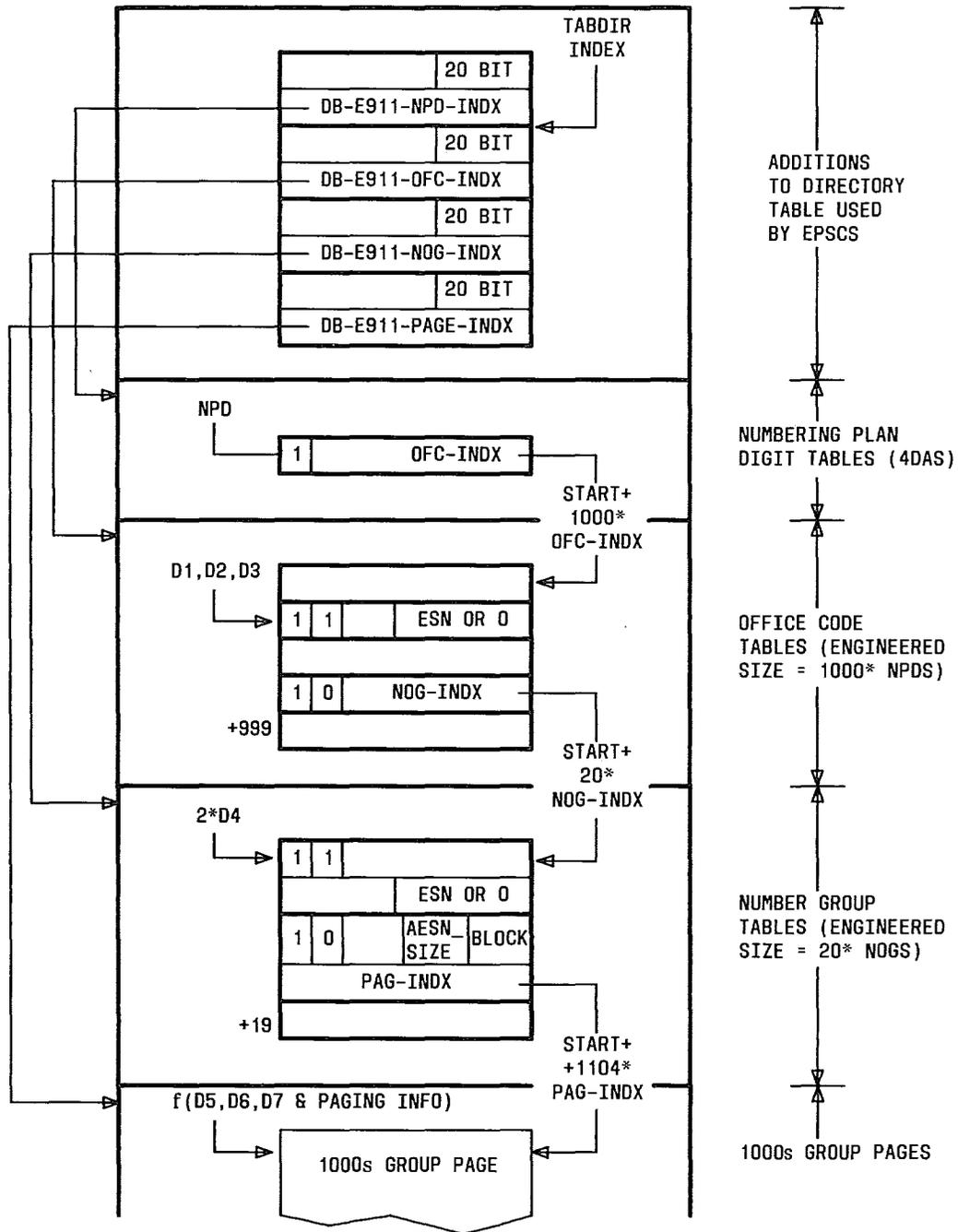


Fig. 1—TN-ESN File

An example of a keyword unit with two variables is:

ESN fff,bbb

Delimiter: Punctuation (, or; or: or/) used so that the PDSP can determine the end of fields, subfields, keywords, keyword units, or arguments within a message.

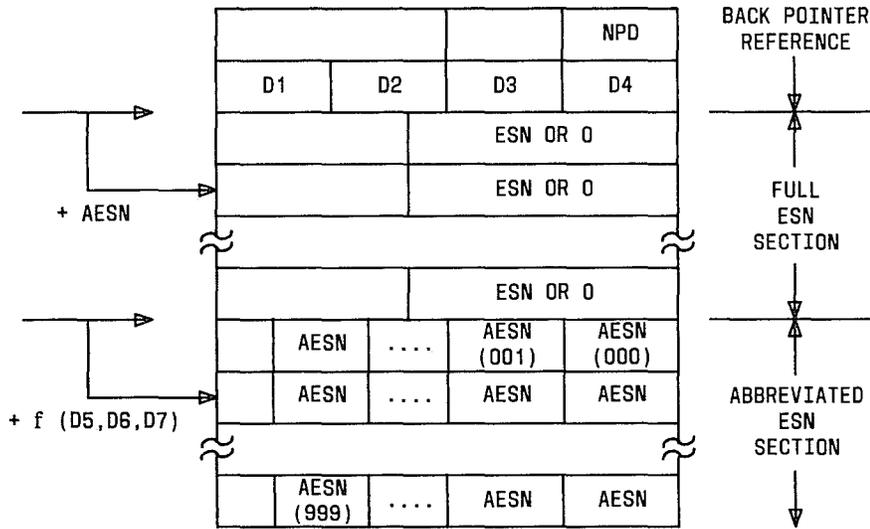


Fig. 2—General Structure of an Abbreviated 1000s Group Block

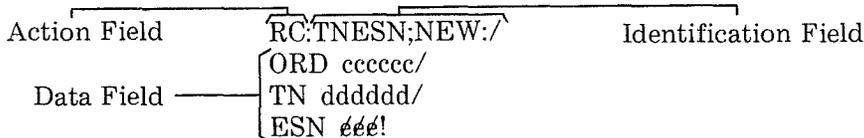
BITS FOR AESN	AESNs PER 16 BIT WORD	SIZE OF AESN SECTION	SIZE OF FULL ESN SECTION	TOTAL (INC. BACKPOINTER)
1	16	63	2	67
2	8	125	4	131
3	5	200	8	210
4	4	250	16	268
5	3	334	32	368
6	2	500	64	566
7	2	500	128	630
8	2	500	256	758

Fig. 3—Abbreviated 1000s Block Sizes

Default Value: The value that the PDSP assumes if no value is specified. For example: If the NPD keyword is not inputted, NPD=0 is assumed.

**B. RC Identification and Action Field**

4.02 RC messages are divided into three fields: action, identification, and data. The fields are always separated by a colon. The three fields are shown below using an example of an RC message.



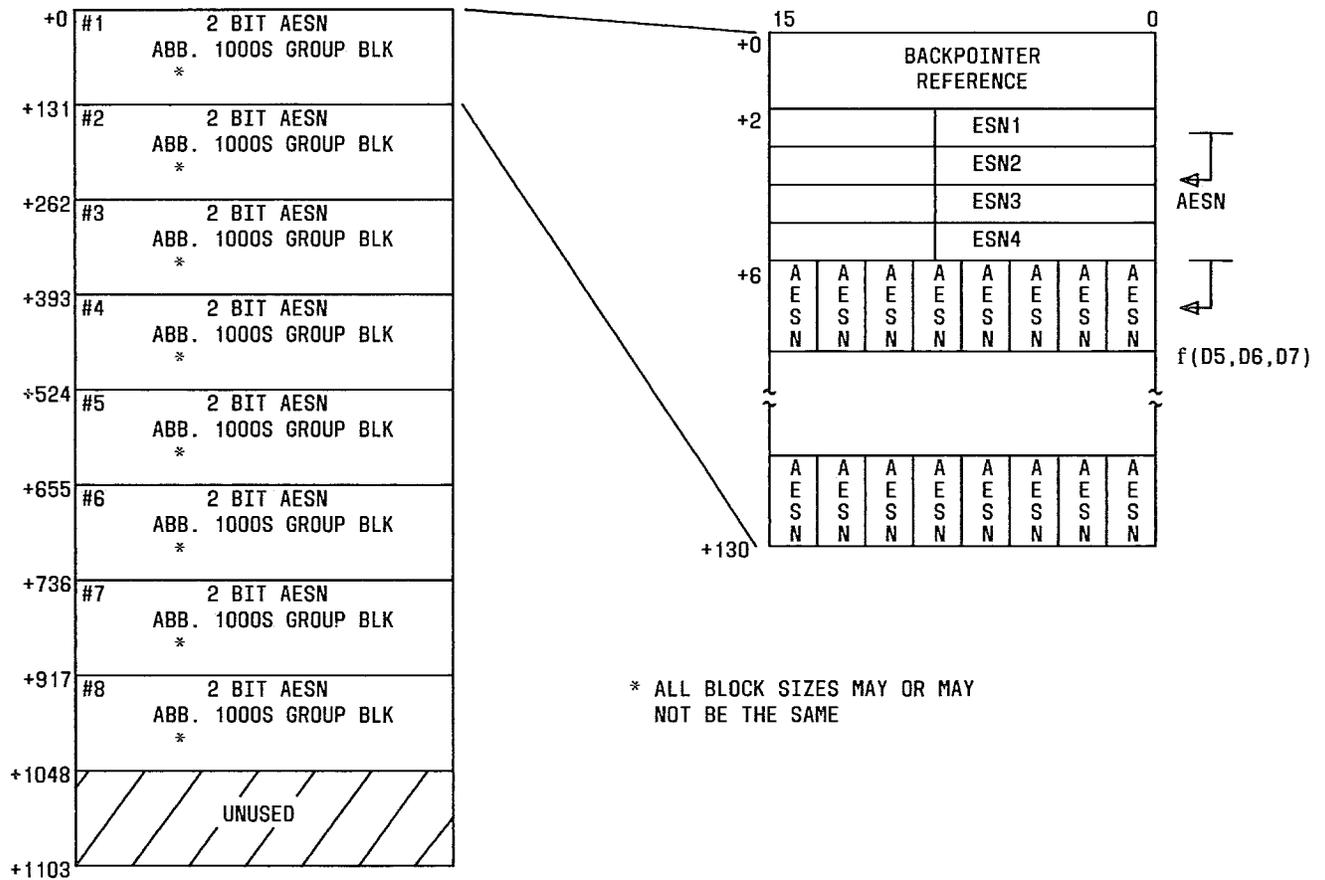


Fig. 4—1000s Group Page: 2-Bit AESN Blocks

- The action field in all RC messages is RC. The action field is the first field of the RC messages and is ended by the first colon. The action field for all RC messages is RC, to be interpreted as the command to make a recent change.
- The identification field contains two subfields. The first subfield indicates the object of the RC (PDSP data base table in this example). The second subfield indicates the type of message (NEW, CHG, or OUT). If the second subfield is omitted (eg, RC:TNESN:/), NEW is the default and no assignment check is made. The subfields in the identification field are separated by semicolons. Semicolons are not used elsewhere in the message.
- The data field is composed of keyword units. The keyword units in the message are composed of the keywords NOG, OFC, TN, RNG, ESN, ORD, and NPD, and their respective associated data. Note that the slashed e's in the ESN ~~eee~~ representation means the number may be 1, 2, or 3 digits. The TN must be 7 digits. The data field consists of keyword units and is the remaining portion of an RC message. The data field must begin on the second line of the message if entered at the TTY. Message heading lines and keyword units in the data field are separated by a slash. All messages are terminated by an exclamation mark (!).

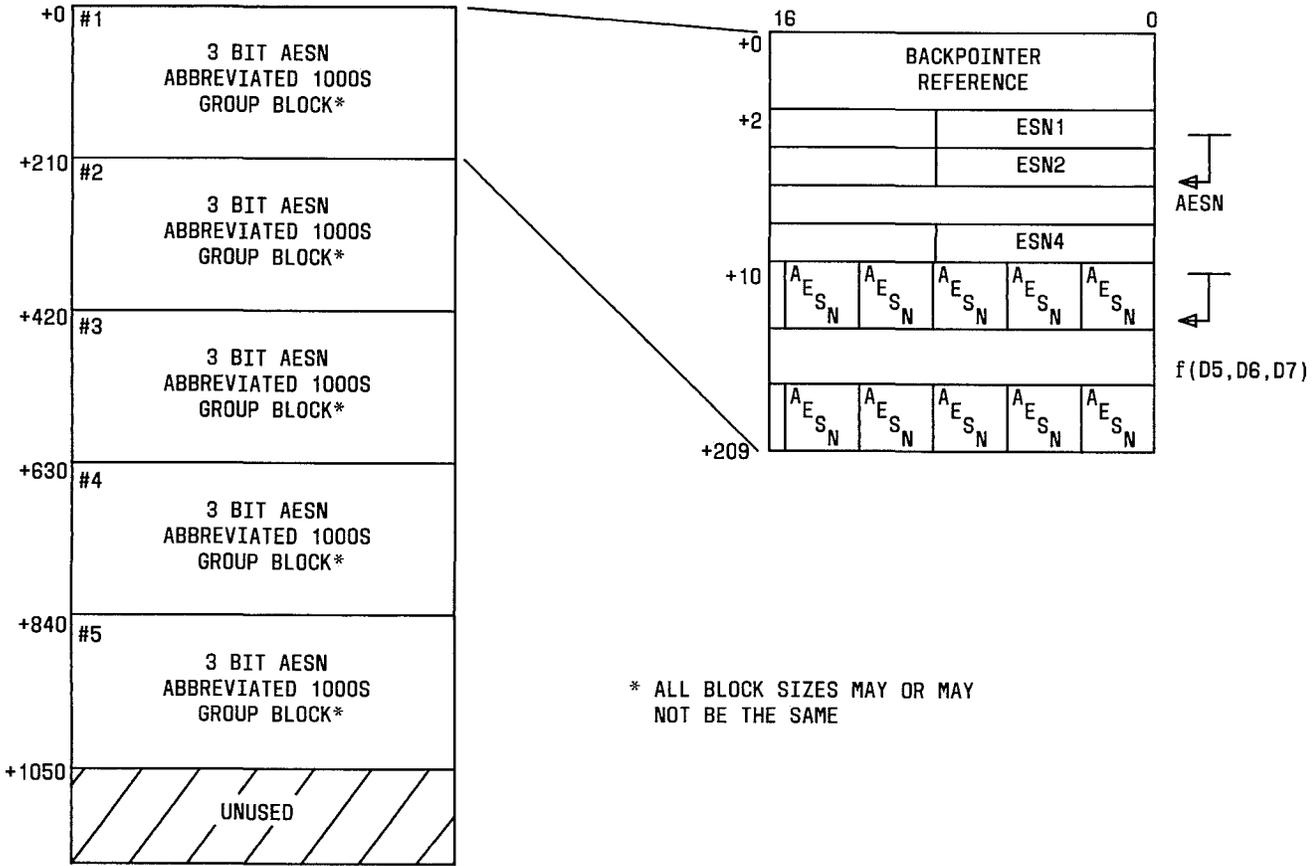


Fig. 5—1000s Group Page: 3-Bit AESN Blocks

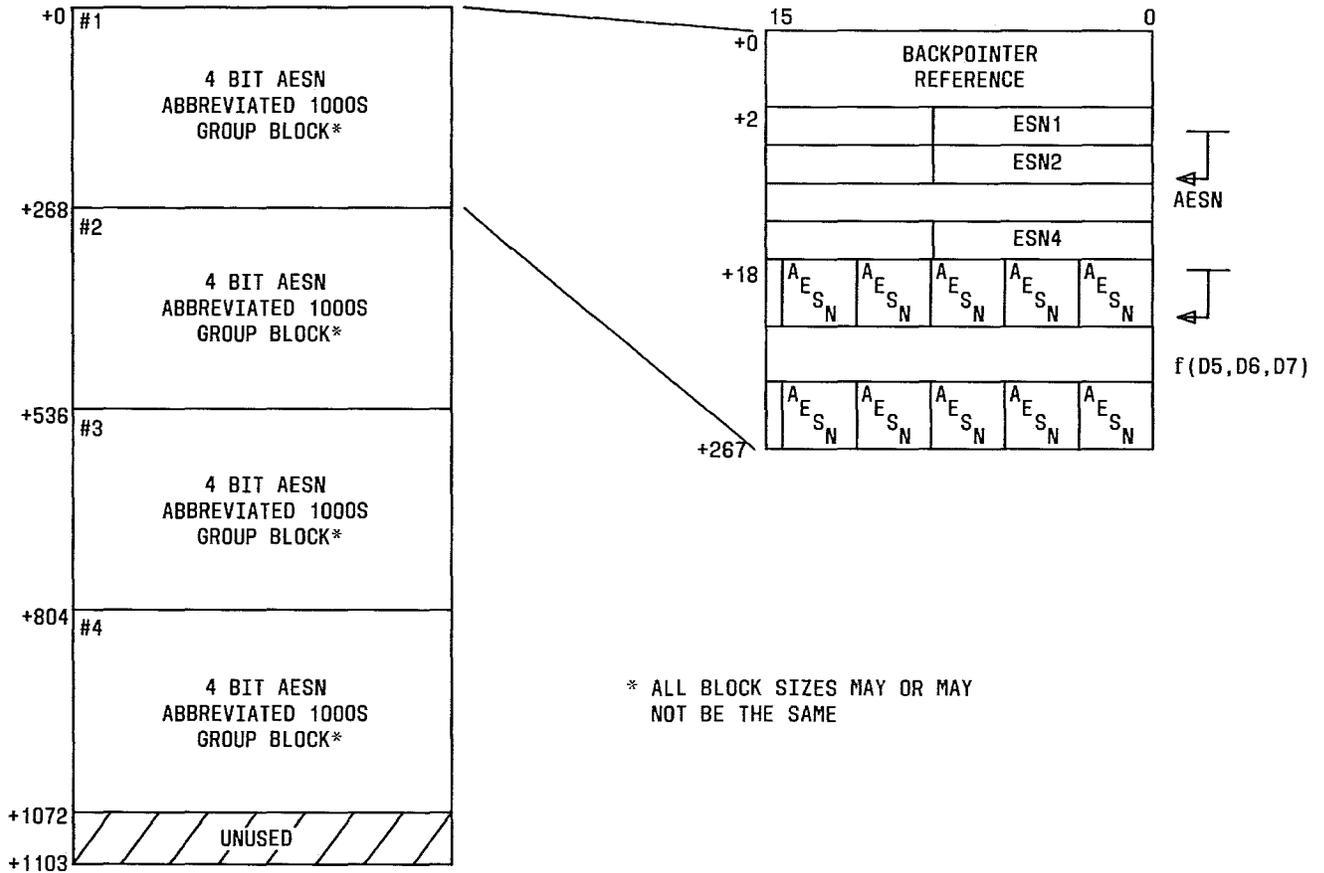


Fig. 6—1000s Group Page: 4-Bit AESN Blocks

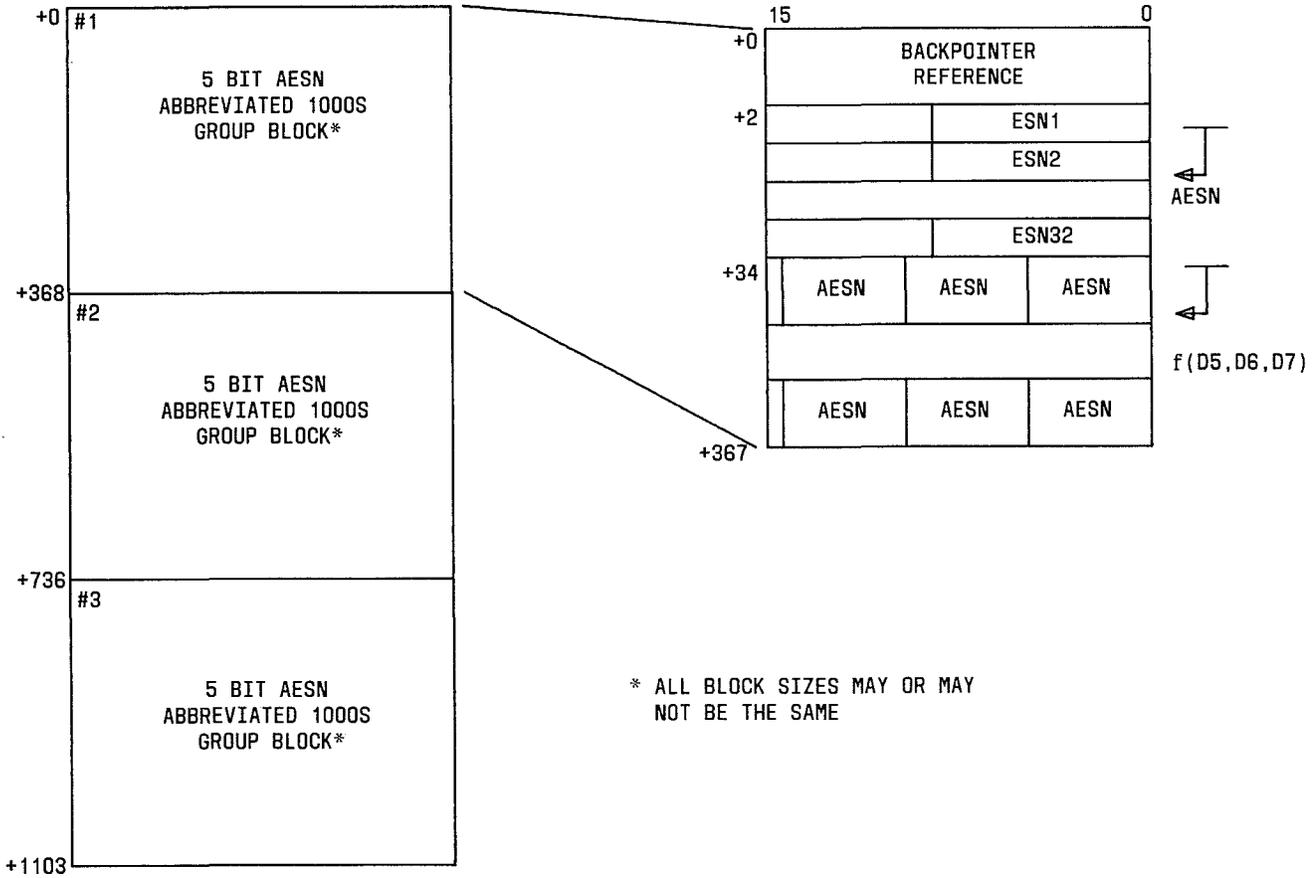


Fig. 7—1000s Group Page: 5-Bit AESN Blocks



5. 911 RECENT CHANGE MESSAGES

A. RC:TNESN Message Description

5.01 The complete set of E911 RC message formats is:

RC:TNESN (;NEW):

NOG hhhh "xor" OFC ggg "xor" TN aaaaaaa  
|  
○  
|  
RNG eeee

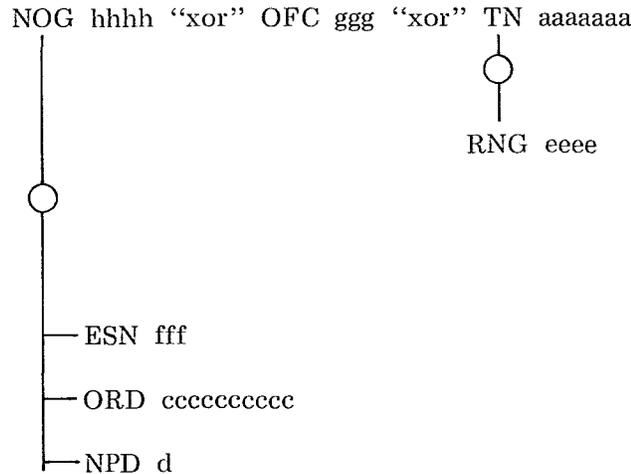
ESN bbb  
|  
○  
|  
— ORD ccccccccc  
|  
— NPD d

RC:TNESN;CHG:

NOG hhhh "xor" OFC ggg "xor" TN aaaaaaa  
|  
○  
|  
RNG eeee

ESN bbb "xor" ESN fff,bbb  
|  
○  
|  
— ORD ccccccccc  
|  
— NPD d

RC:TNESN;OUT:



**5.02** Underscored parts indicate leading zeros can be deleted, parentheses indicate optional fields, aaaaaaa specifies the TN, bbb specifies a new ESN assignment, ccccccccc specifies 10 character alphanumeric order label (not used for processing, but available on error messages for identification purposes along with TN), d specifies the NPD (Numbering Plan Digit), eeee specifies a range of TNs aaaaaaa through aaeeee, and fff specifies the existing ESN assignment on CHG or OUT orders.

**5.03** Instead of using the RNG keyword for assigning an entire office code or number group, all TNs in office code ggg (ie, TNs ggg0000-ggg9999) or all TNs in number group hhhh can be assigned using OFC and NOG keywords, respectively. The message flowchart indicates that for the NEW type message the keywords ORD, NPD, and RNG are optional. The keyword ESN plus one of the following - TN, NOG, or OFC - are required. If NPD is not input, NPD = 0 is assumed. The NEW, CHG, and OUT parameters are referred to as subheadings. If the subheading NEW is specified, any segment is rejected if the TN is already assigned; likewise, if CHG is specified, any segment is rejected if the TN is not assigned. If neither CHG or NEW is specified, no check is made to determine whether the TN is assigned or unassigned. Finally, if the old data is specified (ie, fff) with the CHG or OUT subheading, the message is rejected if the ESN assigned does not match the value specified by fff. For DS40 input, the punctuation required is simply a slash at the end of each line except the last line which requires an exclamation point.

**5.04** RC messages can be rejected for many reasons, such as invalid ESN, invalid use of the range keyword (eg, negative range), invalid punctuation or characters, and no room in memory. If the message source is PROMATS, tape processing will be stopped (unless the IPSOE option described in paragraph 5.06 is to be used). Regardless of whether the source of the message was the DS40 or PROMATS, the error messages are routed to the DS40. A complete list of error messages is contained in the program which prints the errors (PIDENT RCTN). Refer to Table A.

## B. RCCNTL Mode Message

**5.05** A set of control modes (refer to paragraph 5.18 for message details) allows the RC:TNESN messages to be processed in several ways. Most of these exist directly or indirectly to accommodate RC:TNESN messages entered via PROMATS where no paper copy usually exists. Perhaps it is not obvious, but a printout of each RC:TNESN entered during system initialization would be one foot high for about every 75,000 TNs. Another important fact about these control modes is that they offer over 64,000 ways of processing RC:TNESN messages (via possible combinations of options), but only a handful of these are

recommended. In fact, a default set exists and is appropriate for most message processing. Also, several more control modes are built into the PROMATS tape processing routines (see Part 6), but again the default case is usually sufficient. In short, a well supplied arsenal of control modes is provided to facilitate the handling of unusual and unpredictable errors that can occur when handling such large volumes of data. On the other hand, processing a daily update tape generally only requires typing the PROMATS:RC911;START message with the ID keyword.

### C. RCCNTL Modes for PROMATS

**5.06** Three RCCNTL modes are for PROMATS input only: Inhibit PROMATS Stop On Error (IPSOE), Print Accept message (PACPT), and Print Echo (PECHO). Normally, the PROMATS tape input will stop if an error is encountered; however, if the IPSOE mode is set, the tape input will continue unless the input message could not be deciphered. An accept message is only printed automatically on DS40 RC:TNESN message; however, if the PACPT mode is set, an accept message can be printed for each RC:TNESN message entered via PROMATS. The accept message, which does print the ORD keyword unit if input, has little value in a PROMATS environment unless the ORD keyword is used.

**5.07** The PECHO mode specifies when an RC:TNESN message entered via PROMATS should be printed at the DS40. In the default mode, only RC:TNESN messages with errors are printed. Furthermore, as described in part 6, if the message is so garbled that the PROMATS routines cannot even parse it, an ASCII dump of the record is unconditionally printed and the PROMATS input is suspended. Thus, errors in the context of the default mode refer to errors where RC processing cannot be completed for reasons such as table overflow, invalid ESN assignment, or TN assignment (eg, a NEW order for a TN already assigned). The echo message contains one additional piece of the information; it indicates whether the message caused an error, or if it did not cause an error whether it changed the data base. Since one RC:TNESN can span up to 10,000 TNs, the message will indicate change if any one or all of the TNs are changed in a range. The format of this echo message is:

```
RC TNESN (NEW, CHG, or OUT)
(ORD ccccccccc)
NPD d
TN aaaaaaa
(RNG eeee)
ESN (fff,) bbb
ECHO ERROR COMPLETE
      "xor"
ECHO DATA CHG COMPLETE
      "xor"
ECHO NO CHG COMPLETE
```

**5.08** Where the keywords are the same as described in paragraph 5.01 (Part A), parentheses indicate option output, and keywords are always printed in specified order regardless of order of input. Also, the NPD, which is assumed to be zero if not input, is always printed. The possible values of the PECHO mode, specified in paragraph 5.18 (Part G), allow any combination of the echo messages to be selected for printing (ie, ERROR, DATA CHG, or NO CHG). However, it is strongly recommended that the PECHO values that do not print the ERROR echo message never be used. Also, if an office wishes to keep a daily paper record of each RC:TNESN entered via PROMATS, one PECHO mode provides for the printing of an echo message on every message.

#### D. RCCNTL Modes Used During Comparison Runs

5.09 Three RCCNTL modes were designed to facilitate the comparison of a PROMATS tape to the data base; Inhibit Data Base Change (ICHG), Inhibit Status CHecks (ISTA), and Generate OUT messages (GOUTS). It is recommended that the initialization tape(s) be reentered in a comparison mode to insure the data base is properly built. It is also recommended that periodically a set of tapes identical to the initialization run, but for purposes of comparison with the data base, be produced and compared to the data base. This will insure that the data base is synchronized with the DMS (which supplies the PROMATS tapes) and particularly that no daily update tapes were inadvertently lost.

5.10 The ICHG mode prevents any RC:TNESN message from changing the data base. This mode is not always needed, or even desirable, on a comparison run. If changes are allowed on the comparison run, each of these messages will reflect a data change; if changes are not allowed, the message corresponding to the data in memory after the initialization is complete will not reflect a change and thus will not be printed. Finally, this mode, like all of the modes, is not restricted to the particular case of comparison runs and may be useful for other special purposes.

5.11 The ISTA mode overrides all assignment checks associated with the TN. In particular, an RC:TNESN message will not be rejected if the NEW, OUT, or CHG case does not match the TN assignment or if the old ESN (on the ESN keyword) does not match the ESN assigned. This mode is a must for comparison runs if NEW, OUT, and CHG subheadings are used. For example, if an initialization tape is reentered in the comparison mode, every message would have the NEW subheading but yet every TN in the data base should already be assigned.

5.12 The GOUTS mode applies only if the input TNs on the PROMATS tape are in sequential order within each office code. This mode generates an "OUT message" for all TNs between each two TNs on the tape. This out message is never allowed to change the data base, but will generate the following message whenever the TNs in its range are assigned:

```
RC TNESN GOUTS
(ORD ccccccccc)
NPD d
TN aaaaaaa
(RNG eeee)
ESN (fff,) bbb
ECHO DATA CHG COMPLETE
```

5.13 A GOUTS mismatch is considered an error and will stop the tape unless the IPSOE mode is set. If roughly 50 percent of the TNs in each office code are assigned, the GOUTS mode will increase comparison tape processing by about 30 percent. The GOUTS message deals strictly on an office code basis. For example, if one TN on the tape is 263-9978 and the next is 251-0002, two OUT messages are generated for comparison to the data base. The first is for TNs 263-9979 through 267-9999 and the second for TNs 251-0000 through 251-0001. Any intermediate office codes are always ignored. The GOUTS mode requires a START and STOP message before and after tape processing as follows:

RC:TNESN/  
STOP bbbbbbb



NPD c

and

RC:TNESN/  
START aaaaaaa



NPD c

5.14 In this message, the aaaaaaa and bbbbbbb are the start and stop TNs for comparison. Generally, aaaaaaa can be 0000000 and bbbbbbb can be 9999999. If any TNs are found to be out of order, the GOUTS processing is aborted and an error message is printed on every RC:TNESN until the GOUTS mode is turned off or a START TN is reentered. Thus, it is imperative that TNs be in sequential order.

#### E. RCCNTL Mode to Limit Table Growth

5.15 The Grow Limit (GLIMIT) mode was designed to provide a way, albeit it tedious, to minimize the effect of underengineering memory. The programs were designed to grow from smaller tables to larger tables as required to support the number of different ESNs in a 1000s group. If memory overflow is encountered during initialization, the simplest approach is to route the remaining office code(s) to the trunk group default ESN in No. 1 ESS. However, there is an alternative. Generally speaking, there are a few TNs in a 1000s group, such as FX lines, that cause a 1000s group to grow (or at least prevent it from being smaller). Using the OP:STAT;ESN and OP:STAT;NOG messages for the entire data base, it is possible to pinpoint 1000s groups where a few TNs have caused table growth. Once this is done the data base can be zeroed using standard procedures. Then, before the tape is processed again, a GLIMIT can be established using the RC:RCCNTL message. Next, for any 1000s groups that will exceed this GLIMIT, the "common ESNs" can be entered via the DS40 using any TN in the number group and then doing an OUT order on this TN to return it to its initial state. This reserves a slot in the 1000s block for the common ESNs (ie, the "FULL ESN SECTION" shown in Fig. 2 is filled with the common ESNs to be assigned). Now, the initialization tape(s) can be processed (with the IPSOE mode set because several errors can be expected due to table overflow). When this process is complete, only those few TNs in each number group with unusual ESN assignment will be routed to the trunk group default ESN in No. 1 ESS because they are not assigned in the PDSP.

#### F. RCCNTL Mode To Override ESN Check

5.16 The IESN mode was designed to allow the input of RC:TNESN messages even though the PDSP is not connected to No. 1 ESS via the PIU. If the PIU is up, the PDSP determines which ESNs are assigned in the No. 1 ESS and are assignable in the PDSP.

#### G. Recommended Set of RCCNTL Modes

5.17 On initialization runs all modes should be off except IPSOE which should be set if the tape is to be left unattended. Also, if the PIU interface between No. 1 ESS and the PDSP is down, all

ESNs will appear to be unassigned. In this case, the IESN can be set to test the initialization process, but it should **never be** set during a real initialization. On the comparison run, PECHO should equal print echo on errors or change (value of 2) and if NEW, CHG, and OUT subheadings are used, the ISTA mode must be set. Also, the IPSOE, ICHG, and IESN can be used if necessary (although they should seldom be required). On daily updates, it is recommended that all modes be off unless a paper copy of the RC:TNESN message is desired. In this case, PECHO will be in the print "always" mode (value 1) and the PACPT mode should be set. To warn the craft that the modes are affecting how RC:TNESN messages are being processed, if any of the modes are set, a warning message is printed as the tape process is started, after every 1000 RCs are processed, and with every RC typed in via the DS40. For RCs typed in via the DS40, modes that apply only to PROMATS input (ie, IPSOE and PACPT) are not printed.

#### H. RCCNTL Message Description

5.18 The following RC message sets 911 RC controls. ♦For the RC:PDTABL and OP:PDTABL errors, refer to Table B.♦

RC:PDTABL(;NEW)/  
RCCNTL/



PECHO a  
PACPT b  
GOUTS c  
IPSOE d  
IESN e  
ISTA f  
ICHG g  
PASG h  
PUNASG i  
GLIMIT j

RC:PDTABL;CHG/  
RCCNTL/



PECHO A, a  
PACPT B, b  
GOUTS C, c  
IPSOE D, d  
IESN E, e  
ISTA F, f  
ICHG G, g  
PASG H, h  
PUNASG I, i  
GLIMIT J, j

RC:PDTABL;OUT/  
RCCNTL!

## SECTION 231-144-360

5.19 The default for the ID verb is NEW. On NEW message if a keyword is not input, it is not changed **and** the old data associated with input keywords must be zero. On OUT messages, **all** keyword values are set to zero. On CHG message the old data **must** be specified. The keywords have the following description:

PECHO—Conditions under which the RC:TNESN message should be echoed if input via PROMATS (ie, does not affect DS40 input).

0 = only on error

1 = always

2 = on error or data base change

3 = only on messages that change data base (2 is recommended over 3)

4 = on error or no data base change

5 = only on messages that do not change data base (4 is recommended over 5)

6 = request no echo whatsoever (0 is recommended over 6)

PACPT—Print Accept message on RC:TNESN input via PROMATS if = 1; else 0.

GOOTS—Generate OUT messages for comparison only if = 1; else 0.

IPSOE—Inhibit stopping of PROMATS on error in RC:TNESN message if = 1; else 0.

IESN—Inhibits check of ESN to insure it is assigned in No. 1 ESS if = 1; else 0. This should only = 1 in pre-cutover system testing where No. 1 ESS is not available.

ISTA—Inhibits all checks associated with the assignment status of a TN including check of old data on CHG RC:TNESN order if = 1; else 0. (For example, an RC:TNESN;NEW message is normally allowed only if TN is presently unassigned; this mode will override that check).

ICHG—Presents any RC:TNESN message from changing the data base if = 1; else 0. All other processing proceeds as normally including possibly a message echo that indicates the data base would have been changed and the printing of an accept message.

GLIMIT—Grow limit which limits the number of different ESNs that can be assigned in one 1000s group to 4 if = 1, 8 if = 2, 16 if = 3 and 32 if = 4.

## 6. PROMATS TAPE OF RC:TNESN MESSAGES

6.01 In the E911 system, the PROMATS tape is used to transport RC information from the DMS to the PDSP.

### A. Process Overview

6.02 The E911 data base system (PDSP) contains a process which reads RC (Recent Change) messages from the PROMATS 9-track tape and formats them into an internal format. The PROMATS messages are identical to messages from the TTY, consisting of a string of up to 80 ASCII characters, and the resulting internal format is consistent with the internal format produced by the TTY process.

Because of this, a single data base update program can process messages from either the TTY or the PROMATS processes. See Fig. 9.

**6.03** In producing the internal message format, the PROMATS process can recognize any keyword recognized by the TTY process. This was provided for ease of future expansions. It should be noted that, at this time, the PROMATS process recognizes messages which began with the characters RC:TNESN; however, beyond that check any valid keywords will be recognized. The keyword code and the characters which follow that keyword (ie, the data field — up to ten characters) are moved into an internal format message. That internal message is then passed on to the data base change programs for further processing.

**6.04** The use of the PROMATS process need not be restricted to data base change (RC) messages. By the addition of a check to recognize different message types (other than RC:TNESN) and to route messages selectively to other processes (dependent on message type), the PROMATS process can be generalized so that additional message types can be put in from the 9-track tape.

## B. Process Control Structure

**6.05** The PROMATS process can be envisioned (on the highest level) as a state graph with four states. (See Fig. 10). Each state represents the current activity of the process. The initial state (INIT) is the inactive state; all the internal buffers and counters of the process are zeroed, the process is ready to start. The running state (RUN) indicates that the process is in progress, records are being read from PROMATS, converted to internal format, and sent to the data base change routines. The other two states represent interruptions in the normal processing. The suspend state (SUSPEND) is entered either by operator request or when an error is encountered and permits commands which can be used to directly manipulate the PROMATS tape. The end-of-tape state (EOT) is entered when the end of the currently processing tape is encountered. It permits the processing of multitape files.

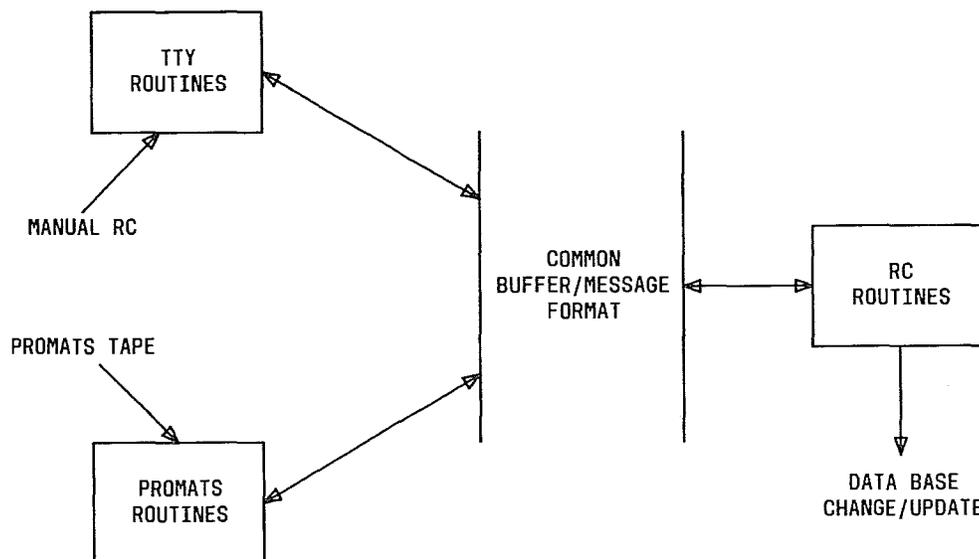


Fig. 9—System Overview

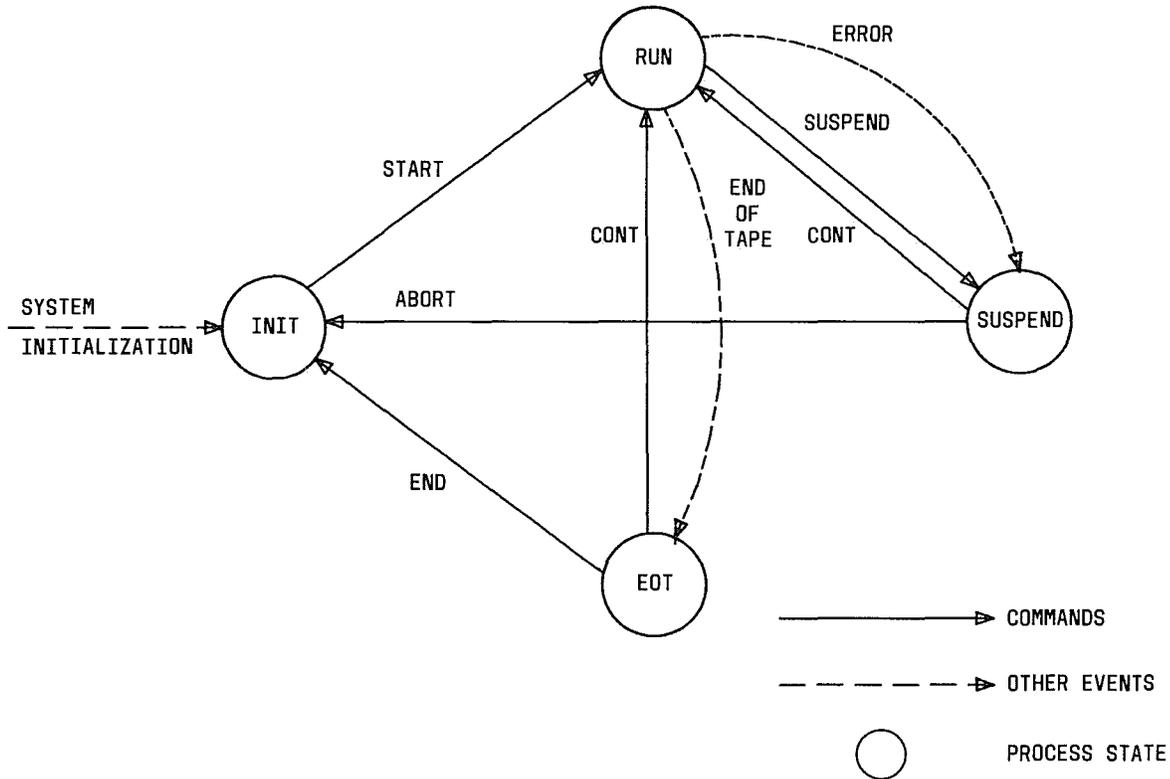


Fig. 10—PROMATS Process State Diagram

### C. Process Control Commands

**6.06** In the PROMATS process the transitions from state to state are caused by three conditions. If either the EOT condition or any error condition is encountered, then a transition to a new state is carried out: from RUN to SUSPEND in the case of an error, and from RUN to EOT in the case of the EOT condition. All other transitions are caused by operator-entered commands. See Fig. 10. These commands permit an orderly control of the process.

**6.07** There are six commands which control the process. These commands are START, SUSPEND, CONTROL, CONT (continue), END, and ABORT.

**6.08** The start command causes a transition from the initial state to the running state. When this transition takes place, the record number is initialized, some checking is done—the tape is “attached” and “opened,” and the first record is read—the header record. The start command has some keywords which the operator can enter which will control echoing of the tape records on the TTY, whether or not the records are sent to the data base system, and where on the tape processing is to begin.

**6.09** The suspend command causes a transition from the run state to the suspend state. This command has the purpose of permitting orderly control of the process and the PROMATS tape. Because control command (such as to skip records) can be executed by the PROMATS process, it is desirable to halt the process before executing those basic controls. In the suspend state all record counts, buffers, and internal flags are kept so that (via the continue command) the process can be resumed where it was interrupted.

- 6.10** The control command enables basic control of the PROMATS tape. It causes a null transition, as going from the suspend state to the suspend state. See Fig. 10. Through the use of this command, an operator can cause the PROMATS process to skip forward over records, backspace over records, or rewind the tape.
- 6.11** The continue command (CONT) is used to resume the processing. It causes one of two transitions. If the process had been suspended by command or an error had been encountered (eg, the process was in the suspend state) then the continue command causes a transition from the suspend state to the running state. The other state from which the continue command can cause a transition is the EOT state. In multitape files, a collection of related data base changes which cover more than one tape, when the end of a tape is encountered and a transition from the running state to the EOT state has taken place, to start the next tape the operator (after mounting the next tape) types the continue command. This causes the transition from the EOT state to the RUN state.
- 6.12** The two other commands (END and ABORT) cause transitions to the initial state. The end command (END) is used to go from the EOT state to the initial state. The abort command (ABORT) causes the transition from the suspend to the initial state. Both of these commands cause the initial buffers, counters, and flags to be zeroed. Also, the PROMATS tape is rewound, closed, and detached.
- 6.13** To summarize, there are four states in the process: INIT, RUN, SUSPEND, and EOT. They are related as shown in Table C.
- 6.14** If an error during processing in the RUN state will cause a transition to the SUSPEND state, an EOT during processing will cause a transition to the EOT state.
- 6.15** The normal sequence of actions and commands is:
- (1) mount the tape
  - (2) enter the START command
  - (3) when the EOT is encountered
    - (a) for multitape files mount the next tape, enter the CONT command, and repeat step (2)
    - (b) for the last or only tape enter the END command.

#### **D. PROMATS Tape Format**

- 6.16** The PROMATS 9-track tape used to build the data base has the following format. The tape is a 1600 bpi tape consisting of records of 80 ASCII characters each. Seven-bit ASCII is used. The records are blocked 10 records per block; that is, groups of 10 records (800 characters) are written or read at one time—to speed processing. Each record, with the exception of the first and the last one, is an RC:TNESN message. Using the format described in the previous section, the RC messages are put on the tape as \*) character records, one RC message per record, with the record being padded on the right with blanks. The first and last records on the tape are not RC type records; they serve a special purpose. They are identified below as the header record (the first record) and the trailer record (the last record).
- 6.17** The header record contains three items: a tape identification, a sequence number, and a command field. See Fig. 11. The tape identification item is used in the start command as a check. The correct ID must be typed in with the start command. Processing of the PROMATS tape will not start until the ID given with the start command and the tape identification item on the header record match.

On mismatch the header record is printed (so that the tape identification item can be read and correctly matched). This item is given to make inadvertent starting of the wrong tape more difficult.

**6.18** The second item on the header record is the sequence number. This item can be used to enforce sequencing of multitape files. If the sequence number is given in the header record, then it is checked to assure that the tapes which are processed between a start and an end command are checked for sequence. The first tape of such a multitape file supplies the initial sequence number. On each successive tape the sequence number is checked to assure that the number on the current tape is greater than the sequence number on the previous tape. (The sequence numbers should be in increasing order). This check can be bypassed by using the SEQ keyword of either the start or the continue commands (START or CONT). That keyword permits the direct entry of the sequence number against which the check will be made.

**6.19** The last item on the header record is a comment used to document the tape identification in human readable form. This comment portion of the header record is printed whenever the tape is started (by either the start or continue command).

**6.20** The trailer record is the last record on the tape. See Fig. 11. It also has three items: the tape identifier, a record count, and a comment field. Both the tape identifier and the comment field are information fields; they are both printed out upon completion of each tape. The record count item is a process check. During the processing of a tape, a count of records handled is kept by the PROMATS process. If there is a discrepancy between the record count known to the process and the record count item in the trailer record, then a warning message is printed when the tape is finished. Note that this count includes both the header and the trailer records. It is a count of all the records on the tape.

**6.21** These three types of records constitute the contents of the entire PROMATS tape. The header record is followed by any number of RC:TNESN records, which are followed by the trailer record. The records are on the tape in blocks of 10 records each (800 characters per block) with the exception of the last block, which might be a "short" block. If the total number of records is not divisible by 10, the last few records are not enough to occupy a full block. To handle this, the last block on the tape can have fewer than 10 records (fewer than 800 characters).

#### E. Process Control TTY Commands

**6.22** There are six commands: START, SUSPEND, CONTROL, CONT, END, and ABORT. The commands START, CONT, and END are all that are normally needed for processing a tape. If unusual conditions occur, the SUSPEND, CONTROL, and ABORT commands offer additional control over

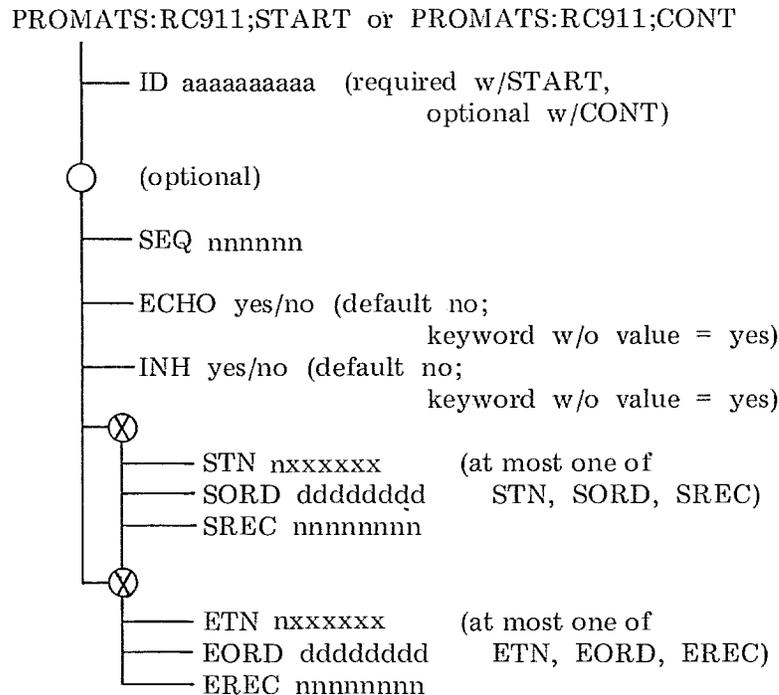
FORMAT



- A) ID LABEL  
1-8 CHARACTERS
- B) SEQUENCE NUMBER (ON HEADER)  
OR  
RECORD COUNT (ON TRAILER)  
1-6 DIGITS
- C) COMMENT FIELD

Fig. 11—Header and Trailer Records

the processing of the PROMATS tape. The start command is used to start the processing of the tape and the continue command causes suspended processing to be resumed. The format of these commands is:



6.23 The following is a list of keywords and definitions.

**START**—is used to signal to the PROMATS/RC system that the RC tape has been mounted and is ready to be read into the E911 system. When this command is entered, the control flags in the PROMATS are reset, the record count and the file sequence number are zeroed, and all buffers are cleared.

**CONT**—the “continue” command is used to signal the continuation of processing both in the case where there are multitape sets for the input (standard) or in the case where normal processing has been suspended (as by an error or by the SUSPEND command—see below).

**ID**—(optional) the tape ID. If the tape header record has an ID value, it is matched against the value given with this keyword. If no value is given with this keyword, or if the keyword is not entered with the message, the value is assumed to be null and will match only a null (unsupplied) ID field.

**SEQ**—(optional) the tape sequence number. If not given, the sequence number is set by the first tape. The **START** keyword causes the sequence number to be cleared; once cleared, it is reset from the header record of the tape. When subsequent tapes are read in (using the **CONT**) keyword, if the sequence keyword *is not* given, then one is added to the sequence number from the previous tape. If the **SEQ** keyword *is* given, then the given value is used. In any case, the sequence number is checked against the sequence number field on the header record of the tape. If the sequence number field of the tape was not omitted and the tape sequence number does not match the sequence number given with the **SEQ** command, processing is suspended and an error message is printed. The process can be resumed with the **CONT** keyword, using the **SEQ** keyword to supply the correct value.

ECHO—(optional) in addition to all processing (including passing the RC messages to the RC EXEC routine) the messages will be echoed on the TTY.

INH—(optional) any RC message which would normally be processed will have all checks performed on it, but it will not be passed on to the RC EXEC routine. Those messages will be put out on the TTY instead. In addition to the appropriate checks on the RC messages, all the tape checks (such as record count and header ID field) will also be performed.

STN and ETN—(optional) “start TN” and “end TN” can be used to select a range of records with which processing is to begin and end (respectively). The RC records are read from the tape until the first occurrence of the starting TN is located. The record with that TN is the first one processed. If no start TN is located before the end of the tape, then no records are processed. Once processing is commenced, it is continued until the end TN is matched or the EOT condition occurs, whichever is first. The end TN is the last record processed and the rest of the records on the tape are skipped.

SORD and EORD—(optional) “start order number” and “end order number” are the same as STN and ETN except that they search on order number.

SREC and EREC—(optional) “start record” and “end record” also select a range of records to process, based on record number. Records are numbered sequentially from the first RC record on the tape, starting from one.

- 6.24 Normal processing is terminated by the END command, after the end-of-file on the last (or only) tape. The format of this command is:

PROMATS:RC911;END

Note that this command is only valid when EOT has been encountered.

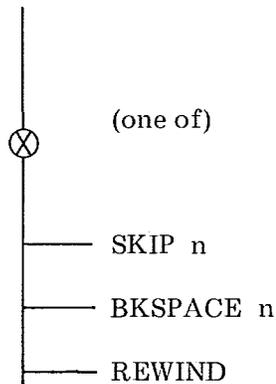
- 6.25 The commands (START, CONT, and END) are all that are needed to load and process the PROMATS/RC tape. However, it may occasionally be necessary to suspend processing. A special command is available for this. The message:

PROMATS:RC911;SUSPEND

will suspend the processing of the PROMATS tape. The process will retain all the information about current status of the tape, including the sequence number, ID field, record count and buffers in use. This is the only command which will be permitted while the processing of a tape is active. In addition to this command, an error can also cause the suspension of the processing of the tape. While the tape processing is active, all other messages will be rejected until processing has been suspended by either a SUSPEND command or an error condition.

- 6.26 Once the processing *has* been suspended it can either be resumed by using the CONT command, terminated by the ABORT command, or controlled directly by using the control command. The command CONTROL facilitates the direct control of the PROMATS tape. The format of that command and keyword definitions are as follows:

PROMATS:RC911;CONTROL



SKIP *n*—skip forward *n* records (the default for *n* is one). This keyword updates the record count and record number and skips forward the given number of records on the tape, or until EOT, whichever comes first.

BKSPACE *n*—backspace *n* records (default for *n* is one). As in the SKIP keyword, the record count and number are updated and the tape is positioned to the appropriate record, or the first RC record (nonheader record), whichever is closest.

REWIND—causes the tape to be rewound, but only the end-of-tape and new-tape flags are affected. No other flags are set; no buffers are zeroed.

**6.27** The abort command causes the processing to be terminated abnormally. The current tape will be rewound, all internal buffers, flags, and counters will be zeroed, and the process will be returned to the initial state. The format of the command is:

PROMATS:RC911;ABORT

**Note:** Any RC messages which have been accepted from the tape prior to this command being entered *have* been executed by the data base change routine; hence, the ABORT keyword cannot reverse changes made to the data base by those RC messages.

#### F. Output Messages

**6.28** There are three general classes of output messages for the PROMATS process. The first are those messages which are information messages, used to indicate the current state of the process. The second class are those messages which are information messages, used to echo the contents of a tape record, either when an error is encountered or when the record is normally printed (as with the header record). The third class of messages are the error messages. The two classes of information messages are described, and the error message format is outlined in this part.

**6.29** Associated with each state of the PROMATS process is an information message which indicates the current state of the process. Whenever the process changes state a message indicating the new state is printed; whenever a control command is entered, a message indicating that the process is suspended is printed; and whenever the process is active (in the RUN state) a message indicating that state is printed every one thousand records. Also, in the running state, a message is printed indicating those values of the RC control flags which have nondefault settings. The format of these messages is:

M tt PROMATS RC911 keyword

where

tt is minutes after the hour and

keyword is

ACTIVE for running state

SUSPEND for suspend state

END for end-of-tape state

STOPPED for initialize state

and

RCCNTL for print of  
RC control words

**6.30** The second general class messages are those used to echo PROMATS tape records. These messages are used in three places: first, in connection with error messages; second, when the ECHO keyword with a YES value has been entered on the START or CONT command; and third, when echoing the comment field of either the header or trailer records. If the data base update routines encounter an error in a message which came from the PROMATS tape, and they cannot make enough sense of it to print a meaningful error message, or if the PROMATS process cannot make enough sense of the record from tape to format a request to the data base update routines, then the record in question is echoed in the TTY, using the following format:

M tt PROMATS RC911 ECHO

text

where

tt is the minutes after the hour

and

text is the record in question

6.31 The same format is used to print either the header or trailer records. The entire header record is printed if there is an ID mismatch. The entire trailer record is printed when a trailer record error is encountered. When no error conditions have occurred, only the comment fields of the header or trailer record are printed.

6.32 The final general class of output messages are the error messages. These messages are printed whenever errors are encountered. Errors may occur in the processing of commands or during the reformatting of PROMATS records for the data base routines. The format of this message is:

M tt PROMATS RC911 ERROR

CODE n

where

tt is the minutes after the hour

and

n is the error message

number

6.33 These three general message formats permit information to be put out about the current state of, the records read into, and the errors encountered by the PROMATS process.

## 7. TTY INTERACTIVE DEVICE

7.01 The E911 data base system will use the DS40 as an interactive I/O device for making small volume recent changes, diagnosing hardware, verifying data, collecting traffic counts, etc. The DS40 will accept messages to change or verify the data base and to set system options. The results of these messages will be printed or displayed at the interactive device. All message formats will be in an ASCII line-by-line unblocked format.

## 8. OUTPUT MESSAGES

### A. OP:TNESN Messages

8.01 The following is output (OP) message including input, keyword, and display. ♦Refer to Table D for OP:TNESN errors.♦

VERIFY MESSAGE

INPUT MESSAGE FORMAT

OP:TNESN/  
TN aaaaaaa/ required  
RNG eeee/  
ESN fff! required  
NPD g (g = 0 assumed)

or

OP:TNESN/  
TN aaaaaaa/ required  
RNG eee!

NPD G (g = 0 assumed)

EXPLANATION

This message will display the contents of the data base. At present, inquiry is limited to one office code or ten thousand telephone numbers per message. It will take approximately one minute to process an entire office code.

The first form will display one telephone number or a range of TNs assigned to the specified ESN within an office code or 1000s group number. If the ESN is zero the displayed TN(s) are those with unassigned status. A message will be printed when no TN assignment matches the input ESN.

The second form will display one TN or a range of TNs within an office code or 1000s group number.

KEYWORDS

TN aaaaaaa  
Telephone number where the first three digits is the office code followed by four digits representing a 1000s number.

RNG eeee  
The range must be a positive displacement from the input 1000s number, but not greater than the last 1000s number (aaaa) in an office code, otherwise only the input TN will be displayed which is the same result when no range is specified.

ESN fff  
Emergency service number (0-511)



```

OP TNESN
START TN  END   ESN
aaaaaaa   eeee  bbb
aaaaaaa   eeee  bbb
.         .     .
.         .     .
aaaaaaa   eeee  bbb
OP TNESN COMPLETE

```

The "OP TNESN COMPLETE" trailer recorder only indicates that a print segment is complete; if more than one segment is printed, complete message with no data is printed to indicate all segments have been printed. If the ESN keyword is input and no TNs with that ESN assigned are found, the following is printed:

```

OP TNESN
NO ESN MATCH FOUND

```

### B. RCCNTL Verify

9.02 The RCCNTL verify message is:

```

OP:PDTABL:/
RCCNTL!

```

with the output message response of (see note)

```

OP RCCNTL
PECHO a
PACPT b
GOUTS c
IPSOE d
IESN e
ISTA f
ICHG g
PASG h
PUNASG i
GLIMIT j
OP REQUEST COMPLETE

```

**Note:** Valid where data items are the same as described in paragraph 5.18.

### C. ESN Statistics Message

9.03 To obtain a printout of the distribution of ESNs with the office code(s) range, the following message is used:

OP:STAT;ESN:



OFC aaa

NPD b (b=0 assumed if not input)

ESN xxx

where aaa=a specific office code, b=Numbering Plan Digit, and xxx=a specific ESN. If no office code is input, the ESN statistics for all office codes in specified NPD will be printed. If an ESN is input, then only the number of TNs assigned to that ESN in a particular office code will be given. The following output message is printed:

```

OP  STAT  ESN
NPD b OFC  aaa
ESN  NUM
xxx  yyy
.    .
.    .
xxx  yyy
OP  STAT  COMPLETE

```

where xxx=an ESN and yyy=the number of times that ESN is assigned within specific office code. ESN=0 is printed for unassigned lines. If a request for the statistics on all office codes is requested, no report will be generated if an entire office code is unassigned. As with previous segmented messages, the "OP STAT COMPLETE" only indicates a segment is complete and more data may follow.

#### D. MAXESNS Statistics Message

9.04 To obtain a printout of the distribution of 1000s blocks by abbreviated ESN size (see Fig. 4), the following message is input:

OP:STAT;MAXESN:



NPD b (b=0 assumed if not input)

where NPD=Numbering Plan Digit (ie, 0-3). The following output message is printed:

```

OP STAT MAXESN
NPD b
MAXESNS      NUM
    0         xxx
    1         xxx
    4         xxx
    8         xxx
   16         xxx
   32         xxx
  511         xxx
OP STAT COMPLETE

```

where xxx=the number of number groups that can be assigned up to the number of different ESNs specified under the MAXESNS column (including ESN = 0 for the unassigned case). The MAXESNS of zero is used for all number groups in an unassigned office code (ie, this xxx value divided by 10 equals the number of office codes with no assignments). The total of all xxx values will always be 10,000 because there are 10,000 possible number groups in an NPD.

**E. NOG Statistics Message**

9.05 To obtain the size of the 1000s block associated with particular number groups, the following messages are input:

```

OP:STAT;NOG:
  |
  O
  |
OFC aaa or NOG cccc
NPD b (b=0 assumed if not input)

```

where aaa=a specific office code or cccc=a specific number group for which MAXESNS is requested and b=Numbering Plan Digit. If neither OFC or NOG is input, the MAXESNS for all office codes are printed, and MAXESNS = 0 are not printed. The following output message is printed:

```

OP STAT NOG
NPD b
NOG      MAXESNS
cccc      ddd
.         .
.         .
cccc      ddd
OP STAT COMPLETE

```

where cccc=the four digits that specify a number group and ddd=0, 1, 4, 8, 16, 32, or 511 indicating the maximum number of ESNs that are assigned to that number group. Note that ccc=0 is used for number groups in an unassigned office code.

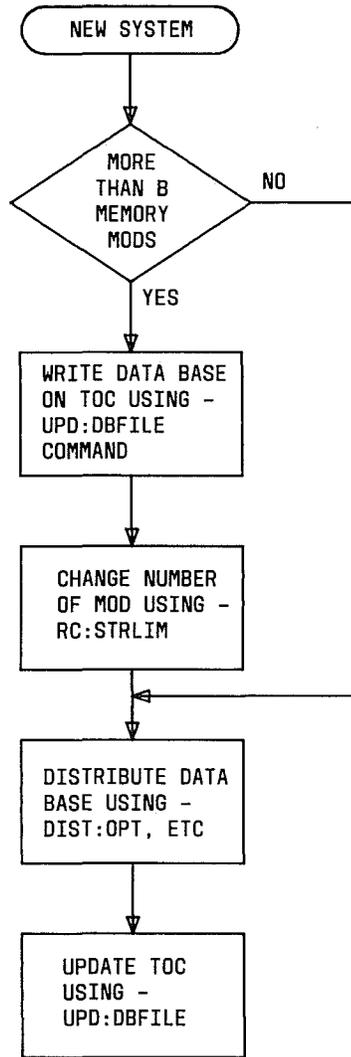
#### 10. DISTRIBUTING DATA BASE

**10.01** For distributing the data base, refer to Data Base Reallocation Section 231-144-303, and procedures and data base generation for EPSCS, Section 231-144-350.

**10.02** The message RC:STRLIM should be used to update the number of MODs in the system first. Also, STRLIM cannot be used until one update of data base has been performed. Refer to the following message and then to the following flowchart for initialization.

```
RC:PDTABL;CHG:/  
STRLIM/  
MOD a,b!
```

**Note:** STRLIM contains the address of the last accessible memory location. It should be changed whenever more memory is added to the PDSP. For MOD, a is the old data (number of mods of memory before growth) and b is new data (number of memory mods after growth). One mod of memory is 32K words.



11. SYSTEM AUDITS

11.01 A data base audit insures every TN is unassigned or assigned a valid ESN and that the data base has not been mutilated. First of all, it detects and prints at the DS40 any TN assigned an invalid ESN as follows:

AUDIT TNESN		
NPD	TN	ESN
a	Obbbbbbb	c c c

**11.02** An RC erasure in No. 1 ESS could cause an ESN already used in the PDSP to suddenly become invalid, so an error in TN-ESN assignment cannot be corrected by the audit. The audit cycles at about a minute per assigned office code and prints no more than 5 error messages per ESN per cycle through entire data base. Since most E911 applications will serve 100 or more office codes, the audit only completes a full cycle about once every 2 or 3 hours. The audit prints the total number of errors found at its conclusion regardless of the amount. The audit can be turned off by the message:

AUDIT:TNESN;STOP!

It can be turned on or restarted with the message:

AUDIT:TNESN;START!

When this start message is used, an error summary is printed even if the number of errors is zero.

**11.03** The audit also checks that all levels of the TNESN file are properly linked, including the backpointer in the 1000s block. (This backpointer contains the NPD and four digits that select the 1000s block). If any errors are detected, up to five of the following messages will be printed per audit cycle:

TNESN REPORT ERROR TYPE DBFILE ab  
CSECT ccccc DISP dddd  
"plus 48 words for analysis"

**11.04** Where a indicates which data base access program encountered the error, b is an error code, ccccc names the program, and dddd is the relocatable address where the print was requested. In general, this audit does everything possible to insure that the TNESN file is valid and has not been mutilated.

## 12. GLOSSARY

BCD	Binary Coded Decimal—represents a dialed digit in four bits ranging from 1-10 with dialed zero stored as 10.
DMS	Data Management System—the Telco owned software, which resides on a non-ESS computer, that prepares the input PROMATS tapes.
DS40	Dataspeed 40—input and output device which consists of a high speed printer and cathode ray tube (CRT) display.
EDDL	ESS Data Definition Language—provides a means of specifying an ESS data structure in a higher level language and then permits accessing each abstract data item independent of its physical size or location. The purpose of using this language is to provide easier data base modifiability by assigning data base access algorithms to the compiler rather than to the programmer.
EOS	Extended Operating System—a special PDSP software that is designed to provide commonality between PDSP applications to support basic event software event scheduling and common hardware devices.
EPL	ESS Programming Language—the highest level ESS programming language available on the PDSP at this time.

**SECTION 231-144-360**

- EPSCS      Enhanced Private Switched Communication Service—the first No. 1 ESS application to use the PDSP connected to the No. 1 ESS via the Processor Interface Unit (PIU).
- ESN        Emergency Service Number—a number (1-511) that defines the set of emergency services assigned to a telephone number (eg, police, fire, and hospital). Each ESN may also specify whether the calling line display should be flashed (see FCLD).
- FCLD      Flash the calling line display is an option provided to each ESN. This option could, for example, be assigned to FX lines that terminate outside the 911 serving area but may still inadvertently originate 911 calls.
- NPA        Numbering Plan Area or area code.
- NPD        Numbering Plan Digit—of value 0-3, specifies the NPA or area code. If nonzero, it is also the first digit displayed on a calling line display at the Emergency Service Bureau.
- PDSP      Peripheral Data Storage Processor.
- PIU        The Processor Interface Unit—recently developed hardware that couples a PDSP to a No. 1 ESS. It operates in a duplex mode and has full maintenance diagnostics. It has a first in - first out hardware buffer on each side of the interface unit and is attached to the No. 1 ESS call store bus. The cable length restriction is expected to be slightly less than 200 feet.
- PROMATS   The PROgrammable MAgnetic Tape System—a 9-track tape unit that is a component frame on the 3A auxiliary processor. It is controlled by a separate WE microprocessor called PROCON and uses a Direct Memory Access scheme to transfer its data to the 3A processor.
- TDC        Tape Data Cartridge—a cartridge tape storage device.

## →TABLE A←

## RC: TNESN ERRORS

CODE	DEFINITION
200	Improper keyword input
201	Character number 1 is invalid in data field
202	Character number 2 is invalid in data field
203	Character number 3 is invalid in data field
204	Character number 4 is invalid in data field
205	Character number 5 is invalid in data field
206	Character number 6 is invalid in data field
207	Character number 7 is invalid in data field
208	Character number 8 is invalid in data field
209	Character number 9 is invalid in data field
210	Character number 10 is invalid in data field
230	RNG keyword specifies negative range
231	Keyword is missing
232	TN assignment conflicts with new out or new
233	ESN not assigned in No. 1 ESS
234	Keyword is out of range (eg, NPD > 3 or ESN >511)
235	ESN keyword gives old ESN which does not match existing ESN in data base
236	No start for genouts received
237	TNs received during generate outs not sequential
251	NPD table overflow
252	Office code table overflow
253	NOG table overflow
254	1000's block page table overflow
255	AESN can only be 3 bits in SYS 1
271	Internal error in PIDENT RCWL routine WL D47 ENTS
272	Internal error in PIDENT RCLL routine LALCHGS
273	System error in RC EXEC
274	Internal error in PIDENT routine RCWL WL NPD of 1
275	Internal error in PIDENT RCWL routine arrange page
276	Internal error in PIDENT RCWL routine WL OFI NGI
277	Internal error in PIDENT RCLL RT write memory
278	Internal error in PIDENT RCLL RT grow 1000s

## →TABLE B←

## RC:PDTABL AND OP:PDTABL ERRORS

CODE	DEFINITION
9	Invalid tablename entered.
10	Keyword with data entry of zero (0) is not allowed for tablename entered.
11	Address of requested table is invalid.
12	Necessary keyword is missing from RC message
13	Keyword data is out of allowable range.
14	A paged program was not paged in.
15	There is no room in requested table for additional entries.
16	CHG message not allowed for requested table; must use OUT and then NEW RC messages.
17	Entry requested to be assigned is already assigned.
18	Old data is CHG message does not agree with information already in table.
19	Requested data cannot be output during reallocation.
20	Incorrect identification field entered in message header line.
21	Requested action cannot be inhibited during reallocation.
22	Requested action cannot be allowed during reallocation.
23	Incorrect sequence of commands entered.
24	Data in off-line and on-line memories does not match.
25	Data base entered is too large for existing memory.
26	Invalid data has been entered (perhaps wrong combination of data).
27	Invalid keyword has been entered.
28	Incorrect memory address has been specified.
33	Off-line CPU has not been put out of service or it is the manual mode; requested action cannot be completed.
34	Off-line CPU could not be reserved for use by this task.
35	Paging is in progress; requested action cannot be completed.
36	Copying of buffer from one CU to the other failed.
41	Input parameter should be even multiple of 4096 (4K) and was not.
42	The ADD parameter of input message is invalid if OFL parameter does not appear.

→TABLE C←

## PROCESS CONTROL COMMANDS

STATE	ENTER	CAUSES
INIT	START	a transition to RUN
RUN	SUSPEND	a transition to SUSPEND
SUSPEND	CONTROL	direct control of PROMATS
SUSPEND	CONT	a transition to RUN
SUSPEND	ABORT	a transition to INIT
EOT	CONT	a transition to RUN
EOT	END	a transition to INIT

→TABLE D←

## OP: TNESN ERRORS

CODE	DEFINITION
300	Invalid message number
301	Invalid action verb
302	Invalid ID1 verb
303	Invalid ID2 verb
304	Keyword missing
305	Invalid keyword data