

\$5.50 US, \$8.15 CAN (Haha.. April Fools!)

"... The Afghans are selling 7 to 8 billion dollars of drugs in the West a year. [Osama] bin Ladin oversees the export of drugs from Afghanistan. His people are involved in growing the crops, processing and shipping. When Americans buy drugs, they fund the jihad." – Yosef Bodansky, Director of the Congressional Task Force on Terrorism, 1998

Table of Contents

Table of Contents

- ◆ Page 2 / System Acknowledgement RC18 & RC16 Output Messages / #1 ESS
 - ♦ Overview of the most commonly seen #1/#1A ESS Recent Change output messages.
- ♦ Page 41 / Explanation of Nortel FMT-150 CRT Functions
 - ◆ The password is Eagle.
- ♦ Page 57 / Nortel DMS-100 Line-to-Line Translations
 - ♦ How dialing some numbers makes your phone ring.
- ◆ Page 76 / Nortel DMS-100 Introduction to Query Commands
 - ♦ Overview of how the DN/LEN query commands work on a DMS-100.
- ◆ Page 83 / Prison Pen Weapon
 - ◆ Turn a flexible safety pen into a stabbing weapon.
- ◆ Page 86 / Linux FRS Radio Control
 - ♦ Easy way to control a FRS radio from a computer.
- ◆ Page 96 / Bonus
 - ◆ Old ESS Dial-Up Inventory Form
- ◆ Page 97 / The End
 - ◆ Editorial and Rants.

RC18 & RC16 Output Messages / #1 ESS

BELL SYSTEM PRACTICES AT&TCo Standard SECTION 231-118-320 Issue 3, February 1981

RECENT CHANGE MESSAGE PROGRAM LISTINGS, SYSTEM ACKNOWLEDGEMENT, AND RC18 AND RC16 OUTPUT MESSAGES (THROUGH 1E5 GENERIC PROGRAM)

2-WIRE NO. 1 ELECTRONIC SWITCHING SYSTEM

	CONTENTS	PAGE	CONTENTS	PAGE
۱.	GENERAL	2	ERROR DICTIONARY SECTION	12
2.	DESCRIPTION OF RECENT CHANGE MESSAGE PROGRAM LISTINGS	4	A. Keyword Table	13
	A. Pident Introduction	6	B. Synonymous Keyword Table C. Keyword (Table) Error Dictionary .	13 13
	B. Message Head Table	6	D. Data Check Table Dictionary	13
	C. Definition and Reference Tables . INPUT SECTION	6	E. Validity (Tree) Error Dictionary	13
	A. Keyword Table	6	F. Format Selector List	13 13
	B. Internal Keyword Table	11	G. Validity Tree Diagram	14
	C. Keyword Equivalence Table D. Keyword Collision Table		3. TTY ACKNOWLEDGMENT (TACK) SYSTEM RESPONSE	15
	VALIDITY SECTION	11	A. Acceptable TACK (OK)	15
	A. Data Check Table	11	B. Line Error TACK	16
	B. Validity (Tree) Table	11	C. Message Error TACK (ER)	19
	C. Format Selector Table	11	D. Busy TACK (BUSY)	19
	TRANSLATION FORMAT SECTION	. 12	E. Invalid Heading TACK (?IH)	20
	A. Translation Format Table	12	F. Internal Program Error TACK (<interr>)</interr>	20
	B. Change Translation Screening Table .		G. Timeout TACK (<timeout>)</timeout>	20
	C. Change Keyword List	. 12	4. TTY OUTPUT MESSAGES FOLLOWING THE	21

NOTICE

Page 1

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

Printed in U.S.A.

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

	CONTENTS PA	GE	CONTENTS	PAGE
,	A. RC18 Output Message	21	Tables	
E	B. RC18 Interpretation Procedure	26	A. Nonmessage and Representative Message Pidents	3
•	C. ACPT-Type RC18 Message	26	B. RC Message Indexes	5
C	D. BRKRC-Type RC18 Message	28	C. Description of Columns on the Line	7
	E. CHGER-Type RC18 Message	28	, and the second	
ı	F. DELAY-Type RC18 message	28	D. Description of Columns on the Definition and Reference Tables	9
•	G. INED-Type RC18 Message	29	E. Validity Tree Nodes	14
ı	H. INFO-Type RC18 Message	29	F. List of Error Codes Appearing in	
١	I. INPUT-Type RC18 Message	30	Corrective Error TACKs and in RC18 Output Messages	17
	J. MTYPE-Type RC18 Message	30	G. RC18 Output Message Types	22
-	K. NOCR-Type RC18 Message	32	H. RC 6-Bit and ASCII Codes Associated	
1	L. OBJT-Type RC18 Message	32	With Input Terminal Characters and Operations	27
1	M. PUNCT-Type RC18 Message	32	I. Input Data Type Symbols in Program Listings	31
	N. VALER-Type RC18 Message	32		
	O. XLER-Type RC18 Message	32	1. GENERAL1.01 This section describes the contents of re	ocent
	P. NAUX, NAWL, NLLS, NOPS, NPRC, PLUGF, PRTER, and TABER Types of RC18 Message	33	change (RC) message program listings, sy acknowledgment, and RC18 and RC16 ou messages. This information applies to the 2- No. 1 Electronic Switching System (ESS) (three	stem Itput Wire ough
	Q. RC16 Output Message	33	1E5 generic programs). General RC informatis given in Section 231-118-321.	ation
	R. RC16 Interpretation Procedure	35	1.02 This section is reissued for the follo	wing
5.	ABBREVIATIONS AND ACRONYMS	36	reason: (a) To delete generics prior to 1E3.	•
Figu			(b) To add more detail to pident contents layout description (Part 2).	and 4
1.	Layout of a Typical Line Format (Table C Included)	7	(c) To make minor corrections as required	ı. ·
2.	Layout of the Definition and Reference Tables (Symbol and Macro List) (Table D		Program Listing	
	Included)	9	1.03 A program listing (PR) consists of a comp generated hardcopy list of program instruc	
3.	Octal-Decimal Conversion Table	30	and related information for one or more program	

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

units identified by program identifications (pidents). A pident is that segment of a program that is compiled as a unit by a compiler program (on a general purpose computer). Most PRs consist of a single pident; only a few contain more than one. This listing produced as a result of this compilation becomes the PR or a part of it if pidents are combined to perform a system function. No pident is split between PRs. The PR contains an index sheet which lists the pidents it contains and their issues, followed by the listings for each pident. PRs are numbered as PR-1A... for No. 1 ESS (... is some unique 3-digit number).

Program Identification

1.04 The recent change message interpretation programs form a system to interpret RC input messages, check them for validity, and

assemble the input data into the proper format for storage in translation memory. The RC message system consists of over 70 message pidents (1 per message), 10 code (or control) pidents, and 2 pidents shared by all the message pidents. The message pidents are tables of data defining the messages and the control pidents are the programs of executable code (ie, ESS instructions) that perform the functions of the RC message system. Many of these functions are accomplished by applying fixed algorithms to the data in the message tables; therefore, a substantial portion of the system is table driven by the message pidents.

1.05 Table A lists the 12 nonmessage pidents, a representative message pident, and references to associated PR documents. A brief description of the function of the control, shared, and message types of pidents shown in Table A follows.

TABLE A

NONMESSAGE AND REPESENTATIVE MESSAGE PIDENTS

PIDENT	TYPE	TITLE (EACH PREFIXED BY RECENT CHANGE)	PR-NO. 1 ESS
RCIG	Control	Initialization and General Control	PR-1A300
RCIE	1	Input Editor	PR-1A301
RCKI		Keyword Input	PR-1A302
RCVC		Validity Check	PR-1A303
RCTF		New Pass Translation Format	PR-1A304
RCCH		Change Pass Translation Format	PR-1A305
RCFI		Format Interpretation	PR-1A306
RCWL		Work List Processing	PR-1A307
RCDY		Delayed Order	PR-1A308
RSUB	♦	General Purpose Subroutines	PR-1A309
RCTS	Shared	Table Subroutines	PR-1A319
RCSI	1	Shared Information	PR-1A320
RCxx*	Message	(A Pident Per Message)	PR-1A3yy+

^{*} Two Letters Identifying a Message

Control Pidents

- RCIG (Initialization and General Control)—Determine availability of the RC system for the first line of the message and directs the flow of control among other modules in the RC system.
- *RCIE* (Input Editor)—Links the input/output terminal (IOT) buffer and the RC message processing system.
- RCKI (Keyword Input)—Performs processing and local error checking of RC message keyword units.

^{+ 21, 22,...}nn

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

- RCVC (Validity Checks)—Assures valid translation data base transitions.
- RCTF (New Pass Translation Format) RCCH (Change Pass Translation Format) and RCFI (Format Interpretation)—The format section builds images of translation data in the RC work list and RC auxiliary area on NEW and CHG messages and retrieves translation data from program store and RC area on CHG and OUT messages.
- **RCWL** (Work List Processing)—Processes the work list (WL) entries made by the message pidents.
- RCDY (Delayed Order)—Processes delayed service order by storing message storage buffer (MSB) input in a delayed activation block (DAB) for future reloading in the MSB.

Shared Pidents

- RCTS (Table Subroutines)—Consists of special purpose subroutines used by work list pident RCWL and the RC message pidents.
- RCSI (Shared Information)—Consists of five data tables which are shared by several of the code (control) pidents during RC message processing.

Message Pidents

• RCxx —In general there is a message pident for each type of translator. The message pident is made up of data tables which provide the decision information to control the processing flow of the associated code (control) pidents.

System Acknowledgment

1.06 The system's response to an RC input message is a TTY acknowledgment (TACK). If at anytime during an RC message input an input control character (!, %, ., &, or &) is received, the system responds immediately with some form of TACK. The TACK is immediate, in that, no other output message can intervene. The type of TACK received will indicate correct or incorrect

input of message, or some system condition preventing acceptance of the input message. See Part 3 for a detailed description of the various TACK responses.

Output Message (OM)

1.07 The ESS provides RC18, RC16, and/or RC FAILURE output message in response to the RC input messages. Twenty-one types of RC18 messages are available to indicate accepted RC messages, irregular system conditions affecting RC messages, or rejected RC messages that contain errors (see Part 4). The RC16 output messages consist of data (in octal) for use as an aid in the analysis of an error resulting from the generic program. These OMs have a lower priority rating than a TACK and could be delayed by some higher priority message.

2. DESCRIPTION OF RECENT CHANGE MESSAGE PROGRAM LISTINGS

2.01 A separate message pident is provided for each RC input message with the exception of the delayed mode activation message RC:ACT which is contained in control pident RCDY (RC delay order) PR-1A308. Table B displays the RC input messages and their corresponding pidents, PR numbers and message indexes. Each pident is provided with a message head table, an input section, a validity section, a translation format section, and an error dictionary section which is useful in the interpretation of RC18 and RC16 output messages.

PROGRAM LISTING PAGE FORMAT

The PR follows a standardized format in presenting significant information about the The pident name consists of four program. alphabetic characters PPPP; the pident issue consists of up to seven alphanumeric characters PPPPAAA. The first four characters must be identical to the program name. The remaining three characters identify the generic program. At the top of each page is a title line giving the time and date of assembly, the pident name and generic program number, the issue number of the associated compool libraries, and the version number of the program. At the bottom of each page is a title line giving the pident name and program generic number, issue number, page number, and the PR number

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

TABLE B RC MESSAGE INDEXES

MESSAGE INDEX (ii)	MESSAGE (RC:)	MESSAGE PIDENT	PR REF. (PR-1A)	MESSAGE INDEX	MESSAGE (RC:)	MESSAGE PIDENT	PR REF. (PR-1A)
0	LINE	RCLI	336	38	TRFHC	RCHC	335
1	PSWD	RCPS	334	39	TRFSLB	RCLB	337
2	TG	RCTG	359	40	JUNCT	RCJG	334
3	RI	RCRI	347	41	PLM	RCPM	343
4	CHRGX	RCCI	326	42	TGBVT	RCBV	321
5	TRK	RCTK	360	43	OBS	RCOB	341
				44	CAMA	RCCA	322
. 7	CFTRK	RCCF	325				
8	PSBLK	PCPB	342				
9	TGMEM	RCTM	362	47	TOLDIG	RCTL	361
10	DIGTRN	RCDG	328	48	TNCTX	RCTX	366
11	RATPAT	RCRP	348	49	ANIDL	RCAI	367
12	SCLIST	RCSC	350	50	TOBS	RCTO	371
13	MLHG	RCHG	332	51	NMTGC	RCNM	372
14	MSN	RCSN	355	52	ACT	RCDY	308
15	CPD	RCCD	.324	53	CTRF	RCCT	373
16	NOGRAC	RCNR	340	54	NUTS	RCNU	379
17	SUBTRAN	RCST	356	55	DALNK	RCDL	374
18	NOCNOG	RCNN	339	56	DATER	RCDT	376
19	CTXEXR	RCXR	369	57	DAMBI	RCDX	377
20	SIMFAC	RCSF	351	58	DAMSK	RCDM	375
21	CTXCB	RCCX	327	59	GENT	RCCT	378
22	DITABS	RCDI	330	60	CLAM	RCCM	380
23	CTXDI	RCXD	368	61	CFV	RCFV	383
24	CCOL	RCCC	323	62	UNASSIGNE	D (No.1 ES	S Unit Type
25	ROTL	RCRT	370	63	CXDICH	RCXI	382
26	DNHT	RCDH	329	64	DLG	RCDA	381
27	TMBCGA	RCTC	357	65	(Reserved for	AMPS)	
28	TNDM	RCTN	363	66	CCIS	RCIS	311
29	IDDD	RCID	333	67	CFG	RCCG	314
30	TDXD	RCTD	358	68	EPSID	RCEI	312
31	FLXDG	RCFD	331	69	AC	RCAC	315
32	FLXRS	RCRS	349	70	ACTABL	RCAT	316
33	FLXRD	RCRD	346	71	ESCO	RCES	317
34	MPTY	RCMP	338	72	ESN	RCEN	318
35	TWOPTY	RCTP	364	73	TNESN	RCER	384
36	PLUG	RCPU	345	74	TKCONV	RCTV	385
37	TRFLCU	RCTU	365				

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

(Fig. 1). In addition, within the program section, every program unit will start on a new page and have the program unit name at the top of each page.

PROGRAM LINE FORMAT

- 2.03 Each line in the program section of a PR contains a specified set of fields of information. It is not necessary for each line to include all fields of information as this depends on the type of instruction. The line format may include:
 - (a) Octal machine language
 - (b) Decimal numbers denoting line numbers
 - (c) Source data set line numbers
 - (d) Marco generation level numbers
 - (e) Program language statement
 - (f) In-line comments.

Figure 1 provides a layout of the line format and Table C gives a description of columns on the line format.

A. Pident Introduction

2.04 Each pident starts with Generic Feature Information and a Program Administration Section. The Generic Feature Information includes the generic and PR number and provides in feature and package parameters that are set for the particular assembly. The Program Administration Section includes program corrections (if any) that are included in the assembly.

B. Message Head Table

- 2.05 The message head table follows the Program Administration Section and contains the following:
 - Pident message identification
 - Relative address of associated tables
 - Types of RC messages allowed (NEW, CHG, OUT)

- Number of associated keyword units
- Number of MSB (messge storage buffer)

Note: The message storage buffer is a storage facility in the call store area. The MSB consists of the MSB proper and auxiliary areas. The MSB proper consists of a one-word entry assigned for each keyword in the RC message, and the MSB auxiliary area is for use when the one-word entry is inadequate to store the keyword data.

C. Definition and Reference Tables

- 2.06 At the end of the pident program section, a list of symbol references and a list of macros are provided. Each symbol (name) appearing in the pident is listed and referenced to the page and line numbers of its occurrences. Figure 2(A) provides a layout of the definition and reference tables and Table D gives a description of the columns on the definition and reference tables.
- 2.07 A separate list of macros are provided as part of the cross reference list (following the symbol references) providing the count or times a macro is used within a pident and the page and line reference numbers applicable to the macro call. Refer to Fig. 2(B) for an example. The columns of the cross reference list and their corresponding descriptions are listed in Table D.

INPUT SECTION

2.08 The input section, following the message head table, always contains a keyword table. It may also contain an internal keyword table, a keyword equivalence table, and a keyword collision table. The input section is concerned with reading in data. A description of these four tables is given in paragraphs A through D.

A. Keyword Table

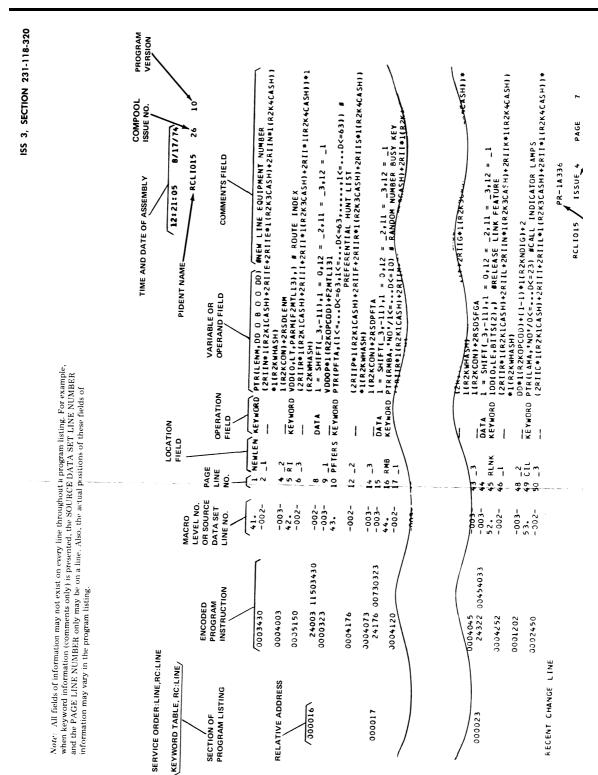
2.09 The keyword table contains all valid keywords for the message in condensed hased form. This hashed form is a number created by combining the keyword character codes according to an arbitrary but fixed algorithm to achieve a compact code for the total keyword.

RC18 & RC16 Output Messages / #1 ESS

TABLE C
DESCRIPTION OF COLUMNS ON THE LINE FORMAT

FIELD OF INFORMATION	DESCRIPTION
RELATIVE ADDRESS	Octal representation of the displacement of the instruction or data that has been translated by SWAP on that line of the pident listing. In each pident, the 6-digit address begins with 000000; the number identifies the relative location of the word.
ABSOLUTE ADDRESS	(Core loaded program). The actual program store or call store address of the instruction.
ENCODED PROGRAM INSTRUCTION	Octal representation of the encoded program instruction or data. In No. 1 ESS the 13 digits to the right of each relative address represents an instruc- tion or data containing 37 binary bits. In No. 1 A ESS there are 8 digits to the right of the relative address containing 24 binary bits. In No. 1 ESS the three preceding 12-bit words are used to derive this data statement. In No. 1A ESS only two 12-bit words are used. This column may also contain relative octal addresses of the EQU psuedo-operation. In the case of recent change PRs this column may contain the octal representation of a macro expansion.
ADDRESS VALUE	One alpha character that denotes that the address computation for the instruction may not be complete and tells what the loader will do to complete the address computation for the instruction. "R" denotes relative address meaning that the loader does nothing to the address. "L" denotes local (absolute) address meaning that the loader will add in the starting address of the pident to the address. "V" denotes "extern" or a reference to an address external to this program which is resolved by going through the transfer vector table. It tells the loader to find the symbols address in the transfer vector table. "X" denotes "direct" or a reference to an address external to this program which should be resolved without an entry in the transfer vector table.
SOURCE DATA SET LINE NUMBER	A decimal number which specifies a line in the source data set. The lines of the source data set are continuously numbered from the beginning to the end of the program listing and are used when modifying a source program. The format of the source data set line number may be XXXXX.XXXX or XXXXX., where X is a decimal digit.
MACRO GENERATION LEVEL NUMBER	A level number which identifies which level of nested macros generated this line of the program listing. (A nested macro is a macro that is called by another macro.) The macro level number format is "XXX." where X is a decimal digit. Those lines which include macro level numbers are macro or EPL expansions produced by the assembler program from the last source data set line preceding the macro expansion line.
PAGE LINE NUMBER	Decimal digits denoting a line (01 to 50) per a page of a program listing.
LOCATION FIELD	A symbolic name (address) of a program point by which other instruc- tions may refer to the instruction named. Such symbolic names are required at points to which control is transferred.
OPERATION FIELD	Symbolic instruction operation code, macro name, or pseudo-operation instruction code. The instruction code can consist of one to seven letters; macro names can be larger.
VARIABLE FIELD or OPERAND FIELD	Contains the parameters related to the operation designated in the operation field. The variable or operand field is formed by several subfields.
COMMENTS FIELD	Begins with a number sign (\pm) and contains comments, remarks, or references made by the programmer to aid in the maintainability and usability of the program listing.

RC18 & RC16 Output Messages / #1 ESS

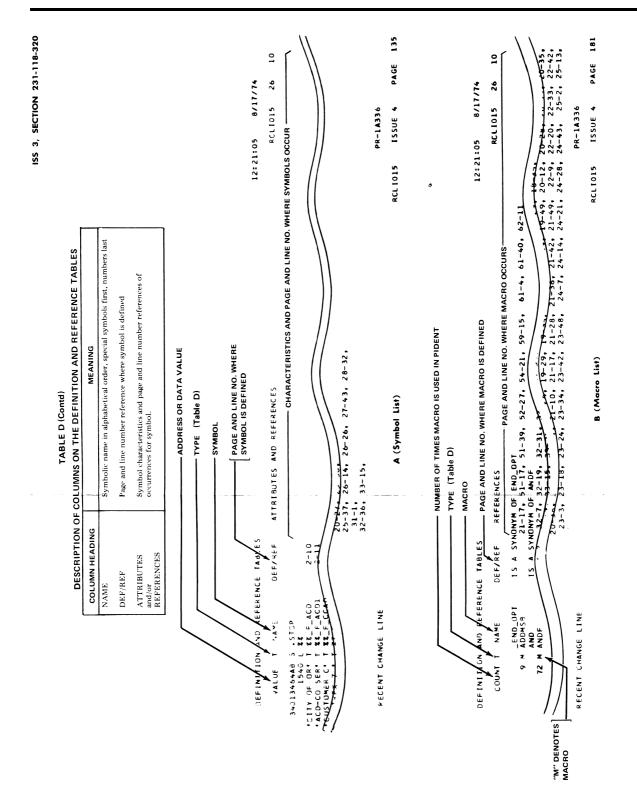


RC18 & RC16 Output Messages / #1 ESS

TABLE D DESCRIPTION OF COLUMNS ON THE DEFINITION AND REFERENCE TABLES

COLUMN HE	ADING	MEANING
VALUE (for symbol list) or COUNT (for macro list)		Octal address or other data value dependent upon type of data as specified under the "TYPE" column.
т	T	Type of symbol as specified below:
	Α	ABSOLUTE — fixed location of data in memory
	В	ATTRIBUTE NAME — the name assigned to a characteristic of a symbol
	C	CALL STORE TABLE — data address of first word in call store table, scatable or block
	F	PROGRAM STORE TABLE — data address of first word in program store table, scatable or block
	G	$\operatorname{GLOBAL}-\operatorname{pseudo-operation}$ for global label with this name in the listing
	I	ITEM — address of data bits in data layout
	J	TRUTH VALUE or BOOLEAN — A value that a boolean expression is to have TRUE (FALSE) if its value is non-zero (zero)
	к .	FUNCTION NAME — The name of a user or predefined function. This name may be used as an operand of an arithmetic expression just as a symbol or number might be used (ie, x(ae) causes any unqualified numbers, including letters A-F, to be interpreted as hexadecimal).
	L	${ m LOCAL-pseudo-operation}$ for local label with this name in listing
	M	MACRO NAME — instead of column "VALUE", all macros are listed in a separate list where the first column is "COUNT" (number of times macro is used in pident)
	N	PURE NUMBER — A string of numerics that represent an N-bit integer.
	О	MACHINE OPERATION — An instruction which when decoded provides an action for the machine hardware to perform.
	P	ITEM IN REGISTER — The layout of items. This is the name of the item which makes up the subdivision of the layout for a particular register.
	R	REGISTER — central control register name
	S	SEQUENCE SYMBOL — A source level symbol preceded by an underscore which has meaning only to the assembler
	Т	TEXT — character string in quotes
	U	UNDEFINED SYMBOL — miscellaneous symbol
	V	EXTERN — pseudo-operation for an external location label in the vector table for a transfer to a point outside this program
	x	DIRECT — pseudo-operation for an external location label
	Y	ARRAY NAME — Name of an array of arithmetic values defined by using the ARRAY pseudo-operation
	Z	PSEUDO-OPERATION NAME — An ESS assembly statement that provides instruction to the assembler or loader and does not generate machine code

RC18 & RC16 Output Messages / #1 ESS



RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

- 2.10 For each keyword, the table contains the following:
 - Hashed form of the keyword.
 - Either a description of the data that must accompany the keyword (including limited number of basic checks, such as type of data, number of digits, etc) or an index into the supplementary data assembly Table for more complex forms of keyword data.
 - An indicator for allowing the keyword to be repeated in segmented messages.

B. Internal Keyword Table

2.11 The internal keyword table is actually part of the keyword table table is located at the end of the keyword table. The internal keyword table contains no user inputted keywords. It is merely a list of keywords used internally by the message pident. The purpose of this table is to assign MSB locations and truth value indexes to the internal keywords.

C. Keyword Equivalence Table

- 2.12 The keyword equivalence table will be located immediately after the internal keyword table. Entries in this table are made in order to specify any alternate names for keywords listed in the keyword table. The table contains:
 - Hash code of alternate name
 - Keyword index of the equivalent keyword in the keyword table.

D. Keyword Collision Table

- 2.13 The keyword collision table is located immediately after the keyword equivalence table. Entries are made into this table when two keywords have the same hash code. Each entry contains:
 - Complete character codes for the collision keyword.
 - Keyword index of the keyword in the keyword table.

VALIDITY SECTION

2.14 The validity section of a message pident always contains a data check table, validity (tree) table and a format selector table for each message type (NEW, CHG, OUT or equivalents) allowed by the message. In some cases, the validity tables for CHG messages will be combined with the validity tables for OUT messages. The validity section is concerned with whether the input data is valid. Following is a description of the three basic types of validity tables.

A. Data Check Table

- 2.15 The data check table contains the following:
 - The detailed checks performed on keyword data which includes comparing keyword data with translation values, checking translations assignments, and testing for specific values of keyword data.
 - The algorithms to calculate internal keyword
 value

B. Validity (Tree) Table

- 2.16 The validity (tree) table specifies and contains the following:
 - The valid combinations of keywords and keyword data by establishing a validity tree (each node on the tree is provided a name).
 - A keyword (KW) value for each node and a set of tests that must be passed to set the truth value bit for a particular node. Some nodes cause RC18 messages to be printed if they fail the set of tests. In such cases, reference to the proper (NEW, CHG, or OUT) validity (tree) error dictionary (paragraph 2.29) will be required to determine the error.

C. Format Selector Table

2.17 The format selector table contains a set of Boolean operations to be performed upon the truth value bits to derive the format selector bits which control the execution of the translation format table.

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

TRANSLATION FORMAT SECTION

2.18 The translation format section always has a translation format table and may have a change transition screening table, change keyword list, and a change YES/NO keyword list. This section is concerned with the format of the translation output of the message. A description of the tables contained in this section is given in subparagrphs A through D.

A. Translation Format Table

- 2.19 The translation format table contains the data format description of the translator words changed by the RC message. Each item in a word of the affected translator is described in terms of size, displacement, auxiliary block word number, etc. The new contents of the translator words and items are described in terms of:
 - (a) truth value bits,
 - (b) format selector bits,
 - (c) data in the MSB (message storage buffer),
 - (d) data in the MSB auxiliary area.

The control pident RCTF (RC new pass translation format) PR-1A304 uses this data to build translaton data from MSB data and the MSB auxiliary area data. The control pident RCCH (RC change pass translation) PR-1A305 uses it to reconstruct MSB data, MSB auxiliary area data, and truth value bits from old data in translations.

B. Change Translation Screening Table

- 2.20 The change translation screening table is used only on CHG or OUT messages during the reconstruction of keyword value from old translation data under control of the translation format table. This screening table is checked each time a keyword is reconstructed. Entries into this table can indicate any of the following:
 - Old and new data can be compared (failure to match results in RC18 CHGER).
 - The truth value bit can be unconditionally set for the keyword.

 The new value of a keyword can be stored in the MSB location of an internal keyword and the old data from translations can be stored in the MSB location of the keyword.

C. Change Keyword List

2.21 Entries into the change keyword list are used at the end of the change pass of the format table to move data from one keyword to another in preparation for a new pass of the format table and allows truth table bits for the moved data to be updated.

D. Change YES/NO Keyword List

- 2.22 Entries must be made into the change YES/NO keyword list for all YES/NO and NO/data keywords if CHG or OUT are valid forms for the message. The entries are used at the end of the CHG pass to correctly set the truth value bit for these keywords, as follows:
 - If a value for the keyword was input, then the new value is used.
 - If the keyword has not been input and the old data from translations indicates NO, then the truth value bit remains 0.
 - If the keyword has not been input and the old data incidates YES or data, then the truth value bit is set and the old data value is entered in the MSB.

ERROR DICTIONARY SECTION

- 2.23 Following the data tables in a PR is the pident error dictionary. The error dictionary section is designed to be used in conjunction with the following three types of output messages:
 - RC18 INPUT—An input error has been detected while the program was checking the keyword unit.
 - RC18 VALER—A validity error has been detected—specifically, a fatal node has failed. Reference to the validity (tree) error dictionary is required to determine the error.
 - RC18 XLER—A translation error has been detected:

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

- (a) A nonexistent head table or translator
- (b) An invalid address for a required pointer associated with an auxiliary block in translation
- (c) Erroneous pointer to a word that should be located in a translation auxiliary block (lies beyond the end of the block).
- 2.24 The error dictionary section consists of some or all of the following parts, as appropriate for the particular message.

A. Keyword Table

2.25 The keyword table located in the error dictionary section contains all valid keywords for a message as well as the data type associated with each keyword (YES/NO, data/NO, data) and a brief description o the keyword. The keyword table will be located at the beginning of the error dictionary section.

B. Synonymous Keyword Table

2.26 As indicated by its name, the synonymous keyword table contains a list of keywords and their equivalents. The use of either of two equivalent keywords will be accepted by the ESS and mean the same thing. The synonymous keyword table follows the keyword table in the error dictionary section.

C. Keyword (Table) Error Dictionary

2.27 The keyword (table) error dictionary is referenced when the table identifier in the RC18 INPUT error message is KWT. The dictionary contains index, data type, and error description of associated keyword units by separate NEW, CHG, and OUT tables, as appropriate for the message.

D. Data Check Table Dictionary

2.28 The data check table dictionary contains indices, labels, and descriptions of each data check performed on associated keyword units. The tables are provided with prefixed NEW, CHG, and OUT to the table title, as appropriate for the message.

E. Validity (Tree) Error Dictionary

2.29 The validity (tree) error dictionary contains nodes, labels, and descriptions of each validity check performed. The tables are provided with prefixed NEW, CHG, and OUT to the table title as appropriate for the message. The validity (tree) error dictionary is referenced when a test established by the validity (tree) table has failed, resulting in an RC18 VALER output message.

F. Format Selector List

2.30 The format selector list contains a list of format selector names and their corresponding equivalencies. This error dictionary is generated only for format selectors of type NEW messages.

G. Validity Tree Diagram

- 2.31 The validity tree diagram (applicable to earlier generic programs only) provides a graphical description of valid keyword unit combinations for use in RC messages. The diagram uses node symbols to associate validity checking procedures with the cause and source of rejected input messages.
 - (a) The validity tree processing consists of performing Boolean functions upon previously defined truth values (keyword units, data checks, and validity tree nodes). The validity tree starts with a single word (12 bits) which defines the Boolean functions of the first node. The two functions of a node are:
 - (1) For the node to be true or false

true: the node value is 1

false: the node value is 0

(2) For the node to fail (and, in turn, the message) or not.

Where a node failure causes a message failure (rejection), it is called a **fatal** node. Where a node failure does not in itself result in an input message rejection, the node is called **nonfatal**.

(b) Table E lists the nodes and symbols descriptions used in graphical presentations of validity tree diagrams.

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

TABLE E VALIDITY TREE NODES

SYMBOL*	NODE	NODE = 1 if and only if:	FATAL if and only if:	DESCRIPTION
&	AND			AND (non-fatal)
•	ANDF	all args [†] = 1	Node = 0	AND, Fatal if fail
*	ANDFN		Node = 0 and Sum ≠ 0	AND, Fatal if fail and Sum of args non 0
÷	OR	at least		OR (non-fatal)
	ORF	one arg = 1	Node = 0	OR, Fatal if fail
×	XORF	exactly	Node = 0	Exclusive OR, Fatal if fail
#	XORFN	one arg = 1	Node - 0 and Sum ≠ 0	Exclusive OR, Fatal if fail and sum of args‡ non 0
@	BCF	all args have same value	Node = 0	Biconditional, Fatal if fail
0 (÷)	OPT (for any)	any arg of list 1 = 1, or all args of list 2 = 0	Node = 0	Optional: FORANY (list 1) OPTIONAL (list 2)
0 (&)	OPT (if all)	all args of list 1 = 1, or all args of list 2 = 0	Node = 0	Optional: IFALL (list 1) OPTIONAL (list 2)
a (b)	IMP (a⇒b)	a=1=b	a=1 and b=0	Implies

*These symbols were used on validity tree diagrams that appear in some message pident PSs. However, they are not reliable because they have not been updated, and are being removed from all PRs.

‡Where the **sum of args** is the node value sum of the adjacent node located on all branches extending from a specified or referenced node.

targs is arguments

SHARED PIDENTS (RCTS AND RCSI)

- 2.32 Shared pident RCTS (RC table subroutine) PR-1A319 and pident RCSI (RC shared information) PR-1A320 will sometimes be referenced while using other pidents in connection with problem analysis procedures.
- 2.33 Pident RCTS consist of special purpose subroutines and are used by RC message pidents—input (through pident RCSI), data check, and format—and by control pident RCWL (RC work list) PR-1A307. The subroutines provide functions that are too specialized to be handled efficiently by the macro languages of input, data check, and format.
- 2.34 Pident RCSI consists of five data tables accessed by several code pidents during RC message processing. The tables are as follows:
 - (1) **Message Head Table:** The message head table contains a vector for each RC message pident. This table is used by control pident RCIG (RC and general control) PR-1A300.
 - (2) TAG and Assignment Table: Each table entry provides the structure of a translator in terms of head table, number of bits in selector and index, primary translation word (PTW) and head table unassignment codes, and power of expansion values. These tables are used by routine RSGTG in control pident

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

RSUB (RC subroutines) PR-1A309 when generating the program store address (TAG) of a translator.

- (3) Parameter Table: The table is a transfer vector (TV) table for the parameters used in format checking of the keyword data. This table is used by control pident RCKI (RC keyword input) PR-1A302.
- (4) TV Table for Supplementary Input Tables: Each table entry contains relocatable addresses of two supplementary input table assembly routines. The table is used when the entry in the keyword table of message pident points to a supplementary input table assembly routine in RCSI. This table is used by control pident RCKI.
- (5) Supplementary Input Table: Each table entry consists of two or more 12-bit words that describe the data of a keyword and indicate whether to store the data in the MSB or MSB auxiliary area. Any of the RC message pidents can point to entries in this table. This table is used by control pident RCKI, using the TV table. Routines in this table may point directly to other routines in the table without having to refer back to the TV table.

3. TTY ACKNOWLEDGMENT (TACK) SYSTEM RESPONSE

- **3.01** A TACK is an *immediate* response from the ESS after a specific input control character is received. The term immediate means that no other output can intervene. There are seven TACK categories:
 - (1) Acceptable
 - (2) Line Error
 - (3) Message Error
 - (4) Busy
 - (5) Invalid Heading
 - (6) Internal Program Error
 - (7) Timeout

A. Acceptable TACK (OK)

Acceptance of an input is indicated either by a carriage return and line feed or a printed OK immediately following an input control character. A carriage return and line feed occurs following an input check (/), a line cancel (#), or a message cancel (&) character. An OK is printed immediately following an end-of-message (! or .) or an end-of-segment (%) character. The carrier return and line feed or OK indicate that the system is available for use, that no format errors were detected in the input, and that the information is entered into the system as instructed. In the case where a line cancel (#) or message cancel (&) character was inputted, the acceptable TACK response indicates that the required action has been taken. Examples of the acceptable TACK follows:

> RC:LINE: ORD 4804 TN 9227468 OE 01302103 LCC 1FR!OK

> > RC18 0 4804 ACPT 6/5 18:20

RC:LINE;CHG: ORD 1765 TN 9222828 OE 01004502 ICP!OK

RC18 0 1765 ACPT 7/19 13:55

RC:TWOPTY;OUT:/ ORD 3614/ TN 9220130!OK

RC18 35 3614 ACPT 5/30 11:54

RC:SCLIST:/ ORD 5 TN 8920102

CODE 1.DGS 2228811%OK

CODE 2.DGS 6247230%OK

CODE 3.DGS 6248588!OF

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

B. Line Error TACK

- 3.03 The line error TACK is printed immediately following the input check character (/) and the end-of-segment character (%) in the following form:
- $\label{eq:aa} \textbf{aa} = \textbf{Two letter TACK code indicating type of} \\ \textbf{error detected (Table F)}$
- bb = Character (column) number, in octal, of the first character which may be in error. The character (column) spaces of the printout line are numbered from left to right, including spaces, beginning with 1.

Upon determining the error, the erroneous input line can be retyped correctly into the system without affecting the preceding portion of the message. Examples of line error TACK are shown below:

RC:LINE;OUT:/ ORD 1211/ TN 922165/MD 12

(more decimal digits expected in TN)

RC:LINE:/ ORD 1505/ TN 9228950/ OE 01014501/ TTC 7/IA 05

(invalid character received for TTC)

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

TABLE F

LIST OF ERROR CODES APPEARING IN CORRECTABLE ERROR TACKS* AND IN RC18 OUTPUT MESSAGES

ERROR CODE	OCTAL CODE	TYPE OF ERROR
DE	00000021	Data exceeds allocated buffer (too many arguments for the multiargument keyword)
DL	00000020 Data lower bound violated (data too small)	
DS	00000030	Discrete data not valid (data does not equal any of the specific valid values)
DU	00000007 00000011 00000016	Data upper bound exceeded (data too large)
	00000017	Data greater than or equal to upper bound
I	INED	Invalid backspace when just after a carriage return and no character yet typed into the buffer. The — (backspace) appears as ← on some versions of IOT keyboards.
IA	00000015 00000033	Invalid character; alphanumeric (letter or number) expected
	00000046	Invalid character; hyphen or alphanumeric expected
IB 00000014 Invalid character; binary digit expected		Invalid character; binary digit expected
IC INED Illegal character (not one of the set of characters recognized as value change program)		Illegal character (not one of the set of characters recognized as valid by the recent change program)
ID 00000012 Invalid character; decimal digit expected		Invalid character; decimal digit expected
IN	00000032	Invalid character; numeric digit expected
Ю	00000013	Invalid character; octal digit expected
IS	00000023	Invalid character, range-hyphen (_) or comma (,) expected
IT	00000044	Invalid TOUCH-TONE" digit
KI 00000000 Keyword invalid for this message		Keyword invalid for this message
KR	00000001	Keyword repeated in message illegally
КТ	00000003 00000027	Keyword unit termination valid or missing where expected (too many characters in fixed-length data fields)
KU	00000045	Keyword unavailable because associated package is not loaded

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

TABLE F (Contd)

LIST OF ERROR CODES APPEARING IN CORRECTABLE ERROR TACKS* AND IN RC18 OUTPUT MESSAGES

ERROR CODE	OCTAL CODE	TYPE OF ERROR
LP	00000026	Left parenthesis missing
МВ	00000036	More binary digits expected
MD	00000034	More decimal digits expected
MK	00000004	More keywords than permitted on one line
MO	00000035	More octal digits expected
RP	00000025	Right parenthesis missing
SE	00000031	Error detected by input subroutine. (See pident RCSI in PR-1A320 or in PR-6A320 for more information)
SI	00000005	Segmenting illegal in this message
YN	00000022	Invalid data for YES/NO keyword

 $^{^{}ullet}$ A correctable error TACK consists of one of these codes followed by a 2-digit decimal column number of the error.

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

C. Message Error TACK (ER)

3.04 ER is the message error TACK and is printed immediately following the ! (end of message symbol) or the % (end of segment symbol). The ER TACK indicates that a noncorrectable error

is contained in the message and that the entire message or message segment is removed from the system. The message must be reinputted after correction as indicated by the RC18 message which follows the ER. Examples of the ER TACK follow:

RC:LINE: ORD 3504 TN 9620400 OE 02317101 LCC 1MB!ER

> RC18 0 3504 INED IC 00000077 5/30 11:53

LC00010006 INP 00000017 STA **00**0000003

RC:LINE;CHG:
ORD 1287
MLH 5
TER 2
OE 01302001
NEWOE 01302203
TN 9221390
NONH!ER

RC18 0 0 INPUT 00000000 KI 00020006 KWT 00000340 7/3 15:08

RC:LINE: ORD 1987 TN 9220680 OE 02207401 LCC 1MB!ER

> RC18 0 1987 VALER NEW 00000231 7/19 13:53

RC:LINE: ORD 1902 TN 9225196 OE 04614600 LCC 1MB GND!ER

RC18 0 1902 XLER PTA 00000000 CON 00000000 TAG 00000000 UNA 15236210 TIN 17000406 30000000

6/11 5:29

D. Busy TACK (BUSY)

3.05 The BUSY TACK is printed immediately following the !, /, and % input characters. This TACK indicates that the recent change programs

are busy on another channel and that the input must be repeated until an OK TACK is printed. An example of the BUSY TACK message is shown below:

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

RC:LINE;OUT: ORD 1654 TN 9221366!BUSY

RC:LINE;OUT:/BUSY

RC:LINE;OUT:/BUSY

RC:LINE;OUT: ORD 1654 TN 9221366!OK

RC18 0 1654 ACPT 7/19 14:00

E. Invalid Heading TACK (?IH)

3.06 The ?IH is the invalid heading TACK and is printed immediately following a / input character on the first line of a message indicating that an error has been detected in the heading line. Upon correction of the error, the heading fine must be reinputted. An example of the ?IH TACK is shown below:

RC:LIME:/?IH (Line misspelled)

F. Internal Program Error TACK (<INTERR>)

3.07 <INTERR>is the internal program error and is printed immediately following the !, /, and % input characters. The <INTERR> TACK notifies the operator that the program has an internal error and the raw data dumps provided in the RC16 messages can be analyzed to determine the program error. The input message cannot be processed until the program error is corrected. An example of the <INTERR> TACK follows:

RC:LINE:
ORD 305
MLH 355
TER 32
OE 00406501
LCC RXR
CTX 2
TN 8941581
STAH
FREE
MSN 100407!<INTERR>

25 RC16 INTERR 00022106 00000000...00000002

26 RC16 INTERR 00000002 00000001...00000450 27770036 00022106...20000005 00002002 01206347...01176123

G. Timeout TACK (<TIMEOUT>)

3.08 The <TIMEOUT> TACK is printed immediately following an IOT character (except ! and .) in a message that is typed into the system after 45 seconds has elapsed since the last character was typed, or if a tape input is used and the tape has been stopped for 45 seconds. It is necessary to reinput the entire message. An example of the <TIMEOUT> TACK is shown below:

RC:LINE: ORD 13 TN 861 1999 OE 07003703 LCC 1FR <TIMEOUT>

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

4. TTY OUTPUT MESSAGES FOLLOWING THE TACK

4.01 The output message (OM) following the TACK of an RC input message is an RC18 or an RC16 output message. These OMs may indicate acceptance of an input message, may explain errors detected in the input message, or may provide a series of data dumps to analyze problems arising from program errors.

A. RC18 Output Message

- 4.02 Almost all output messages encountered are RC18 messages. An RC18 message may indicate acceptance of an RC input message, or provide data resulting from successful execution of particular messages. Other RC18 messages provide coded data for error diagnosis and/or identification of error sources.
- **4.03** The RC18 output message consists of 21 types which fall into 5 categories as follows:

RC18 MESSAGE TYPE

CATEGORY

Note: RC18 output message types are explained in Table G.

ACPT BRKRC INFO	Nonerror
CHGER DELAY INED MTYPE NOCR OBJT PUNCT	Errors that do not require reference to the PR error dictionary.
INPUT VALER XLER	Errors that require reference to the PR error dictionary.
NAUX NAWL NLLS NOPS NPRC PLUGF	Software problems (system resources exceeded)
PRTER TABER	Errors that pertain to internal generic program

4.04 The format of an RC18 message and the associated explanation are as follows:

MESSAGE INDEX	ORDER NUMBER	RC18 MESSAGE TYPE
RC18 ji	ឃ្លាំឃុំឃុំឃុំឃុំ	### tt
xxx zzzzzzz	XXX ZZZZZZZZ .	XXX ZZZZZZZZ
1st Data Unit	2nd Data Unit	Last Data Unit (Variable Number)

Yi = Message index which is a decimal number 1 or 2 digits and identified the RC message (See Table B).

ท่าทั่ททัก = The 1-to-6 digit (with possible letter prefix) order number assigned to the RC input message (keyword ORD). If no order number was inputted, a 0 will be printed.

ffftt = A 2-to-5 letter
 code identifying
 the RC18 message
 type (See Table
G)

xxx zzzzzzzz = Data units as explained in Table G. The quantity of units may vary from none to a maximum of 5 per line as required for each type message.

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

TABLE G

RC18 OUTPUT MESSAGE TYPES

RC18 TYPE (###tt)	EXPLANATION	xxx zzzzzzz UNITS
ACPT	Message accepted	
BRKRC	Break received (manual inter- ruption of the input program)	
CHGER	Change transition error Mis- match on a CHG or OUT or- der between keyword input and existing translation data	xxx = MSB (Message Storage Buffer) zzzzzzzz = MSB index
DELAY	Error in delayed message, in activation, or deactivation of a delayed message	xxx = SEG — Segmented Message A message of more than one segment cannot be delayed. = ORD — incorrect or no order number specified. In delayed message, the order number must be given, must be less than 262114, and cannot be preceded by a letter character. = ASN Either of two assignment errors: (1) A delayed message entered with an ordernumber that is the same as an existing delayed order. (2) An activation or deactivation order number that does not match any existing delayed order in memory. zzzzzzzz = Not used for this type message.
INED	Input editor error	xxx = IC (invalid character), or = I_[invalid cancel (backspace) beyond the start of the line] zzzzzzzz = aaa000rr, where aaa = American Stan- dard Code for Information Interchange (ASCII) code of the invalid character and rr is the RC (6-bit) code (see Table H). Both aaa and rr are in octal. xxx = LC (Line Column)
		zzzzzzzz = rrrrcccc where rrrr = octal line number 2nd (see Note) and cccc = octal number of the unit invalid character.
		xxx = INP (Input) 3rd zzzzzzzz = Octal column of state table (input) unit

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

TABLE G (Contd)

RC18 OUTPUT MESSAGE TYPES

RC18 TYPE (ffftt)	EXPLANATION	XXX ZZZZZZZZ UNITS					
INED (contd)	Input editor error (contd)	xxx = STA (State) zzzzzzzz = rrrrrrss, where ss = last state and rrrrr = three states preceding last state (each a 2-digit octal number)					
		Note: The line number starts with 1 for the heading	ng line.				
INFO	Information associated with successful execution of an RC message	xxx = NOG (Number Group) zzzzzzzz = Octal number group number	1st unit*				
	inessage	xxx = MBI (Mask Block Index) zzzzzzzz = Octal mask block index number	1st unit*				
		xxx = ADD (Address) zzzzzzzz = Octal address of PS block seized	1st unit+				
		xxx = OCT (Length) zzzzzzzz = Octal length of PS block seized	2nd unit+				
		xxx = HTA (Head Table Address) zzzzzzzz = Octal head table address that must be initialized to the PS block address	3rd unit+				
		*Only the 1st unit appears for RC:NOCNOG and RC:DAMSK messages. For RC:NOCNOG the 1st unit will be the number group (NOG). For RC:DAMSK the 1st unit will be the mask block index (MBI). These messages will then be followed by a standard RC18 ACPT message.					
		+Only the 1st and 2nd units appear for RC:PSBLK and RC:SUBTRAN messages using keywords LNGR, LNGL14, or LNGL23. In all other RC:SUBTRAN messages, two RC18 INFO messages appear automatically: 1st and 2nd units appear in the 1st RC18 INFO message and the 3rd unit appears in the 2nd RC18 INFO message. These messages will then be followed by a standard RC18 ACPT message.					
INPUT	Input error	xxx = Three blank spaces zzzzzzzz = Octal error code used by pident RCKI PR-1A302 (see Table F for a listing of error codes)	1st unit				
		xxx = Error code (2 or 3 letters — see Table F)					

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

TABLE G (Contd)

RC18 OUTPUT MESSAGE TYPES

RC18 TYPE (ffftt)	EXPLANATION	xxx zzzzzzzz UNITS
INPUT (contd)	Input error (contd)	zzzzzzzz = rrrcccc where rrr = octal line number (see Note) and cccc = octal column number of the input error
		xxx = Input Table Identifier:
МТҮРЕ	Message type identifier error (The message type identifier is the second subfield of the second field — the symbol after the first semicolon in an RC message).	
NAUX	Not enough RC auxiliary area available.	
NAWL	Not enough available work list space (reduce size of RC meassage)	
NLLS (No. 1)	Not enough link list scratch available	
NOCR	No carriage return after first line (first two fields) of mes- sage	
NOPS	No program store block (or not enough) available.	
NPRC	Not enough primary RC register available	

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

TABLE G (Contd)

RC18 OUTPUT MESSAGE TYPES

RC18 TYPE (ffftt)	EXPLANATION	xxx zzzzzzz UNITS
ОВЈТ	Message object identifier error (The message identifier, or object, of the message is the first subfield of the second field — the symbol after RC:)	
PLUGF	Plug-up Failure — Message is the result of an attempt to plug-up more lines for service observing than allowed by of- fice parameters. (Refer to Section 231-118-328)	
PRTER	Print Call Error (Internal program error)	
PUNCT	Punctuation Error in Heading Line	
TABER	Table Specification Error (Internal program error)	xxx = TAD (Table Address) zzzzzzzz = 1 + 12-bit PS address where error was detected
VALER	Validity Error	xxx = Validity Type: NEW or C/O (change or out type message) zzzzzzzz = Octal mode number in validity (tree) error dictionary of the applicable PR section
XLER	Translation Error	One or more of the following units will appear:
		xxx = PTA (Primary tag assignment) zzzzzzzz = Octal index into pident RCSI's tag and assignment error dictionary (PR-1A320) in order to identify the translator associated with a problem.
		xxx = CON (Contents) zzzzzzzz = Contents of primary translation word (PTW) if TAG (next unit) is nonzero, otherwise, garbage
		xxx = TAG zzzzzzzz = Tag (address of PTW)

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

TABLE G (Contd)

RC18 OUTPUT MESSAGE TYPES

RC18 TYPE (ffftt)	EXPLANATION	xxx zzzzzzz UNITS
XLER		xxx = UNA — unassigned head table or translator
(contd)		ALG — auxiliary block length exceeded
		AAD — auxiliary block address invalid (aux block does not exist)
		zzzzzzzz = 1 + 12-bit PS address where error was detected
		xxx = TIN (Translation Input) zzzzzzzz = Translation input (for example, directory number or line equipment number)
		xxx = IDX (Index or #) zzzzzzzz = Current truth index in data check table

B. RC18 Interpretation Procedure

- 4.05 The meaning of an RC18 output message may be determined in the following manner:
 - (1) The message index (%i) is used to identify the message by using Table B.
 - (2) The order number of the associated RC input is specified in the ทำทำทำทำ location.
 - (3) The message type ($\ell\ell\ell\ell$ tt) is explained in Table G.
 - (4) If the message type is one of the applicable error types, data unit(s) will identify the cause(s) of the system rejection by use of Table G. If the message type is INFO, the data units

will specify information concerning the memory storage area for the RC input as specified in Table G.

C. ACPT-Type RC18 Message

4.06 The ACPT message indicates that the input message was accepted and executed by the system as specified. An example follows:

RC:LINE;OUT: ORD 4611 TN 9221554!OK

> RC18 0 4611 ACPT 6/5 17:58

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

TABLE H RC 6-BIT AND ASCII CODES ASSOCIATED WITH INPUT TERMINAL CHARACTERS AND OPERATIONS

TELETYPEWRITER CHARACTER OR OPERATION	RC 6-BIT CODE (rr)	ASCII CODE (aaa)	TELETYPEWRITER CHARACTER OR OPERATION	RC 6-BIT CODE (rr)	ASCII CODE (aaa)
0	00	060	U	36	125
1	01	061	v	37	126
2	02	062	w	40	127
3	03	063	X	41	130
4	04	064	Y	42	131
5	05	065	Z	43	132
6	06	066	- (DASH)	51	055
7	07	067	TAB	52	011
8	10	070	:	53	072
9	11	071	;	54	073
A	12	101	SPACE	55	040
В	13	102	,	56	054
С	14	103)	57	051
D	15	104	(60	050
E	16	105	NEW LINE	61	012
F	17	106	RETURN	62	015
G	20	107	!	63	041
Н	21	110	%	64	045
I	22	111	,,	65	042
J	23	112	&	66	046
К	24	113	X OFF	70 (Note 1)	023
L	25	114	RUB OUT	70 (Note 1)	177
M	26	115	VT (Vertical Tab)	70 (Note 1)	013
N	27	116	FORM		
О	30	117	(Form Feed)	70 (Note 1)	014
P	31	120	\$	72	044
Q	32	121		73	137
R	33	122	_(UNDERSCORE)	77	(Note 2)
s	34	123		77	000 (Note 3)
Т	35	124			(Note 3)

Note 1: Note that 70 is the RC code for RUBOUT, X OFF, VT and FORM. All are ignored, so there is no reason to distinguish between them.

Note 2: Any ASCII code not used by the RC program is converted to RC code 77.

Note 3: This combination is used to indicate an input character parity error.

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

D. BRKRC-Type RC18 Message

4,07 The BRKRC message indicates that a manual break of interruption of the input program was received by the system. This would normally occur as—a result of depressing the BREAK key on any input keyboard connected to that channel. The input message must be correctly reinputted into the system. The following example illustrates the use of the BREAK key following the! in the first message and its correct reinput:

E. CHGER-Type RC18 Message

4.08 The CHGER message indicates that there is a discrepancy between a keyword of a CHG or OUT input message and the existing translation information in the system. This could be an error such as attempting to CHG or OUT a feature on a line which does not exist in the translation information for that line or inconsistency in input information. An example of a CHGER message follows:

```
RC:LINE;OUT:
ORD 12
TN 8938338
OE 07402502!ER
```

RC18 0 12 CHGER MSB 00000035 3/28 16:51

F. DELAY-Type RC18 Message

- **4.09** The DELAY message indicates an error in a delayed input message which falls into three categories as follows:
 - SEG-A segmented input message (containing more than one segment using a % character) cannot be delayed.
 - (2) ORD—An incorrect or no order number was in the input message.
 - (3) ASN-An assignment error either by inputting a delay order number that is already in the system or inputting an activation or deactivation order for a delay order number which does exist in the system.

The corrected message must be inputted. Examples of the DELAY message follow:

RC:SCLIST::DELAY:

```
TN 7273081
ADN 3
DGS 5843742%ER
                         DELAY
      RC18 12 26
    7/24 9:28
  (Segmented Message)
RC:LINE;OUT;DELAY:
TN 8684730
CTX 7!ER
  RC18
ORD
            0
                         DELAY
  6/6 10:27
  (No Order Number)
RC:ACT:OUT:
ORD 300!ER
      RC18
                  300
                         DELAY
  ASN
  6/6 10:26
```

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

G. INED-Type RC18 Message

4.10 The INED message indicates an input editor error. The output message contains four

data units explaining the error and its location by use of Tables G and H. Correct any errors present and reinput message. An example follows with an input character parity error and reinput:

RC:LINE: ORD 4722 TN 9226370 OE 03006202 LCC 1MR!ER

> RC18 0 4722 INED IC 00000077 LC 00040003 INP 00000017 STA 00000003 6/5 18:16

RC:LINE: ORD 4722 TN 9226370 OE 03006202 LCC 1MR!OK

RC18 0 4722 ACPT 6/5 18:17

H. INFO-Type RC18 Message

4.11 The INFO message specifies storage location for the input message in up to three data

units. (See Table G). This output message is only used with the RC:NOCNOG, RC:DAMSK, RC:PSBLK, and RC:SUBTRAN input messages. As examples the following are presented:

RC: DAMSK: GSZ 154 TERS(10, 11, 12, 13, 14, 15)%OK TERS(16, 17, 18, 19, 20, 21, 22, 23, 24, 25)!OK

RC18 58 0 INFO MBI 00000035 3/20 11:06

RC18 58 0 ACPT 3/20 11:06

RC:PSBLK: ORD 131 NEW R12!OK

RC18 8 131 INFO ADD 02163432 OCT 000014 6/13 3:14

RC18 8 131 ACPT 6/13 3:14

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

I. INPUT-Type RC18 Message

- 4.12 The INPUT message indicates that an error was made in the input message to the system. The following procedure should be used to locate and analyze the error:
 - (1) Determine which RC message is in error by using the message index (ii) and, if specified, the order number (miniminin) in the RC18 message. Refer to Table B for the listing of RC message indexes. For those messages identified by error code DE, KI, KT, LP, RP, or SI the error index (3rd data unit) should be ignored. The line and column information can still be used to locate the characters in error and, usually, the definition in Table F of the error code is self-explanatory.
 - (2) Using Table G, locate INPUT type message. Convert the second data unit octal line and column (character) number (zzzzzzzz) to decimal numbers. (See Fig. 3.) Use the decimal line and column numbers to locate the characters in error. Begin line counting with the heading line as number 1. Count column (character) numbers from left to right, including spaces, beginning with 1. The error code (xxx) (Table F) defines the type of error.

	OCTAL NUMBER								
TENS	UNITS DIGIT								
DIGIT	0	1	2	3	4	5	6	7	
0	0	1	2	3	4	5	6	7	
ı	8	9	10	Ш	12	13	14	15	D E C
2	16	17	18	19	20	21	22	23	M A L
3	24	25	26	27	28	29	30	31	L N
4	32	33	34	35	36	37	38	39	U M B E
5	40	41	42	43	44	45	46	47	Ř
6	48	49	50	51	52	53	54	55	
7	56	57	58	59	60	61	62	63	

(OCTAL 34 = DECIMAL 28, DECIMAL 34 = OCTAL 42)

Fig. 3—Octal-Decimal Conversion Table

- (3) Determine the input message error and the corrections necessary to input an acceptable message.
- (4) Reinput the corrected RC message for acceptance by the system.
- (5) If the second data unit error code (xxx) is not self-explanatory, determine by using the 3rd data unit which input table error dictionary should be used to obtain more details.
 - If the 3rd data unit (xxx) is KWT, locate the keyword table error dictionary associated with the applicable input message program PR- (Table G).
 - If the 3rd data unit (xxx) is SIT, locate the supplementary input table error dictionary in PR-1A320 (pident RCSI).

The 3rd data unit's octal number (zzzzzzzz) is the index number into the appropriate error dictionary. The error dictionary will then provide data types and a more detailed error description. Table I lists the codes and symbols used in the data type column of an error dictionary and a description of these codes.

An example of an INPUT output message containing an incorrect TN follows:

RC: LINE: ORD 0005 OE 00000003 TN 5733141!ER

RC18 0 5 INPUT

00000031 SE 00040012 SIT 00000257 4/16 10:50

J. MTYPE-Type RC18 Message

4.13 The MTYPE message indicates an error in the message type identifier which is located on the RC heading line in the right subfield of the identification (2nd) field (usually NEW, CHG or OUT). Reinput corrected message. An example follows with an incorrect ICP input:

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

TABLE I
INPUT DATA TYPE SYMBOLS IN PROGRAM LISTINGS

Data Type	Meaning
D	Decimal digit (0-9)
О	Octal digit (0-7)
В	Binary digit (0-1)
С	Alphabetic character (A-Z)
A	Alphanumeric character (A-Z and 0-9)
CHAR	Also used for character when character can be alphabetic or numeric
D	Variable number of decimal digits with leading zeros optional
00	Variable number of octal digits with leading zeros optional
N	Digit from 2 to 9 (used as first digit of office code; example, NXX)
X	Digit from 0 to 9 (used as second and third digits of office code; example, NXX)
/	Exclusive OR operation. Example, YES/NO means type YES or NO.
.,	Denotes a literal to be inputted. Example, 'RG' means must input RG.
11	Encloses an optional character. Example, C means alphabetic character is optional.
<=	Less than or equal
,,	Stands for the word through when used in an expression that defines a series where only the first and last members of the series are stated in the expression, such as MIN1,, MIN5

Note: These codes and symbols are used to describe keyword data in the keyword lists, the keyword table error dictionaries, and the supplementary input table error dictionary.

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

RC:LINE;ICP: ORD 412 IN 5735030 OE 06410702!ER

RC18 0 0 MTYPE 7/3 6:29

K. NOCR-Type RC18 Message

4.14 The NOCR message indicates that there was no CR and NL following the heading line (first two fields). Reinput message correctly. An example follows:

> RC:LINE;OUT:ORD 2333 TN 9221914!ER

> > RC18 0 2333 NOCR 7/19 14:01

L. OBJT-Type RC18 Message

4.15 Th OBJT message indicates an error in the message object identifier which is located on the RC heading line in the left subfield of the identification (2nd) field (LINE, TWOPTY, MLHG, SCLIST, etc). Reinput corrected message. An example follows of an incorrect (misspelled) LINE input:

RC:LIST: ORD 1 TN 572087!ER

> RC18 53 0 OBJT 4/11 9:08

M. PUNCT-Type RC18 Message

4.16 The PUNCT message indicates a punctuation error in the heading line (first message line). Correct error and reinput message. An example (omitted colon) follows:

RC:LINE;OUT ORD 33 TN 5349745 OE 02114300!ER

RC18 0 0 PUNCT 7/24 14:54

N. VALER-Type RC18 Message

- 4.17 Validity errors (VALER) are detected by **programmed checks** that are performed to assure that the combination of the input keywords and the keywords data are consistent with the existing data base (translation memory). These validity errors appear as **fatal node failures** and are reported by the VALER type RC18 output message (Table G).
- **4.18** Three examples of errors which would be reported as validity errors are:
 - (1) Attempting to assign a line already assigned or to unassign a line which is not assigned
 - (2) Attempting to assign an unworkable combination of features to a line
 - (3) Attempting to use RC CHG messages for transitions not permitted by the system.
- 4.19 An example of a VALER message follows:

RC:LINE;OUT: ORD 1099 TN 9223732!ER

> RC18 0 1099 VALER C/O 00000245 7/19 14:01

In this example, the PR-1A336 Change Validity (Tree) Error Dictionary, RC:LINE; lists the error description of Node 245 as TN must be assigned.

O. XLER-Type RC18 Message

4.20 The XLER message reports that instructions contained in the RC input message cannot be accomplished with the existing system translation data, which could be inconsistent input data (Table G). Such inconsistencies may also result in a VALER or RC16 INTERR output message. The RC TAG and assignment error dictionary (PR-1A320) list three types of translation errors reportable by the XLER output message as follows:

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

		problems or resource limitations (Table G). These types of RC18 output messages are:				
UNA	Head table, translator, or	MESSAGE	CORRECTIVE ACTION			
	subtranslator is unassigned	NAUX	RC update or cardwrite.			
ALG	Auxiliary Length error; index exceeds auxiliary block length or attempts to access a word beyond	NAWL	Reduce size of message.			
	the end of an auxiliary block.	NLLS	$Cut \ in \ spare \ program \ store \ module.$			
AAD	Invalid auxiliary block address.	NOPS	Increase size of translator.			
	-type translation output message n error in input data or a procedural	NPRC	RC update or cardwrite.			
	uence of related input messages.	PLUGF	Remove some lines from service observing.			
messages	and AAD-type of translation output indicate an error in translation e further analysis.	PRTER	Receive assistance from loca TAC, SCCS, or PECC location.			
	nder of the RC TAG and assignment onary provides tabulated information	TABER	Receive assistance from local TAC, SCCS, or PECC location.			
(PR-1A320) that principal use of translator that using the prima provided in the	d assignment table in pident RCSI identifies all translations. The the tabulations is to identify a is associated with a problem by ry TAG assignment (PTA) index RC18 output message.	4.27 The PRTER RC18 message results from a print call error where the print routine expects a particular character(s) for printing but the character is not received (Table G). INFO RC18 messages that provide information associated with an accepted input message but did not provide the expected information to the print routine is				
•	ole of the XLER message follows:	one situation where a PRTER message results. Parity check failures are also causes for the PRTER message. Assistance should be requested from				
RC:LINE;OUT:		the local TAC, SCCS, or from the PECC location.				

TN 5346256 OE 00114320!ER

> RC18 0 12 XLER PTA 00000000 CON 00257027 TAG 02547225 ALG 31200444 TIN 17400406 #1000000 #10000000 7/10 6:40

- $\textbf{4.25} \quad \text{The RC16 INTERR output message is explained}$ in paragraphs 4.30 through 4.38.
- P. NAUX, NAWL, NLLS, NOPS, NPRC, PLUGF, PRTER, and TABER Types of RC18 Message
- There are eight types of output messages which pertain to internal generic program

The TABER RC18 message results from a table specification error, such as an invalid

RC18 message (Table G). An invalid message is generated when an incorrect value is loaded into the buffer for printing. A typical cause for TABER messages is that an associated table is expected to contain particular information, but does not. Assistance should be requested from the local TAC, SCCS, or from the PECC location.

Q. RC16 Output Messages

RC16 output messages indicate that the system has encountered problems which apparently are caused by internal errors in the program. Detection of such an error results in a series of seven RC16 INTERR message segments which consists of a varying quantity of eight digit numbers. The segments consists of:

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

RC16 INTERR MESSAGE NUMBER 1. CC Registers 2. States & Poi 3. Input Contro 4. Truth Value 5. Format Selec 6. First Part of 1 7. More PDS	QUANTITY	6 10 5 16 4 14 21	inforn in the	ation to de	etermine tl rogram. A	ne problem	hould provide s encountered e of the RC16	
MESSAGE NUMBER	RC:LINE: ORD 305 MLH 355 TER 32 OE 00406 LCC RXH CTX 2 TN 89415 STAH FREE MSN 1004	5501 R	ar>					
1	25		TERR 00000000	00000000	01177152	15177765	00000002	
2	25	04100000	TERR 00001002 00207334	00022100 00210007	00022153	00022300	00022420	00022432
3	25		TERR 04000006	00000002	00000014	00000001		
4	26	00000461	00000000	00000231 00000110				
5	26		TERR 00000240	00000000	01177100			
6	26	00022100		00003062 11400002				
7	26	00000002 27770036	00022106	00000000 15177765 00000560	00000002	00022124	01301001	20000005
	Number of Minutes a the hour							

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

R. RC16 Interpretation Procedure

4.30 RC16 message data analysis requires ability to interpret data in the program listings.

First RC16 INTERR Printout

4.31 The first RC16 INTERR printout message segment consists of six words that provide the octal contents of central control registers at the time an internal error is detected. The sequence of the register content words for are:

FFFFFFF KKKKKKK LLLLLLL XXXXXXXX YYYYYYYY ZZZZZZZZ

The L register word is most important because it contains the return address and thus provides the code pident location at which the error was detected. By proceeding to the location in the specified pident pointed to by the L register, the description of the reason for the INTERR output message is provided. Also, the same description allows interpretation of the other associated registers, as well as the use of the remaining segments of the message.

- 4.32 Although the contents of the central control registers vary in accordance with the specific INTERR call, the programming requirements used for the RC system permit some general statements about the probable content of the registers.
 - (a) L Register: Address of the client program that called the INTERR routine. The correct interpretation of the location pointed to by the L register is the most critical step in the interpretation of the message printout segments. In some instances, the L register word provides all required information. In other instances, some or all of the remaining printout segments

must be interpreted. The L register's use within the control pident RSUB is of special importance because RSUB consists of shared subroutines referenced by other code pidents. For this case, L normally indicates the error identity, but does not specify when or the location where the error occurred. For this case, reference is required to the state words and push down scratch to determine the error location.

- (b) X Register: In control pidents RCKI, RCVC, RCFI, RCCH, or RCTF, the X register typically contains the 12-bit address in the table where the INTERR error occurred. Thus, isolation of the error is permitted to a 12-bit word in the message pident.
- (c) F, K, Y, and Z Registers: Use the L register and pident to determine contents.

Second RC16 INTERR Printout

4.33 The second RC16 INTERR printout message segment consists of state and pointer words. The following information shows ten call store words starting with address 17406 through 17417.

R2PSW1 R2PSW2 R2PDS R2NAPDS R2MSB R2MSBAUX R2NAMSBA R2WL R2NAWL R2EWL

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

The four words that provide information of the greatest significance are as follow:

- R2PSW1 Code pident currently in control
- R2NAPDS Points to the next available word of the push down scratch
- R2NAMSBA Specifies next available word in the MSB auxiliary area
- R2NAWL Specifies the next available work list location.

Where the error is identified but not the error location, use C(R2NAPDS) -1 to C(R2PDS), which is all used push down scratch, to construct a history of events leading up to the internal error. Since the scratch system provides client addresses and in some cases the contents of the central control registers when a client called a subroutine, the problem can be identified by working backward from the last used push down scratch entry at C(R2NAPDS) -1/

Note: The R2PDSMSB (points to RC PDS and MSB) program store location provides the size of the push down scratch area (R2PDS and R2NAPDS-128 words) and the size of the message storage buffer (R2MSB) and its associated message buffer auxiliary area (R2MSBAUX and R2NAMSBA) in the left half of the program store word and the starting address of the call store block in the right half of the program store word. The call store block is used exclusively by the RC message interpretation.

Third RC16 INTERR Printout

- 4.34 The third RC16 INTERR printout message segment provides five input control words located in call store with addresses starting with 17450 as follows:
 - 17450 ASCII character last processed
 - 17451 Input editor state
 - 17452 Line buffer unload pointer
 - 17453 Current line
 - 17454 Current column.

- 4.35 The ASCII to RC character table in pident RCIE is used to convert the 7-bit ASCII character code received on the I/O channel into the 6-bit character code used internally by the RC programs.
- 4.36 The state table for input editor is used to obtain subroutine codes (index to ISUBRT) and next state index which points the program to the proper character subroutine for processing I/O characters.

Fourth and Fifth RC16 INTERR Printout

4.37 The fourth and fifth RC16 INTERR printout message segments provide a Boolean history of input, data check, validity, and format selection of the RC input message. By inspecting the truth value table (fourth printout segment), keyword units that were received can determine which data checks were true, and which validity tree nodes were true. The format selector (fifth printout segment) provides the selected paths by using a format based on the input and translation data.

Sixth and Seventh RC16 INTERR Printout

4.38 The sixth and seventh RC16 INTERR printout message segments provide a part of the 128-word push down scratch. The sixth segment prints the first 14 words of the 128-word push down scratch. The seventh segment prints 21 words: 14 words prior and seven words after R2NAPDS (second RC16 INTERR fourth word) which is located within the 128-word push down scratch. In some circumstances, T-READ procedures are required to determine additional words in the 128-word push down scratch.

5. ABBREVIATIONS AND ACRONYMS

5.01 Table C and D provide a description of information fields located within the line format (Fig. 1) and the definition and reference tables (Fig. 2) which will not be included in the abbreviations. Other abbreviation items are as follows:

AAD-Auxiliary block address invalid (auxiliary block does not exit)

ACPT—Message accepted

ADD-Address

Page 36

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

	AGN-Assignment	INP—Input
	ALG-Auxiliary block length exceeded	INPUT-Input error
	ASCII-American Standard Code for Information	INTERR—Internal error
	Interchange	IO—Invalid character - octal digit expected
	ASN—Assignment	I/O—Input/Output
	BRKRC—Break received	IS-Invalid character - range hyphen (-) or comma
	CHGER-Change error	(,) expected
- ,	C/O-Change or out type message	IT-Invalid TOUCH-TONE® digit
	CON—Contents	KI-Keyword invalid for this message
	CR-Carriage return	KR-Keyword repeated in message illegally
	DE—Data exceeds allocated buffer	KT-Keyword unit termination invalid or necessary when expected
	DELAY-Error in delay message	
	DL-Data lower bound violated (data too small)	KU-Keyword unavailable because associated package not loaded
_	DS-Discrete data not valid (data does not equal any of the specific valid values)	KW-Keyword
	DU-Data upper bound exceeded (data too large)	KWT-Keyword table
	ER-Error	LC-Line column
	HTA—Head table address	LP-Left parenthesis missing
	I—Invalid backspace	MB-More binary digits expected
	IA-Invalid character - alphanumeric (letter or	MBI-Mask block index
	number) expected	MD-More decimal digits expected
-	IB—Invalid character - binary digit expected	MK-More keywords than permitted on one line
	IC—Illegal character	MO Mara actal digita arrested
	ID—Invalid character - decimal digit expected	MO—More octal digits expected
	IDX—Index	MSB—Message storage buffer (a storage facility in the call store area)
	IM—Input message	MTYPE-Message type
	IN-Invalid character - numeric digit expected	NAUX-Not enough auxiliary block available
_	INED-Input editor error	NAWL-Not enough available work list space
	$\label{eq:info-information} INFO-Information \ (associated \ with \ execution \ of \\ message)$	NEW-New

Page 37

RC18 & RC16 Output Messages / #1 ESS

SECTION 231-118-320

NL-New	line	(previously	line	feed)	

NLLS-Not enough link list scratch available

NOCR-No carriage return after two fields of messages

NOG-Number group

NOPS-No program store block

NPRC—Not enough primary recent change registers available

OBJT-Message object identifier error

OCT-Octal

OM-Output message
ORD-Order number
PG-Program generic

Pidents-Program identification

PLUGF—Plug-up failure
PR—Program listing

PRTER-Print call error

PTA-Primary tag assignment

PTW-Primary translation word

PUNCT-Punctuation error in heading line

RC-Recent change

 $\begin{array}{lll} RCCH-Recent & change & pass & translation & format \\ (PR-1A305) & & \end{array}$

RCDY-Recent change delay order (PR-1A308)

RCFI-Recent change format interpretation (PR-1A306)

RCIE-Recent change input editor (PR-1A301)

RCIG—Recent change initialization and general control (PR-1A300)

RCKI-Recent change keyword input (PR-1A302)

RCSI-Recent change shared information (PR-1A320)

RCTF-Recent change new pass translation format (PR-1A304)

RCTS—Recent change table subroutine (PR-1A319)

RSUB-Recent change subroutines (PR-1A309)

RCVC-Recent change validity check (PR-1A303)

RCWL-Recent change work list (PR-1A307)

RCxx—One pident per recent change input message. (See Table A)

RP-Right parenthesis missing

SCCS-Switching Control Center System

SE-Error detected by subroutine (see pident RCSI in PR-1A320 for more information)

SEG-Segmented

SI-Segmenting illegal in this message

SIT-Supplementary input table (PR-1A320)

SMB-Segmented

 $\mathbf{STA}\!-\!\mathbf{State}$

 $SWAP-Switching\ Assembly\ Program$

TABER—Table specification error (internal progam

error)

TAC-Technical assistance center

TACK-TTY acknowledgment

TAD-Table address

TAG-Address of primary translation word

TIN-Translation input

TTY-Teletypewriter

TV-Transfer vector table

UNA-Unassigned head table or translator

Page 38

RC18 & RC16 Output Messages / #1 ESS

ISS 3, SECTION 231-118-320

VALER-Validity error

XLER-Translation error

YN-Invalid data for YES/NO keyword

!-Exclamation point - end of message symbol

.-Period - end of non-RC message

 $\%\!-\!\text{Percent sign}$ - execute and save common data

/-Slash - input check

\$-Dollar sign - line cancellation

&-Ampersand - message cancellation.

Page 39 39 Pages

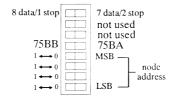
Explanation of Nortel FMT-150 CRT Functions

Setup Procedure for CRT Interface

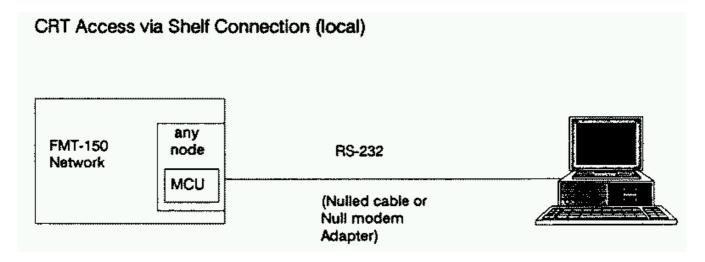
The FMT–150 CRT interface is the most powerful method of monitoring, controlling, and troubleshooting a FMT–150 system. It is typically accessed through a standard CRT terminal (VT100 compatible), either locally or through a dial–up modem. Many terminals and most personal computers can emulate this standard. The addition of an optional Centralized Access Maintenance and Monitoring System (CAMMS) will also provide for a centralized monitoring interface for up to sixteen FMT–150 systems. Alarms and status points of up to sixteen nodes in the system can also be accessed from any site equipped with a NT7H90xx Maintenance Control Unit (MCU).

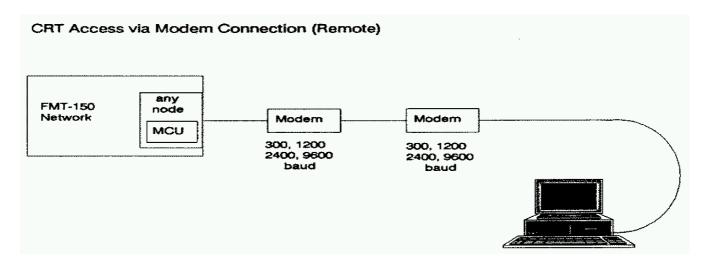
Maintenance Control Unit Installation

The MCU has DIP switches to determine the node's address (1–16). They are set as follows:



The baud rate must also be determined. For local access, it would be 9600 baud (the fastest). For remote access, the rate would be dependent on the modem being used.





Connection to the MCU is made using RS–232. The port on the back of the MCU takes a standard male DB–25 connector. The other end will be determined by the connection port on the particular piece of equipment.

Note: The pin–outs for the RS–232 connector and other information concerning the operations of the Hayes Smartmodem 1200 can be found in the *FMT–150 Quick Reference Guide* under "CRT Operations".

Some of the indications that should be seen on the front of the modem are:

- HS High Speed (1200 baud)
- AA Auto-Answer
- TR Terminal Ready (recognized DTR)
- MR Modem Ready

If the **TR** signal is not there, the modem is not recognizing the MCU. Check the cable connections and cable type; ensure the telephone line is properly connected; make note of the assigned phone number.

For more information see Northern Telecom Publication (NTP) 321–3211–202.

CRT Configuring

Before an operator can log on, the CRT terminal must be configured for proper communication. The basic settings required are as follows:

Terminal Communication Parameters

- VT100 emulation (VT102 and VT220 also accepted)
- 7 or 8 data bits
- 1 or 2 stop bits
- No parity
- No local echo (full duplex)
- Baud rate (1200 for most modems, 9600 for direct connect)

Note: The CRT interface will support both 7 and 8 data bit transmission selectable by the user via switch #8 of the MCU's DIP switch.

```
DIP #8 = '0' --> 7 data bits selected / 2 stop bits DIP #8 = '1' --> 8 data bits selected / 1 stop bit
```

Printer Connection

If a printer is going to be used for logging of alarms, certain protocol settings might have to be changed so that XON/XOFF from the terminal to the FMT-150 might be passed. The printer will only record the alarm/status messages, and not the full-screen displays from the FMT-150. Also, the alarm logger feature must be enabled for the printer to operate.

Logging On

The way in which the connection is made to the FMT-150 will determine what will be seen on the screen of the CRT. If there is a direct connection to the FMT-150, the terminal should be in the "on-line" mode. If the connection is made through a modem, after normal dial-up procedures, there should be a CONNECT prompt. For both ways of connection, the next procedure is the same. The Carriage Return key (<CR>) is pressed three times until a "Enter Terminal Type:" prompt appears on the screen. After the terminal type is selected (#1 for DEC VT-100), a message "Waiting for initialization ..." will be displayed. Next, will come the "Login:" prompt. The operator input would be the password, if it has been set. If there is no password, then the <CR> is pressed, and the operator is logged on.

Nortel FMT-150 Log In Procedures (MCU NT7H90CA)

Steps	Dialogue	Comments
1.	<cr> <cr> <cr></cr></cr></cr>	Enter carriage returns until the message "Querying Terminal Type" appears.
	1 - DEC VT100	
	2 - NT Meridian 6000 (Procom or Crosstalk with VT100 emulation)	
	3 - Tandy 200 (running Telecom)	
	F4 - NT CAMMS MDU	
	Enter Terminal Type:	
	# <cr></cr>	Type number (1, 2, 3, or press F4) corresponding to terminal type.
3.	Login: <password> <cr></cr></password>	Enter user password.
		Note: If password has not been previously selected, or if this is an initial logon into a new system, the password is defaulted to a <cr>.</cr>

Entering Commands

The commands that can be entered cover a wide variety of areas as detailed on the following pages; alarm, switching, configuration, and maintenance.

The command tables are read left to right. Entries are made from each column until a <CR> is reached. Whenever a letter is presented in the command table, it means type that letter on the keyboard. A pound sign (#) refers to a *number* that must be entered. After entering a full command

string (terminated by a <CR>), the requested information will be displayed on the CRT's screen. All numerical entries must be followed by a "space" character (<SP>). All commands are terminated with a carriage return (<CR>).

Help Screens

There are hundreds of possible command strings available. Whenever there is doubt about what character(s) may be entered next, simply press the space bar. A list of legal entries will be displayed at the bottom of the screen (for example, try: "a c m l s ?"). To request additional information about these entries, type a question mark (?). A brief help screen will be displayed explaining the available options at that point in the command string.

Keyboard Shortcuts

If you wish to repeat a string of commands, or enter commands very similar to it, you may use the right-arrow key. This will repeat your previous command line, piece-by-piece. Individual elements may be replaced as it goes by.

Command elements may be deleted with either the left-arrow or backspace keys.

For a detailed explanation of commands and parameters see NTP 231-3211-301.

Remote Loopback

There are two different types of loopbacks available with the FMT-150:

Low Speed Loopback

A low speed loopback (DS-1 or DS-1C), must be inititated at the near-end location. In this case, "near-end" is defined as the location where the signal is to be received for testing. The low speed loopback process will internally loop individual tributaries back to the user for signal testing.

Note: During a loopback condition, any traffic associated with the tributary is interrupted; therefor, caution should be used when performing low speed remote loopbacks.

Translator Loopback

A translator loopback (DS-3 or STX), must be initiated at the far-end location via the CRT which can be accessed from any location in the system. In this case, "far-end" is defined as the location opposite from which the signal is to be received for testing.

To perform a translator loopback the standby translator of the DS-3 stream to be tested must be looped back. At the opposite location, the corresponding translator must be in the active position (a translator force switch operation may be required) for the loopback process to occur.

Note: Although only one of the translators (the standby translator) in a pair will display the actual loopback condition, both translators (that is, the actual DS-3 signals) are looped back. This means that during a translator loopback condition, any traffic associated with the translator (or DS-3) is interrupted. As a result, caution should be used when performing remote loopbacks at the translator level.

Explanation of Nortel FMT-150 CRT Functions: Alarm Commands

> Command Inputs	Comments
a	System-level alarm screen.
a # o	Optical Tx/Rx unit-level alarm screen.
a # t	Translator module–level alarm screen.
a # m	DM-13 multiplexer-level alarm screen.
a # C	Common equipment–level and customer input/output points alarm screen.

⁻End-

Notes (Only valid for MCU NT7H90CA)

- * is ALL nodes.
- # is the Node Number, from 1 to 16, followed by a space.
- #₁ is the point selected from *Table 10-A to F* in the FMT-150 NTP, section 321-3211-104.
- #2 is the point selected from Table 10-A to F in the FMT-150 NTP, section 321-3211-104.
- #3 is the threshold value, from 1 to 65,534, followed by a space.
- $\#_{bc}$ is the Type-B / Type-C Threshold, from 2 to 9,999, followed by a space.
- # is the condition to be inserted, from 1 to 7, followed by a space.
- $\#_{ci}$ is the Customer Input Point, from 1 to 2, followed by a space.
- $\#_{co}$ is the Customer Output Point, from 1 to 16, followed by a space.
- $\#_{\alpha}$ is the Group Number, from 1 to 7, followed by a space.
- $\#_{_{C\!C\!D}}$ is the General Point, from 100 to 128, followed by a space.
- $\#_m$ is the DM-13 Number, from 1 to 3, followed by a space.
- # is the Power Supply Unit Number, from 1 to 4, followed by a space.
- $\#_{pe}$ is the Performance Event Number, from 1 to 4, followed by a space.
- ${\rm \#_{_{\rm S}}}$ is the Signal Degrade Threshold, from 6 to 10, followed by a space.
- $\#_+$ is the Translator Number, from 1 to 3, followed by a space.
- $\#_{_{\mathrm{x}}}$ and $\#_{_{\mathrm{U}}}$ are Synchronous Transport Signal (STX) line numbers, from 1 to 3, followed by a space.

Explanation of Nortel FMT-150 CRT Functions: Switching Commands

> Command Inputs	Comments
S	Display switch status for local node or node last displayed.
s #	Display switch status for node #.
s # t # _t (a or b) f o	Operate force switch on translator unit.
s # t # _t (a or b) f r	Release force switch on translator unit.
s # t # _t (a or b) b o	Operate loopback on translator unit.
s # t # _t (a or b) b r	Release loopback on translator unit.
s # t # _t l o	Operate lockout on translator unit.
s # t # _t l r	Release lockout on translator unit.
s # m # _m	Display switch status of DM-13.
$s # m #_m f g #_g o$	Operate force switch on DM-13 LS Tx/Rx module.
$s # m #_m f g #_g r$	Release force switch on DM-13 LS Tx/Rx module.
s # m # _m f h (a or b)	Force switch on DM–13 HS Tx/Rx module from A–to–B or B–to–A.

⁻continued-

Explanation of Nortel FMT-150 CRT Functions: Switching Commands

> Command Inputs	Comments
$s # m #_m l g #_g o$	Operate lockout on DM-13 LS Tx/Rx module.
$s # m #_m lg #_g r$	Release lockout on DM-13 LS Tx/Rx module.
s # m # _m l h (a or b) o	Operate lockout on DM-13 HS Tx/Rx module.
s # m # _m l h (a or b) r	Release lockout on DM-13 HS Tx/Rx module.
s # m # _m 110	Operate lockout on DS-1 Tx/Rx standby module.
$s # m #_m l 1 r$	Release lockout on DS-1 Tx/Rx standby module.
$s # m #_m lco$	Operate lockout on DS-1C Tx/Rx standby module.
s # m # _m lcr	Release lockout on DS-1C Tx/Rx standby module.
s # m # _m 120	Operate lockout on DS-2 Tx/Rx standby module.
s # m # _m 12 r	Release lockout on DS-2 Tx/Rx standby module.
$s + m +_m b +_g +_g o$	Operate loopback on DS-1C tributary.
$s + m +_m b +_g +_g r$	Release loopback on DS-1C tributary.
s # m # _m do	Disable remote DS-1C tributary.
s # m # _m dr	Enable remote DS-1C tributary.
s # i # _{ci}	Clear customer input (1 or 2) at node #.
s # 0 # _{co} s	Set customer output (manual/latched) at node #.
s # 0 # _{co} C	Clear customer output (manual/latched) at node #.
s * o # _{co} s	Set customer output (manual/latched) at all nodes.
s * o # _{co} C	Clear customer output (manual/latched) at all nodes.

-End-

> Command Inputs	Comments
С	Display all nodes, customer input/output point names, and status of alarm logger.
сае	Enable alarm logger.
c a d	Disable alarm logger.
c i # "name"	Name a customer input point.
c o # "name"	Name a customer output point.
c d #yy #mm #dd	Set date (YY/MM/DD): #yy is last 2 digits of year, followed by a space. #mm is month, followed by a space. #dd is day, followed by a space.
c t #hh #mi	Set time (HH/MM): #hh is hour, followed by a space. #mi is minute, followed by a space.
c p "oldpass" "newpass"	Change password from "oldpass" to "newpass".
c s "system ID name"	Name system ID.

⁻continued-

Notes

To change the default password of <CR>, double quotes "" must be entered for "oldpass".

> Command Inputs	Comments
c n *	Display all nodes, customer input/output point names, and status of alarm logger.
c n #	Display translator, customer status, and output M-Net bridging.
c n # n "name"	Name node #.
cn# t# _t re	Enable received status/alarm overhead.
cn# t# _t rd	Disable received status/alarm overhead.
cn# t# _t te	Enable transmitted status/alarm overhead.
cn# t# _t td	Disable transmitted status/alarm overhead.
cn# t# _t be	Enable blue signal insertion.
cn# t# _t bd	Disable blue signal insertion.
cn # t # _t pe	Enable parity correction.
cn # t # _t pd	Disable parity correction.
cn # t # _t s # _s	Set translator signal degrade threshold (1E-6 to 1E-10)
cn # b * e	Enable M–Net bridging at head hub.
c n # b * d	Disable M-Net bridging at head hub.
cn # b # _x # _y e	Bridge STX $\#_{_{\mathbf{x}}}$ to STX $\#_{_{\mathbf{y}}}$.
c n # b # _x # _y d	Bridge STX $\#_{_{\mathbf{x}}}$ from STX $\#_{_{\mathbf{y}}}$.
c n # m # _m m	Reset monitor and control unit.
c n # m # _m l	Clear line failure minor alarm caused by incoming DS-2 frame loss.
cn# ml	Learns out any removed module position.

⁻continued-

>	
Command Inputs	Comments
cgo $\#_{co}$ (i $\#_{c}$ or a) (rorlorn $\#$ or s $\#$) ca	Add/insert MCU type condition for remote/local sites or node/site #.
cgo $\#_{co}$ (i $\#_{c}$ or a) (rorlorn $\#$ or s $\#$) cc $\#_{1}$	Add/insert MCU type condition for remote/local sites or node/site #.
cgo $\#_{co}$ (i $\#_{c}$ or a) (rorlorn $\#$ or s $\#$) ce $\#_{1}$	Add/insert MCU type condition for remote/local sites or node/site #.
cgo $\#_{co}$ (i $\#_{c}$ or a) (rorlorn $\#$ or s $\#$) cg $\#_{1}$	Add/insert MCU type condition for remote/local sites or node/site #.
cgo $\#_{co}$ (i $\#_{c}$ or a) (rorlorn $\#$ or s $\#$) ch $\#_{1}$	Add/insert MCU type condition for remote/local sites or node/site #.
cgo $\#_{co}$ (i $\#_{c}$ or a) (rorlorn $\#$ or s $\#$) cs $\#_{1}$	Add/insert MCU type condition for remote/local sites or node/site #.
c g o $\#_{co}$ (i $\#_{c}$ or a) (r or 1 or n $\#$ or s $\#$) g $\#_{gp}$	Add/insert general point condition for remote/local sites or node/site #.
c g o $\#_{co}$ (i $\#_{c}$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_{m}$ (1, 2, or c) a	Add/insert DS-1/2/1C standby-type condition for remote/local sites or node/site #.
c g o $\#_{co}$ (i $\#_{c}$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_{m}$ (1, 2, or c) c $\#_{1}$	Add/insert DS-1/2/1C standby-type condition for remote/local sites or node/site #.
c g o $\#_{co}$ (i $\#_{c}$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_{m}$ (1, 2, or c) e $\#_{1}$	Add/insert DS-1/2/1C standby-type condition for remote/local sites or node/site #.
c g o $\#_{co}$ (i $\#_{c}$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_{m}$ (1, 2, or c) g $\#_{1}$	Add/insert DS-1/2/1C standby-type condition for remote/local sites or node/site #.
c g o $\#_{co}$ (i $\#_{c}$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_{m}$ (1, 2, or c) h $\#_{1}$	Add/insert DS-1/2/1C standby-type condition for remote/local sites or node/site #.
c g o $\#_{co}$ (i $\#_{c}$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_{m}$ (1, 2, or c) s $\#_{1}$	Add/insert DS-1/2/1C standby type condition for remote/local sites or node/site #.
cgo $\#_{co}$ (i $\#_{c}$ or a) (rorlorn $\#$ or s $\#$) m $\#_{m}$ g $\#_{g}$ a	Add/insert group number type condition for remote/local sites or node/site #.
cgo $\#_{co}$ (i $\#_{c}$ or a) (rorlorn $\#$ or s $\#$) m $\#_{m}$ g $\#_{g}$ c $\#_{1}$	Add/insert group number type condition for remote/local sites or node/site #.
cgo $\#_{co}$ (i $\#_{c}$ or a) (rorlorn $\#$ or s $\#$) m $\#_{m}$ g $\#_{g}$ e $\#_{1}$	Add/insert group number type condition for remote/local sites or node/site #.

⁻continued-

	> Command Inputs	Comments
c g o # _{co} # _g g # ₁	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_m$ g	Add/insert group number type condition for remote/local sites or node/site #.
c g o # _{co} # _g h # ₁	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_m$ g	Add/insert group number type condition for remote/local sites or node/site #.
c g o # _{co} # _g s # ₁	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_m$ g	Add/insert group number type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_m$ m	Add/insert DM-13 type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_m$ m	Add/insert DM-13 type condition for remote/local sites or node/site #.
c g o # _{co} e # ₁	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_m$ m	Add/insert DM-13 type condition for remote/local sites or node/site #.
c g o # _{co} g # ₁	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_m$ m	Add/insert DM-13 type condition for remote/local sites or node/site #.
c g o # _{co} h # ₁	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_m$ m	Add/insert DM-13 type condition for remote/local sites or node/site #.
c g o # _{co} s # ₁	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) m $\#_m$ m	Add/insert DM-13 type condition for remote/local sites or node/site #.
c g o # _{co}	(i# _c or a) (r or l or n# or s#) s a	Add/insert SCU type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) s e $\#_1$	Add/insert SCU type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) s h $\#_1$	Add/insert SCU type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) s s $\#_1$	Add/insert SCU type condition for remote/local sites or node/site #.

⁻continued-

Notes

n# is the Node Number, from 1 to 16, followed by a space.

 ${\tt s\#}$ is the Site Number, from 1 to 16, followed by a space.

	> Command Inputs	Comments
c g o # _{co}	$(i\#_{c} \text{ or a})$ (r or l or n# or s#) o (a or b)	Add/insert optical type condition for remote/locate sites or node/site #.
c g o # _{co}	$(i\#_{c} \text{ or a})$ (r or l or n# or s#) o (a or b)	Add/insert optical type condition for remote/locate sites or node/site #.
c g o # _{co} e # ₂	$(i\#_c \text{ or a})$ (r or l or n# or s#) o (a or b)	Add/insert optical type condition for remote/locate sites or node/site #.
c g o # _{co} g # ₂	$(i\#_c \text{ or a})$ (r or l or n# or s#) o (a or b)	Add/insert optical type condition for remote/locate sites or node/site #.
c g o # _{co} h # ₂	$(i\#_c \text{ or a})$ (r or l or n# or s#) o (a or b)	Add/insert optical type condition for remote/locate sites or node/site #.
c g o # _{co} s # ₂	$(i\#_{c} \text{ or a})$ (r or l or n# or s#) o (a or b)	Add/insert optical type condition for remote/locate sites or node/site #.
cgo# _{co} b) a	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) t $\#_t$ (a or	Add/insert translator type condition for remote/local sites or node/site #.
c g o # _{co} b) c # ₂	(i $\#_c$ or a) (r or l or n $\#$ or s $\#$) t $\#_t$ (a or	Add/insert translator type condition for remote/local sites or node/site #.
c g o # _{co} b) e # ₂	$(i\#_c \text{ or a})$ (r or l or n# or s#) t $\#_t$ (a or	Add/insert translator type condition for remote/local sites or node/site #.
c g o # _{co} b) g # ₂	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) t $\#_t$ (a or	Add/insert translator type condition for remote/local sites or node/site #.
c g o # _{co} b) h # ₂	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) t $\#_t$ (a or	Add/insert translator type condition for remote/local sites or node/site #.
c g o # _{co} b) s # ₂	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) t $\#_t$ (a or	Add/insert translator type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) p $\#_p$ a	Add/insert PSU type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) p $\#_p$ c $\#_2$	Add/insert PSU type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) p $\#_p$ e $\#_2$	Add/insert PSU type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) p $\#_p$ g $\#_2$	Add/insert PSU type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) p $\#_p$ h $\#_2$	Add/insert PSU type condition for remote/local sites or node/site #.
c g o # _{co}	(i $\#_c$ or a) (r or 1 or n $\#$ or s $\#$) p $\#_p$ s $\#_2$	Add/insert PSU type condition for remote/local sites or node/site #.

⁻continued-

> Command Inputs	Comments
c g p	Displays configuration global performance event alarms.
cgpb# _{bc}	Set type–B / type–C threshold.
cgpe# _{pe} ph# ₃	Define event # as parity errors, including a threshold value, for current hour interval.
c g p e # _{pe} p d # ₃	Define event # as parity errors, including a threshold value, for current day interval.
cgpe# _{pe} pu# ₃	Define event # as parity errors, including a threshold value, for current untimed interval.
$cgpe\#_{pe}ah\#_{3}$	Define event # as type-A seconds, including a threshold value, for current hour interval.
$cgpe\#_{pe}$ ad $\#_3$	Define event # as type-A seconds, including a threshold value, for current day interval.
cgpe# _{pe} au# ₃	Define event # as type-A seconds, including a threshold value, for current untimed interval.
cgpe# _{pe} bh# ₃	Define event # as type-B seconds, including a threshold value, for current hour interval.
c g p e # _{pe} b d # ₃	Define event # as type-B seconds, including a threshold value, for current day interval.
cgpe# _{pe} bu# ₃	Define event # as type-B seconds, including a threshold value, for current untimed interval.
cgpe# _{pe} ch# ₃	Define event # as type-C seconds, including a threshold value, for current hour interval.
cgpe# _{pe} cd# ₃	Define event # as type-C seconds, including a threshold value, for current day interval.
cgpe# _{pe} cu# ₃	Define event # as type-C seconds, including a threshold value, for current untimed interval.
$cgpe\#_{pe}fh\#_{3}$	Define event # as frame losses, including a threshold value, for current hour interval.
$cgpe\#_{pe}fd\#_{3}$	Define event # as frame losses, including a threshold value, for current day interval.
cgpe# _{pe} fu# ₃	Define event # as frame losses, including a threshold value, for current untimed interval.
cgpe# _{pe} lh# ₃	Define event # as frame loss seconds, including a threshold value, for current hour interval.
$cgpe\#_{pe}ld\#_{3}$	Define event # as frame loss seconds, including a threshold value, for current day interval.
cgpe# _{pe} lu# ₃	Define event # as frame loss seconds, including a threshold value, for current untimed interval.
cgpe# _{pe} eh# ₃	Define event # as parity error seconds, including a threshold value, for current hour interval.
c g p e # _{pe} e d # ₃	Define event # as parity error seconds, including a threshold value, for current day interval.
cgpe# _{pe} eu# ₃	Define event # as parity error seconds, including a threshold value, for current untimed interval.
cgpu	Start untimed interval.

-End-

Explanation of Nortel FMT-150 CRT Functions: Performance Commands

> Command Inputs	Comments
p # i c h	Performance at node # for current hour interval.
p # i c d	Performance at node # for current day interval.
p#icu	Performance at node # for current untimed interval.
p#ilm	Performance at node # for last minute interval.
p # i l h	Performance at node # for last hour interval.
p#ild	Performance at node # for last day interval.
p#ilu	Performance at node # for last untimed interval.
p # t # _t a	Display translator A performance at screen node #.
p # t # _t b	Display translator B performance at screen node #.
p # o a	Display optics A performance screen at node #.
p # o b	Display optics B performance screen at node #.

⁻End-

Explanation of Nortel FMT-150 CRT Functions: Maintenance Commands

> Command Inputs	Comments
m r *	Reset all nodes (global reset).
m r #	Reset node # (display does not change).
m t # o	Operate test of customer input/output points.
m t # r	Release test of customer input/output points.

⁻End-

Notes

After executing a local or global MCU reset, the message "PROCESSOR CRASH" will appear at the bottom of the CRT's screen. As a result, the user will have to log back into the system. In addition, a global MCU reset will clear all "names" and "settings" previously defined (i.e. node, system ID, customer inputs/outputs, time and date).

Explanation of Nortel FMT-150 CRT Functions: Logout Procedure / Display Network Status

Logout Procedure

> Command Inputs	Comments
1	The user is now logged out of the system.

⁻End-

Display Network Status

> Command Inputs	Comments
n	Network status screen.

⁻End-

Nortel DMS-100 Line-to-Line Translations

Description

When a call originates from a line, the associated line tables are read and interpreted. The call then enters the screening tables where digit analysis begins. After some general pre–screening, or pre–translation is performed, the call may progress into more detailed screening based on NPA/NXX digits to determine the path into the designated routing tables for defining the final destination or termination of the call.

Operation

Line—to—line translations can be traced using a simplified block diagram representing the major functions within the translation process, as shown in the following figure:

The lines tables contain information about the originator of the call in a DMS-100 switch. These tables have three primary functions:

- Establish the hardware function and specify the hardware location for each line.
- Indicate the type of ringing codes used or options and features assigned to each line.
- Provide the next logical step in translation.

The screening tables contain the information used to analyze the digits that the DMS-100 switch receives. This screening process tests the digits dialed prior to continuing to the next routing stage, to determine, for example, whether this call is local or non-local.

The screening tables establish the call type based on the digits received. The three basic call types are:

- Operator Assisted (OA)
- Direct Dial (DD)
- No Prefix (NP)

The routing tables route the calls to their final destination. The information found in these tables dictates how and where a call will be completed, or if the call will route to a recorded announcement or treatment.

<u>Translations Table Flow for Line-to-Line Translations</u>

As soon as a line goes off–hook, the associated Line Concentrating Module (LCM) informs the switch that the line went off hook and provides the Line Equipment Number (LEN). The originating line has its LEN stored in table LENLINES. From the data stored in table LENLINES, the switch determines the Directory Number (DN), party and options of the line. If there are features associated with the line, the switch accesses table LENFEAT to determine what these features are.

Field LNATTIDX in table LENLINES points to a tuple in table LINEATTR. The entries in this tuple determine the following:

- The standard pretranslator to route to in table STDPRTCT.
- The Local Call Area (LCA) subtable to use for local call checking.
- The Serving Numbering Plan Area (SNPA) to access in table HNPACONT.

Verification of the call type is done in table LCASCRCN and its subtable LCASCR and table PFXTREAT.

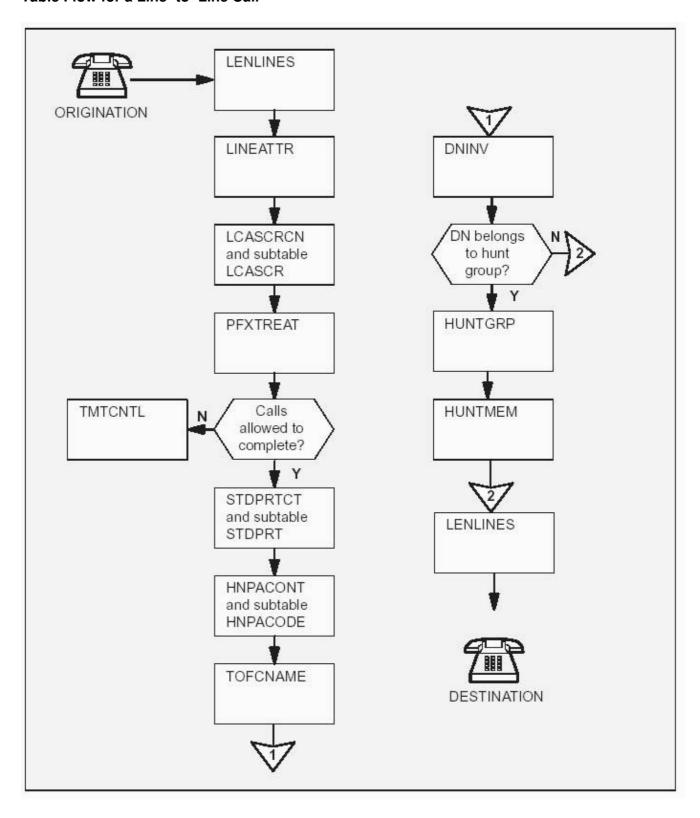
The call then routes to table STDPRTCT and its subtable STDPRT. The pretranslator route selector entered in subfield PRERTSEL determines how call processing progresses. Information on the valid values for this subfield can be located in the table data schema section of this document.

If the entry in this subfield is \mathbb{N} , translations proceeds to table HNPACONT and its subtable HNPACODE. The dialed NXX is located and defined.

Table TOFCNAME defines all the terminating offices in the switch and they are identified by area code and office code. Table DNINV contains the data for all assigned and unassigned DNs. If the DN belongs to a hunt group, tables HUNTGRP and HUNTMEM will be accessed.

The line-to-line translations process is shown in the flowchart that follows on the next page.

Table Flow for a Line-to-Line Call



The following table lists the datafill content used in the flowchart for a call routed between DNs (418) 225-2136 and (418) 225-2164. In the example:

Number Planning Area (NPA) = 418

Calling Line DN = 2252136

Called Line DN = 2252164

Standard Pretranslator = P225

Datafill Example for Line-to-Line Translations

Datafill Table	-		
LENLINES	RCU0 00 0 03 16 S 0 2252136 DT 1 \$		
LINEATTR	1 1FR NONE NT NSCR 0 418 P225 P225 NONE 0 NIL NILSFC NILLATA 0 NIL NIL 00 Y AINRESGRP 0 0 \$		
LCASCRCN	418 P225 (3) MAND N		
subtable LCASCR	225 226		
PFXTREAT	MAND DD N DD UNDT		
STDPRTCT	P225 (1) (65021)		
subtable STDPRT	418 418 N DD 0 NA		
HNPACONT	418 128 0 (68) (1) (2) (0)		
subtable HNPACODE	225 225 DN 418 225		
TOFCNAME	418 225		
DNINV	418 225 2164 H 11 0		
HUNTMEM	11 0 N D 2252164 N		
HUNTGRP	11 418 2252164 DNH N N N RCVD N N N N N 4 \$		
LENLINES	RCU0 00 0 14 04 S 0 2252164 DT 1 \$		
-End-			

Datafilling Office Parameters

The following table shows the office parameters used by line–to–line translations. For more information about office parameters, refer to the *Office Parameters Reference Manual*:

Office Parameters Used by Line-to-Line Translations

Table Name	Parameter Name	Explanation and Action
OFCENG	ACTIVE_DN_SYSTEM	This parameter specifies the type of DNs that can be used in an office. If this parameter is set to "NORTH_AMERICAN", the directory number must use the form NPA-NXX-XXXX. If this parameter is set to "UNIVERSAL", the directory number may vary in length.

Office Parameters Used by Line-to-Line Translations (continued)

Table Name	Parameter Name	Explanation and Action
OFCENG	AIN_ACTIVE	This parameter controls the activation of the Advanced Intelligent Network (AIN). Enter "Y" to activeate AIN software. Enter "N" to deactivate AIN software. If this parameter is set to "N", parameter AIN_OFFICE_TRIGGRP in table OFCVAR is disregarded.
OFCVAR	AIN_OFFICE_TRIGGRP	This parameter is used to subscribe trigger behaviors on an office-wide basis. The entry in field AINGRP in table TRIGGRP is entered here. The default value is "NIL".
-End-		

Table Datafill Sequence

The following table lists the tables that require datafill to implement line—to—line translations for the *calling line*. The tables are listed in the order in which they are to be datafilled:

Datafill Tables Required for Line-to-Line Translations for the Calling Line			
Table	Purpose of Table		
HNPACONT	The Home Numbering Plan Area Control table lists all the home or serving area NPAs for a particular area.		
subtable HNPACODE	The Home Numbering Plan Area Code subtable lists the route treatment or table to which the translation routes for each of the assigned NPAs.		
STDPRTCT	The Standard Pretranslator table lists the names of the standard pretranslator subtables.		
subtable STDPRT	The Standard Pretranslator subtable determines the next stage of translation, based on the range of leading digits.		
LCASCRCN	The Local Calling Area Screening Control table lists the NPA code and Local Calling Area name and its prefix selector.		
subtable LCASCR	The Local Calling Area Screening Code subtable determines from the dialed digits if the call is local or non-local.		
PFXTREAT	The Prefix Treatment table determines the call treatment to which a call is routed.		
LINEATTR	The Line Attribute table provides pointers to screening and billing tables and assigns line attributes for digit analysis.		
LENLINES	The Line Assignment table contains the DN, hardware location, and		

options associated with the calling line.

-End-

The following table lists the tables that require datafill to implement line—to—line translations for the *called line*. The tables are listed in the order in which they are to be datafilled:

Datafill Tables Required for Line-to-Line Translations for the Called Line

Table	Purpose of Table
TOFCNAME	The Terminating Office Name table defines all terminating offices in the switch.
DNINV	The Directory Number Inventory table is a read-only table that contains data for all assigned and unassigned DNs.
HUNTGRP	The Hunt Group table lists data assignments for each hunt group in the switching unit.
HUNTMEM	The Hunt Member table lists the members assigned to the hunt groups.
-End-	

Datafilling Table HNPACONT

The following table shows the datafill specific to line—to—line translations for table HNPACONT. Only those fields that apply directly to line—to—line translations are shown:

Datafilling Table HNPACONT

Field	Subfield or Refinement	Entry	Explanation and Action
STS		0 to 9,999,999	Serving Translation Scheme (STS) Enter a STS or SNPA code.
HNPACODE		See note	Home Numbering Plan Area (HNPA) Code This field is an index into subtable HNPACODE.
			Note: This field does not accept any input.
-End-			

Datafill Example for Table HNPACONT

The following example MAP display shows sample datafill for table HNPACONT:

STS	NORTREFS	NOAMBIGC	RTEREF	HNPACODE	ATTRIB	RTEMAP
418	128	0	(68)	(1)	(2)	(0)

Datafilling Subtable HNPACONT.HNPACODE

The following table shows the datafill specific to line-to-line translations for subtable HNPACONT.HNPACODE. Only those fields that apply directly to line-to-line translations are shown:

Datafilling Subtable HNPACONT.HNPACODE

Field	Subfield or Refinement	Entry	Explanation and Action
FROMDIGS		Numeric (3 digits)	From Digits Enter the number representing a single code or the first in a block of consecutive codes that have the same input data.
TODIGS		Numeric (3 digits)	To Digits If field FROMDIGS represents a single code, enter the same single code as in field FROMDIGS. If field FROMDIGS represents the first number of a block of consecutive numbers, enter the last number in the block.
CDRRTMT		See subfield	Code Type, Route Reference or Treatment This field consists of subfield CD.
	CD	DN	Code Type Enter "DN" for terminating office code and datafill refinements SNPA and NXX.
	SNPA	Numeric (3 digits)	Terminating Serving Numbering Plan Area Enter the SNPA of the called terminating line DN. If the operating company uses screening to intraswitch SNPAs, translation of the dialed digits proceeds to table TOFCNAME, using SNPA and NXX as the key.
	NXX	Numeric (3 digits)	Terminating NXX Enter three digits for the NXX code of the called terminating line DN.

-End-

Datafill Example for Subtable HNPACONT.HNPACODE

The following example MAP display shows sample datafill for subtable HNPACONT.HNPACODE:

FROMDIGS	TODIGS	CDRRTMT
225	225	DN 418 225

Datafilling Table STDPRTCT

The following table shows the datafill specific to line-to-line translations for table STDPRTCT. Only those fields that apply directly to line-to-line translations are shown:

Datafilling Table STDPRTCT

Field	Subfield or Refinement	Entry	Explanation and Action
EXPRTNM		Alphanumeric (up to 8 characters)	External Standard Pretranslator Subtable Name Enter the name defined by the operating company to represent the standard pre- translator.
STDPRT		See note	Standard Pretranslator The field is an index into subtable STDPRT.
			Note: This field does not accept any input.
-End-			

Datafill Example for Table STDPRTCT

The following example MAP display shows sample datafill for table STDPRTCT:

EXPRTNM	STDPRT	AMAPRT	
P225	(1)	(65021)	

Datafilling Subtable STDPRTCT.STDPRT

The following table shows the datafill specific to line—to—line translations for subtable STDPRTCT.STDPRT. Only those fields that apply directly to line—to—line translations are shown:

Datafilling Subtable STDPRTCT.STDPRT

Field	Subfield or Refinement	Entry	Explanation and Action
FROMDIGS			From Digits Enter the digit or digits to be translated.
			If the entry represents a block of consecutive numbers, enter the first number in the block.
TODIGS		Numeric (up to 18 digits)	To Digits If field FROMDIGS represents a block of consecutive numbers, enter the last number in the block.
PRETRTE		See subfield	

-continued-

Datafilling Subtable STDPRTCT.STDPRT (continued)

Field	Subfield or Refinement	Entry	Explanation and Action
	PRERTSEL	N or NT	Pretranslator Route Selector Enter "N" or "NT".
			Enter "NT" for DMS-500 switches only.
	TYPCALL	DD, OA, NP, or NL	Type of Call Enter the type of call:
			DD (Direct Dial) OA (Operated Assisted) NP (No Prefix) NL (nil)
			For Traffic Operator Position System (TOPS) calls, there can be a mixture of 0 and 1 (OA and DD) call types. Enter "NL" for these cases.
	NOPREDIG	0 to 7	Number of Prefix Digits Enter the number of digits that are to be interpreted as prefix digits.
	TRANSYS	NA, IN, or IP	Translation System Enter "IN" if the translation routes to international translations (on a local and toll combined switching unit only).
			Enter "IP" if the translation routes to international partitioned translations (DMS-250 only).
			Enter "NA" if the translation routes to national translations.

-End-

Datafill Example for Subtable STDPRTCT.STDPRT

The following example MAP display shows sample datafill for subtable STDPRTCT.STDPRT:

FROMDIGS	TODIGS	PF	RETRT	E	
418	418	N	DD	0	NA

Datafilling Table LCASCRCN

The following table shows the datafill specific to line–to–line translations for table LCASCRCN. Only those fields that apply directly to line–to–line translations are shown:

Datafilling Table LCASCRCN

	Subfield or Refinement	- 4	Explanation and Action
NPALOCNM		See subfields	
	STS	Numeric (3 digits)	Serving Translation Scheme Enter a serving NPA code for the trunk group.
	LCANAME	(up to 4 characters)	Local Calling Area Name Enter the key to subtable LCASCRCN.LCASCR.
LCASCR		See note	
			Note: This field does not accept any input.
PFXSELEC		Alphanumeric	Prefix Selector Enter the name of the prefix selector that is assigned to subtable LCASCR.
 -End-			

Datafill Example for Table LCASCRCN

The following example MAP display shows sample datafill for table LCASCRCN:

NPALOCNM	LCASCR	PFXSELEC	PFSFOR10
418 P225	(3)	MAND	N

Datafilling Subtable LCASCRCN.LCASCR

The following table shows the datafill specific to line—to—line translations for subtable LCASCRCN.LCASCR. Only those fields that apply directly to line—to—line translations are shown:

${\tt Datafilling\ Subtable\ LCASCRCN.LCASCR}$

Field	Subfield or Refinement	Entry	Explanation and Action
FROMDIGS		Numeric (3 digits)	From Digits Enter the three-digit local NNX code. This number represents a single code or the first in a block of consecutive local NNX codes.

⁻continued-

Datafilling Subtable LCASCRCN.LCASCR (continued)

Field	Subfield or Refinement	Entry	Explanation and Action
TODIGS		Numeric (3 digits)	To Digits If field FROMDIGS represents the first number of a block of consecutive local NNX codes, enter the last NNX code in the block. If field FROMDIGS represents a single local NNX code, enter the NNX code entered in FROMDIGS.
-End-			

Datafill Example for Subtable LCASCRCN.LCASCR

The following example MAP display shows sample datafill for subtable LCASCRCN.LCASCR:

FROMDIGS	TODIGS
225	226

Datafilling Table PFXTREAT

The following table shows the datafill specific to line—to—line translations for table PFXTREAT. Only those fields that apply directly to line—to—line translations are shown:

Datafilling Table PFXTREAT

Field	Subfield or Refinement	Entry	Explanation and Action
TYPLCLCD		See subfields	Type of Call and Local Code This field consists of subfields PFXSELEC, TYPCALL, and LOCAL.
	PFXSELEC	•	Prefix Selector Enter the prefix selector.
	TYPCALL	DD, NP, OA, or NL	Type of Call Enter the type of call:
			DD (Direct Dial) OA (Operated Assisted) NP (No Prefix) NL (nil)

-End-

Datafill Example for Table PFXTREAT

The following example MAP display shows sample datafill for table PFXTREAT:

TYPLC	LCD		UPDTYPCA	TREAT
MAND	DD	N	DD	UNDT

Datafilling Table LINEATTR

The following table shows the datafill specific to line–to–line translations for table LINEATTR. Only those fields that apply directly to line–to–line translations are shown:

Datafilling Table LINEATTR

Field	Subfield or Refinement	Entry	Explanation and Action
LNATTIDX		0 to 31,999	Line Attribute Index Enter the index into table LINEATTR.
SCRNCL		Alphanumeric (up to 4 characters or NSCR)	Class of Service Screening If screening by class of service is required, enter the key to the class of service subtable assigned to the line attribute index. If screening by class of service is not required, enter "NSCR".
STS		Numeric (3 digits)	Serving Translation Scheme (STS) Enter the SNPA assigned to the line attribute index. The STS of an existing tuple cannot be changed.
PRTNM		Alphanumeric (up to 4 characters or NPRT)	Standard Pretranslator Subtable Name If pretranslation of digits is required, enter the key to the standard pretranslator subtable assigned to the line attribute inde If standard pretranslation is not required, enter "NPRT".
LCANAME		Alphanumeric (up to 4 characters or NLCA)	Local Calling Area Screening Subtable Name If screening of local central office codes (NNX) is required, enter the key to the loca calling area subtable assigned to the line attribute index. If screening of local NNX codes is not required, enter "NLCA".

-End-

Datafill Example for Table LINEATTR

The following example MAP display shows sample datafill for table LINEATTR:

LNATTIDX TRAFSNO RESINF	LCC MRSA OPTIONS	CHGCLSS SFC	COST LATANM	SCRNCL MDI	_	STS DGCLNAME	PRTNM FANIDIGS	LCANAME	ZEROMPOS
1 0 Y	1FR NIL AINRESG	NONE NILSFC RP 0 0 \$	NT NILLATA	NSCR 0	0 NIL	418 NIL	P225 00	P225	NONE

Datafilling Table LENLINES

The following table shows the datafill specific to line–to–line translations for table LENLINES. Only those fields that apply directly to line–to–line translations are shown:

Datafilling Table LENLINES

	Entry	Explanation and Action
LEN	 See subfields	Line Equipment Number (LEN) This field defines the physical location of the equipment that is connected to a specific telephone line.
		Note: Field LEN consists of subfields SITE, FRAME, UNIT, DRAWER or LSG SHELF, SLOT and CIRCUIT.
PTY	R1 to R5, T1 to T5 or S	If the line is assigned to a two-, four-, eight-, or ten-party line, enter the party "R1" to "R5" or "T1" to "T5", of the DN assigned to the line. If the line is assigned to an individual line, enter "S" for single party.
LNATTID	0 to 4,095	Line Attribute Index Enter the index into table LINEATTR.
-End-	 	

Datafill Example for Table LENLINES

The following example MAP display shows sample datafill for table LENLINES:

LEN	PTY	RINGCODE	DN
	SIGTYPE	LNATTIDX	OPTLIST
RCU0 00 0 03	l6 S DT	0	2252136 \$

Datafilling Table TOFCNAME

The following table shows the datafill specific to line—to—line translations for table TOFCNAME. Only those fields that apply directly to line—to—line translations are shown:

Datafilling Table TOFCNAME

Field	Subfield or Refinement	Entry	Explanation and Action
AREACODE		0 to 9,999,999	Area Code The area code identifies a major geographical area served by the switch. This field can contain one to seven digits. In a North American office, the area code must be three digits in length. Enter an area code that has been defined in table SNPANAME or table HNPACONT.
OFCCODE		0 to 9,999,999 or \$	Office Code The office code is a subregion of the area code. It can have zero to seven digits. In a North American office, the office code must be three digits in length.
 -End-			

Datafill Example for Table TOFCNAME

The following example MAP display shows sample datafill for table TOFCNAME:

AREACODE	OFCCODE
418	225

Datafilling Table DNINV

Table DNINV contains the data for all assigned and unassigned DNs. Table DNINV is a read–only table. Information is added to it as DNs are assigned or used in other tables such as LENLINES.

The following table shows the datafill specific to line—to—line translations for table DNINV. Only those fields that apply directly to line—to—line translations are shown:

Datafilling Table DNINV

Field	Subfield or Refinement	Entry	Explanation and Action
AREACODE		0 to 9,999,999 (1 to 7 digits)	Area Code The area code identifies a major geographical area served by the switch. If office parameter ACTIVE_DN_SYSTEM in table OFCENG is set to North American, the area code must be three digits long.

-continued-

Datafilling Table DNINV (continued)

Field	Subfield or Refinement	Entry	Explanation and Action
OFCCODE		0 to 9,999,999 (1 to 7 digits)	Office Code Digit Register The office code is a subregion of the area code. If office parameter ACTIVE_DN_SYSTEM in table OFCENG is set to North American, the area code must be three digits long.
			The office code must be specified in table TOFCNAME.
STNCODE		0 to 99,999,999 (1 to 8 digits)	The station code identifies a unique station within the Terminating Office (TOFC). If office parameter ACTIVE_DN_SYSTEM in table OFCENG is set to North American, the area code must be one or four digits in length. If one digit is entered, it is treated as a D-digit, where the D-digit represents the fourth digit in the format ABC-DEFG.
 -End-			

Datafill Example for Table DNINV

The following example MAP display shows sample datafill for table DNINV:

AREACODE	OFCCODE	STNCODE	DNRESULT
418	225	2164	H 11 0

Datafilling Table HUNTGRP

Table HUNTGRP is datafilled if the called line belongs to a hunt group.

The following table shows the datafill specific to line–to–line translations for table HUNTGRP. Only those fields that apply directly to line–to–line translations are shown:

Datafilling Table HUNTGRP

Field	Subfield or Refinement	Entry	Explanation and Action
HTGRP		0 to 8,191	Hunt Group Number Enter the hunt group number assigned to the hunt group.
DN		Numeric (up to 15 digits)	Directory Number Enter the listed DN of the hunt group.

⁻continued-

Datafilling Table HUNTGRP

Field	Subfield or Refinement	Entry	Explanation and Action
GRPTYP		DNH	Hunt Group Type Enter DNH and datafill refinement TRMOPT.
GRPDATA	TRMBOPT	Y or N	Terminating Billing Option If the optional terminator software package is provided, and if a record is generated for each call to a member of the hunt group, enter "Y". Otherwise, enter "N".
-End-			

Datafill Example for Table HUNTGRP

The following example MAP display shows sample datafill for table HUNTGRP:

HTGRP SNPA GRPDATA				D	N						GRPTYP
11	11 418		2252164							DNH	
N	N	N	RCVD	N	N	N	N	N	N	4	\$

Datafilling Table HUNTMEM

Table HUNTMEM is datafilled if the called line belongs to a hunt group.

The following table shows the datafill specific to line–to–line translations for table HUNTMEM. Only those fields that apply directly to line–to–line translations are shown:

Datafilling Table HUNTMEM

	Subfield or Refinement	Entry	Explanation and Action
HTGRP		0 to 8,191	Hunt Group Number Key Enter the hunt group number to which the member belongs.
HTMDATA		See subfield	Hunt Member Data This field consists of subfield SEL.
	SEL	B, D, L or P	Select Hunt Group Type Enter one of the following:
			<pre>B = Bridged Night Number D = Directory Number L = Multiline or Distributed Line P = Multiple Position</pre>
			If the entry in this field is "D", complete subfields DN and BNNDAT.

-continued-

Datafilling Table HUNTMEM (continued)

Field	Subfield or Refinement	Entry	Explanation and Action
	DN	Numeric (up to 15 digits)	Directory Number Enter the DN assigned to the hunt group sequence number.
-End-			

Datafill Example for Table HUNTMEM

The following example MAP display shows sample datafill for table HUNTMEM:

HTGRP	SEQNO	INSERT	HTMD	ATA	
11	0	N	D	2252164	N

Translation Verification Tools

The following example shows the output from the TRAVER command when it is used to verify line-to-line translations between DNs (406) 335-8691 to (406) 335-8692:

```
>TRAVER 1 3350691 3358692 b
TABLE LINEATTR
10 1FR NONE NT NSCR 7 406 P329 P329 NONE 0 NIL NILSFC NILLATA 0 NIL NIL 00 N $
LCABILL OFF - BILLING DONE ON BASIS OF CALLTYPE
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
TABLE LENFEAT
TUPLE NOT FOUND
TABLE OFCVAR
AIN_OFFICE_TRIGGRP OFCTRIG
AIN Orig Attempt TDP: no subscribed trigger.
TABLE STDPRTCT
P329 (1) (65021) 0
 . SUBTABLE STDPRT
WARNING: CHANGES IN TABLE STDPRT MAY ALTER OFFICE BILLING.
CALL TYPE DEFAULT IS NP. PLEASE REFER TO DOCUMENTATION.
 . 335 336 N NP 0 NA
 . SUBTABLE AMAPRT
 . KEY NOT FOUND
 . DEFAULT VALUE IS: NONE OVRNONE N
TABLE HNPACONT
406 128 0 ( 69) ( 1) ( 2) ( 0) 0
. SUBTABLE HNPACODE
 . 335 335 DN 406 335
AIN Info Collected TDP: no subscribed trigger.
TABLE TRIGGRP
OFCTRIG INFOANAL
 . N11 ( DG N11DIG) $ NIL
 . PODP ( DG PODPDIG) $ NIL
AIN Network Busy TDP: trigger criteria met, querying the database if network is busy.
TABLE TOFCNAME
406 335
TABLE DNINV
```

```
406 335 8692 L HOST 00 0 01 01
AIN Term Attempt TDP: no subscribed trigger.
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
TABLE LCASCRCN
406 P329 (3) MAND N
. SUBTABLE LCASCR
. 335 336
TABLE PFXTREAT
MAND NP Y NP UNDT
+++ TRAVER: SUCCESSFUL CALL TRACE +++
DIGIT TRANSLATION ROUTES
1 LINE
                      4063358692
                                  ST
TREATMENT ROUTES. TREATMENT IS: GNCT
1 *OFLO
2 LKOUT
+++ TRAVER: SUCCESSFUL CALL TRACE +++
-End-
```

The following example shows the output from the TRAVER command when it is used to verify line-to-line translations between DNs (418) 225-2136 and (418) 225-2164, where the called number belongs to a hunt group.

```
>TRAVER 1 2252136 2252164 b
TABLE LINEATTR
1 1FR NONE NT NSCR 0 418 P225 P225 NONE 0 NIL NILSFC LATA1 0 NIL NIL 00 Y AINRESGRP 0 0 $
LCABILL OFF - BILLING DONE ON BASIS OF CALLTYPE
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
TABLE LENFEAT
TUPLE NOT FOUND
TABLE OFCVAR
AIN_OFFICE_TRIGGRP OFCTRIG
AIN Orig Attempt TDP: no subscribed trigger.
TABLE STDPRTCT
P225 (1) (65021) 0
 . SUBTABLE STDPRT
WARNING: CHANGES IN TABLE STDPRT MAY ALTER OFFICE BILLING.
CALL TYPE DEFAULT IS NP. PLEASE REFER TO DOCUMENTATION.
. 225 225 L NP 0 418 225
AIN Info Collected TDP: no subscribed trigger.
TABLE TRIGGRP
OFCTRIG INFOANAL
 . N11 ( DG N11DIG) $ NIL
 . PODP ( DG PODPDIG) $ NIL
AIN Info Analyzed TDP: trigger criteria not met.
TABLE TOFCNAME
418 225
TABLE DNINV
418 225 2164 H 11 0
AIN Term Attempt TDP: no subscribed trigger.
TABLE DNATTRS
```

TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
TABLE HUNTGRP

11 418 2252164 DNH N N N RCVD N N N N N N 4 \$

TABLE HUNTMEM

11 0 N D 2252164 N

- . SUBTABLE AMAPRT
- . KEY NOT FOUND
- . DEFAULT VALUE IS: NONE OVRNONE N

LATA IS NIL, THEREFORE NOT AN EQUAL ACCESS CALL

+++ TRAVER: SUCCESSFUL CALL TRACE +++

AIN Network Busy TDP: trigger criteria met, querying the database if network is busy.

DIGIT TRANSLATION ROUTES

1 LINE 4182252164 ST

TREATMENT ROUTES. TREATMENT IS: GNCT

1 *OFLO

4 LKOUT

+++ TRAVER: SUCCESSFUL CALL TRACE +++

-End-

Nortel DMS-100 Introduction to Query Commands

Introduction

Query commands consist of a command name followed by a series of parameters. Operating companies use query commands to display the characteristics of telephone lines. Query commands allow users with access to a DMS I/O Device (IOD) to determine the status (working or unassigned) of Directory Numbers (DN) or Line Equipment Numbers (LEN) associated with lines.

The information received through query commands simplifies service order preparation. Entering the query command QDN (Query Directory Number) and a DN gives a user information about the hardware and software associated with the DN of a particular line.

Query Commands

The Line Data Base (LDB) query commands are used to:

- Determine the status (working or unassigned) of a DN.
- Determine the status (working or unassigned) of a LEN.
- Identify the parameters associated with a working line.

The commands can be executed at any level of the Man–Machine Interface (MMI) system. A command is not needed to enter or leave the query mode. A user logged on at a MMI position can enter a query command.

Either the *prompt* or *no-prompt* mode of entry can be used. Entering a \$ (dollar sign) character indicates that the user is either finished entering data for a parameter or has accepted the default parameter. The user can confirm, reject, or edit the input the same way as for Service Order (SERVORD) commands.

Note: The format for the query commands output, with the exception of QDN, has been altered when the summary option is used. The drawer totals for each peripheral are written on two lines when the peripheral has more than 17 drawers. Two lines are required if the total number of drawers chosen is greater than 17. No change is made to the type of information that is displayed.

Entering Query Commands in No-Prompt Mode

In *no-prompt mode*, query commands are entered by the user along with the required correct parameters. In case of an error, the DMS switch reverts to the prompt mode of entry, beginning at the point where the invalid parameter was entered.

Entering Query Commands in Prompt Mode

To enter query commands in *prompt mode*, perform the following actions:

- 1. Log on.
- 2. Enter one of the commands shown in the following "Query Commands" table.

- 3. Refer to the "Query Command Prompts" table in this article for an explanation of the query command prompts and the data to be entered. If an incorrect parameter is entered, the system prompts for the correct information.
- 4. On entry of a valid parameter, the DMS switch displays the next prompt. The DMS switch continues to prompt until all necessary parameters have been entered.

When all parameters have been entered, the DMS switch displays or prints the order as entered. The user must enter a Y (to accept the command), a \mathbb{N} (reject), or a \mathbb{E} (edit).

Query Commands

-End-

Command	Description
MQDN	Query Mobile DN (DMS-100 Wireless Switch Only)
QBERT	Query Bit Error Rate Testers
QCM	Query Call Memory
QCOUNTS	Query Counts
QCPUGNO	Query the CPU Group Number
QCUST	Query Customer Information
QDN	Query Individual Line Data
QDNA	Query Data Network Address
QDNSU	Obtain a Summary of Unassigned DNs
QDNWRK	Obtain a Summary of Assigned DNs
QGRP	Query a Call Pickup or Long Speed Call User Group
QHA	Obtain a Detailed Listing of Assigned Hardware
QHASU	Obtain a Summary of LEN Hardware Assigned and Software Unassigned
QHLR	Query Home Location Register (DMS-100 Wireless Switch Only)
QHU	Obtain a Summary of LEN Hardware Unassigned
QLEN	Query Line Data Related to a Given LEN
QLENWRK	Obtain a Summary of Working (Hardware Assigned and Software Assigned) LENs
QLOAD	Obtain a Summary of LEN Assignments by Line Class Code
QLRN	Query Location Routing Number
QMADN	Query Multiple Access Directory Mumber
QMODEL	Query Model
QMSB	Query Make Set Busy
QNCOS	Query Network Class of Service
QPHF	Query Packet Handler
QPRIO	Query Guaranteed Background Class of Process
QSCUGNO	Query Speed Call User Group Number
QSIMR	Query Simultaneous Ringing (SimRing) Group
QSL	Query Screening List
QTOPSPOS	Query TOPS Positions
QWUCR	Query Wake-Up Call Requests

Query Commands Line Class and Agent Class Code Compatibility

The following table lists query commands and each type of line class or agent class with which they are compatible. The line types are General Services Framework Enhanced Residential (GSF), International Business Networks (IBN), and Meridian Digital Centrex (MDC), which functions within IBN.

Query Commands Line Class and Agent Class Compatibility

Command	GSF	IBN	MDC	
MQDN	N	Y	У	
QBERT	Y	Y	Y	
QCM	N	Y	Y	
QCOUNTS	N	Y	N	
QCPUGNO	N	Y	Y	
QCUST	N	Y	Y	
QDN	Y	Y	Y	
QDNA	N	Y	N	
QDNSU	N	Y	Y	
QDNWRK	Y	Y	Y	
QGRP	N	Y	Y	
QHA	N	Y	Y	
QHASU	N	Y	Y	
QHLR	N	Y	Y	
QHU	N	Y	Y	
QLEN	Y	Y	Y	
QLENWRK	Y	Y	Y	
QLOAD	N	Y	Y	
QLRN	N	Y	Y	
QMADN	N	Y	Y	
QMODEL	N	N	N	
QMSB	N	Y	Y	
QNCOS	N	Y	Y	
QPHF	N	N	N	
QPRIO	N	Y	Y	
QSCUGNO	N	Y	Y	
QSIMR	N	Y	Y	
QSL	N	Y	Y	
QTOPSPOS	N	N	N	
QWUCR	N	Y	Y	

-End-

Query Command Prompts

The following table lists the query command prompts and the correct data to be entered for each prompt.

Query Command Prompts

Prompt	Valid input	Explanation
CARD_CODE	6X17AA, 6X18AA, 6X18AB, 6X21AA, 6X58AA, or NIL_CTN	The type of line card to be queried. Defaults to all card types.
CUSTGRP	Alphanumeric	The customer group is a group of lines identified by a common language name.
CUSTNAME	1 to 16 characters	The customer name is the character name assigned to the customer group.
DIRECTORY_ NUMBER	Digits	The DN to be queried.

Query Command Prompts (continued)

Prompt	Valid input	Explanation
DIRECTORY_ NUMBER_	R, ALL or R nnnnnnn nnnnnnn	
RANGE		* R prompts you to set a range.
		* ALL queries every DN.
		* R and the two series of digits represent the starting and final DNs of the set to be queried.
DN_OR_LEN	DN = 7 or 10 digits entered with no spaces or hyphens.	Directory Number (DN) associated with the service that is to be established, modified, or deleted. 7-digit DN ambiguity exists if the DMS-100 switch serves more
		than one NPA and the same 7-digit DN is used in multiple NPAs.
		To resolve this, you will be prompted to enter the full 10-digit DN.
	<pre>LEN = An entry in the format:</pre>	The Line Equipment Number (LEN) associated with a service to be established, modified, or deleted.
	<site> ff u dd cc</site>	
	Where:	
	<pre>* <site> = Site Name Defaults to HOST (4 alphanumeric characters</site></pre>	3)
	* ff = Frame Number (0 to 511)	
	* u = Unit (0 to 19)	
	* dd = Drawer Number Drawer number of the line spread group. (0 to 19)	
	* cc = Line Circuit Number (0 to 31)	

Query Command Prompts (continued)

Prompt	Valid input	Explanation
FORMATTED_OR _HEX_(F H)	H or F	Applies to QCM command only. The hexadecimal (H) option provides the same information as the formatted (F) option, and includes:
		* Display of what is now in system memory (the "physical view").
		* The information that the DMS switch needs for table control (the "logical view").
FORMAT	S (summary) or D (detailed)	Specifies if the printout is to be a summary or a detailed view of the information.
FROM_DN	Digits	The first DN in a range (R) of DNs queried.
	Digits	The first LM in a range (R) of LMs queried.
GND	Y or N	Applies to QHA and QHASU commands only. Specifies if ground start only or both loop and ground start lines are to be queried.
		If "Y" is entered, the data on ground start lines is printed. If "N" is entered, the data on loop and ground start lines (N) is printed.
GRP_TYPE	CPU or SCU	Applies to QGRP command only. Specifies the type of group to be queried.
LEN	Refer to DN_OR_LEN above.	The LEN associated with a service to be established, modified, or deleted.
LINE_CLASS_	Refer to the "Line Class Code Table" in Issue #11.	The Line Class Code (LCC) of the service to be queried.
		Defaults to all line class codes (NLCC).
LINE_DRAWER_ RANGE	Valid input format: R n1 n2 n3 \$ALL	The line drawers in each line module to be queried.
		Where:
		n1, n2,, nn, are the designated numbers (0 to 19) of the line drawers to be queried and $\$$ is the list delimiter.
		Defaults to all drawers (ALL).

Query Command Prompts (continued)

Prompt	Valid input	Explanation
LINE_MODULE_ RANGE	Valid input format: lm_ident fn1 un1 lm_ident fn2 un2	A range of Line Modules (LM) or Line Concentrating Modules (LCM) to be queried.
	IM_Ident Inz unz	Where:
		<pre>lm_ident = is the Site Name</pre>
		fn1 = First Frame number (0 to 99)
		un1 = First Unit Number (0 to 9)
		<pre>fn2 = End Frame Number (0 to 99)</pre>
		un2 = End Unit Number (0 to 9)
		Defaults to all LM or LCM.
NUMBERING_ PLAN_AREA	3 digits	The Numbering Plan Area (NPA) of the DN to be queried.
		DMS-100 Wireless QHLR and MQDN commands only
	3 digits	The central office code of the DN to be queried.
		DMS-100 Wireless QHLR command only.
OPTION		Applicable to QDNWRK and QLENWRK commands only.
		Refer to the "Line Service Options" table in Issue #11 for a list of valid inputs.
		If one option is entered, only data on lines with the specified option is printed out.
		If a "\$" character is entered, the printout includes all options. When the option is entered in no-prompt mode, the option must be delimited by the "\$" character.
RANGE	R (range) or N (no)	Allows a group of DNs or LENs to be queried.
	4 digits	The station of the DN to be queried.
		DMS-100 Wireless QHLR command only.
-continued-		

Query Command Prompts (continued)

r D	The type of printout required. "S" specifies a summary printout. Produces a count of the DNs or LENs being queried. "D" specifies a detailed printout. Provide
	the same information as "S", but includes other information that varies according to the query command selected.
	Defaults to Summary (S).
its	The last DN in a range (R) being queried.
its	The last LM in a range (R) being queried.
	The type of treatment to be queried. ANCT = Machine Intercept Announcement BLDN = Blank DN OPRT = Operator Intercept TRBL = Trouble Intercept UNDT = All Treatments Defaults to UNDT. Treatments are defined in the data schema section of the Translations Guide.
i - i -	its tts

Examle Output of the QDN Command

Example output message of the Query Directory Number (QDN) command in prompt mode with an unique 7–digit directory number.

>QDN 6210100

```
DN: 6210100
TYPE: SINGLE PARTY LINE
SNPA: 416 SIG: DT LNATTIDX: 0
LINE EQUIPMENT NUMBER: HOST 02 0 01 00
LINE CLASS CODE: 1FR
IBN TYPE: STATION
CUSTGRP: RESG000 SUBGRP: 0 NCOS: 0
CARDCODE: 6X17AC GND: H PADGRP: STDLN BNV: NL MNO: N
PM NODE NUMBER: 26
PM TERMINAL NUMBER: 33
OPTIONS:
CWT DGT
RES OPTIONS:
SCF NOAMA INACT SCRJ NOAMA ACT $
```

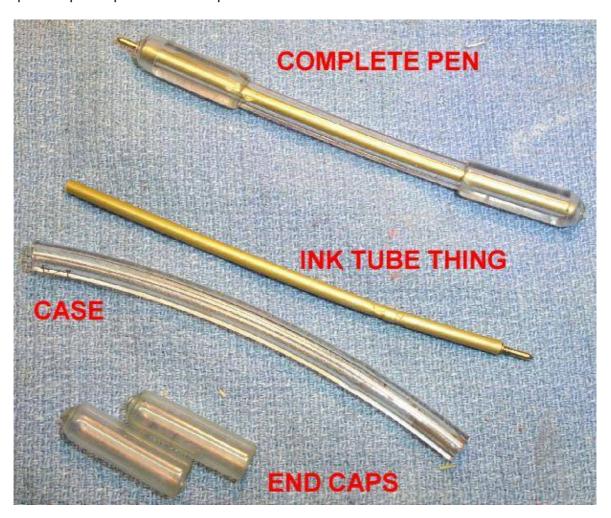
Prison Pen Weapon

Overview

This is a little trick to turn a simple prison "safety" pen into a one-time use stabbing weapon or "shank." This is a useful tool if you happen to be sharing a prison cell with any script-kiddie / sell-out computer hackers, and desperately want to put them out of their misery.

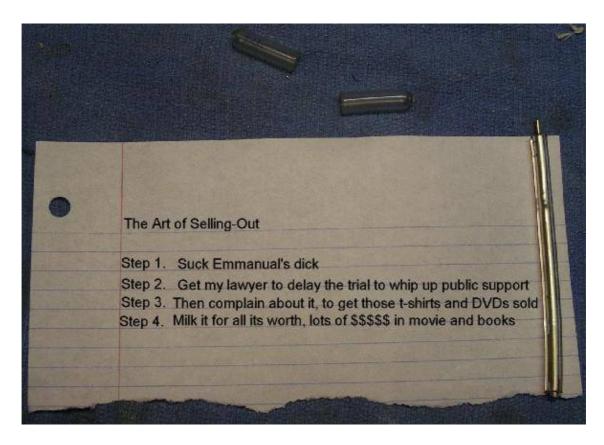
Remember: Real computer security consultants don't spend five years in jail.

Example of a prison pen and its components:

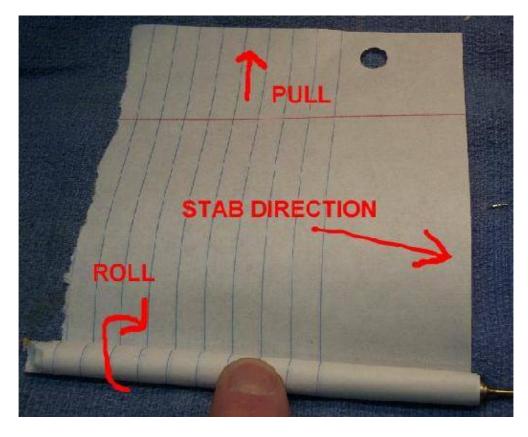


These style of prison security pens are just the ink tubes from regular "click" pens, but with the outside case replaced with a piece of 1/4-inch flexible rubber tubing. Two 1/4-inch end caps (one with a hole in it for the tip to poke out) complete the pen. The idea is that the soft tubing will reduce the chance of the pen being used as a stabbing weapon.

Now, to turn the pen into a stabbing weapon, all you have to do is somehow strengthen the outside case of the pen. The easiest way to do this, with the limited materials available, is to tear a sheet of paper in half, and *very tightly* wrap the flexible tube in it. Wrapping the pen case in paper will significantly increases its strength.



Example of how to wrap the pen case with a torn sheet of paper. You should remove the pen's end caps, and place the pen along the smooth edge of the paper.



Example of rolling. You'll have to pull the paper *very tightly* for this to work. Practice if you can. The end with the tip sticking out should be used as the stabbing tip.



Die Motherfucker Die!

It works pretty well, but it's only a one-time shot as the paper will unravel.

Linux FRS Radio Control

Overview

This is a simple hardware computer interface and Linux program which can be used to control the most commonly used buttons on any consumer Family Radio Service (FRS) or General Mobile Radio Service (GMRS) radio.

There are 22 available FRS/GMRS frequencies in the 462–467 MHz band. Each channel is also capable of 38 "private" subchannels. The subchannels are the result of using different Continuous Tone Coded Squelch System (CTCSS) tones, also referred to as "PL" tones, for each channel. By remotely controlling the **Mode**, **Up Arrow**, and **Push–to–Talk** (PTT) buttons, you can control just about any feature or function of the radio. This may be useful for anyone trying to build a store–and–forward "cellular" network using cheap FRS/GMRS radios (hint, hint). It's possible to control other (or more) buttons with trivial hardware and software modifications.

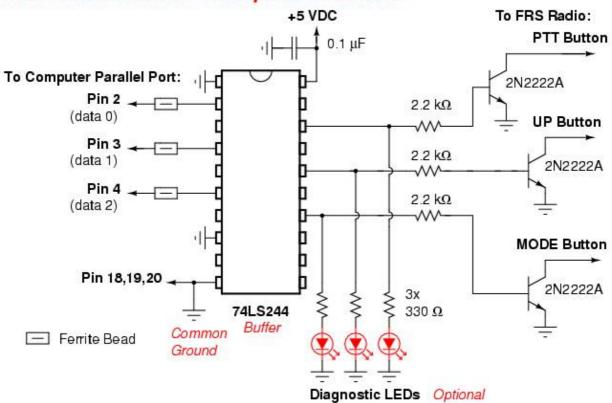
This particular example is built using a Cobra PR950DX FRS/GMRS radio (FCC ID: BBOPR950DXD) and a 74LS244 buffer/interface board. The radio's buttons are active—low (ground to enable) logic, so the hardware interface is quite simple. The computer software was written to run under the Linux operating system. It's not the best software in the world, but that's because I'm a fucking retard. I have no idea how to do this under Windows — or why you'd want to.

Other model radios should be quite similar. Check the FCC ID database to see if a schematic is available for the radio.

Computer Interface

The computer–to–radio interface hardware consists of parts (except the 74LS244) which are available at your local Radio Shack. The 74LS244 buffer IC is actually optional, as most computer parallel ports are limited to the amount of current they can source (output). This buffer will prevent any damage to your computer hardware should something short out, or if you need to control any additional hardware. The diagnostic LEDs are also optional, but are helpful to verify what the computer is outputting, data wise, at a quick glance.

FRS Radio Control - Computer Interface

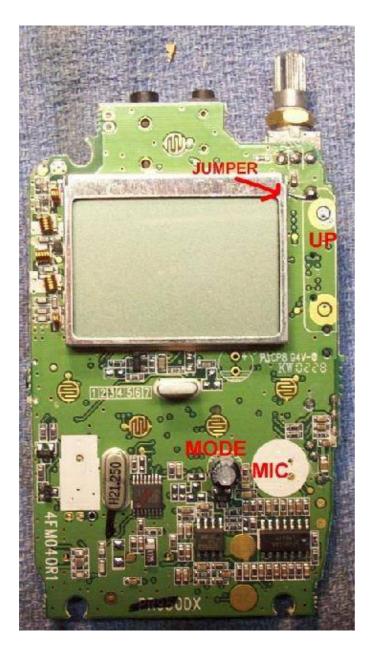


Pictures



Stock Cobra PR950DX overview.

Buttons are: Talk / PTT, Monitor / Light (left side), Call, Hi / Lo / Lock, Mode, Enter (under LCD screen), Up & Down (right side).

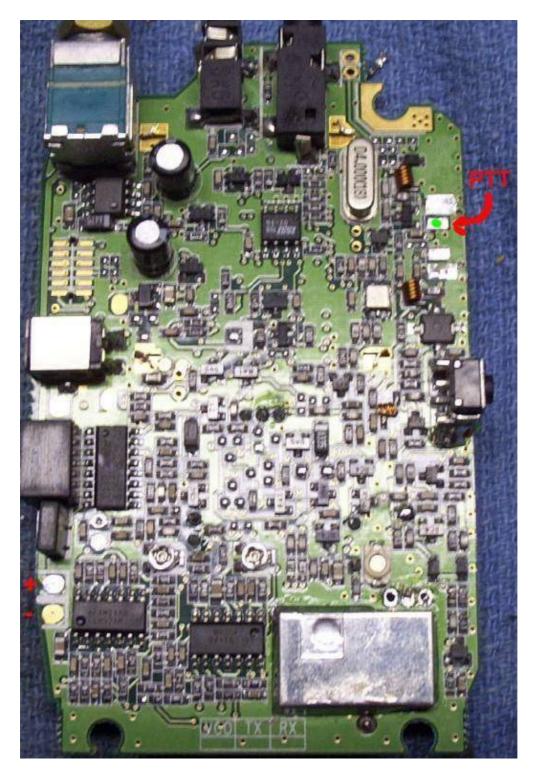


Cobra PR950DX's internal front view. The motor/vibrator and electret microphone have been removed. Inject your own audio (say from another radio) into the microphone jack on the top of the radio. The **On/Off** switch on the volume potentiometer is locked in the **On** position by soldering a jumper wire across its terminals.



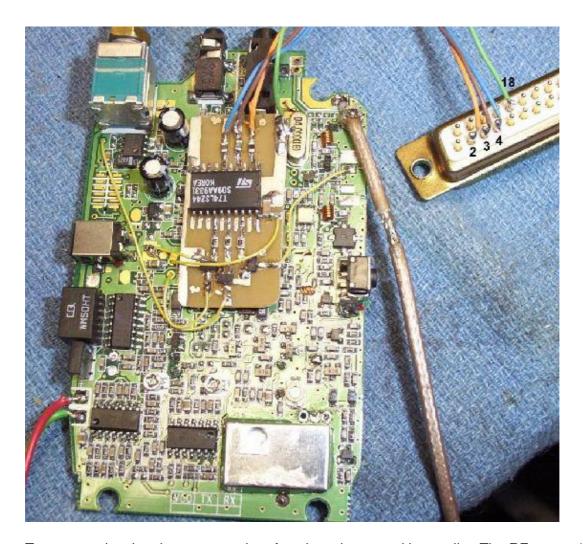
Alternate view. The **RED** dot shows where to solder the control wire from the computer interface board to control the **Mode** button.

The **BLUE** dot shows where to solder the control wire from the computer interface board to control the **Up Arrow** button.

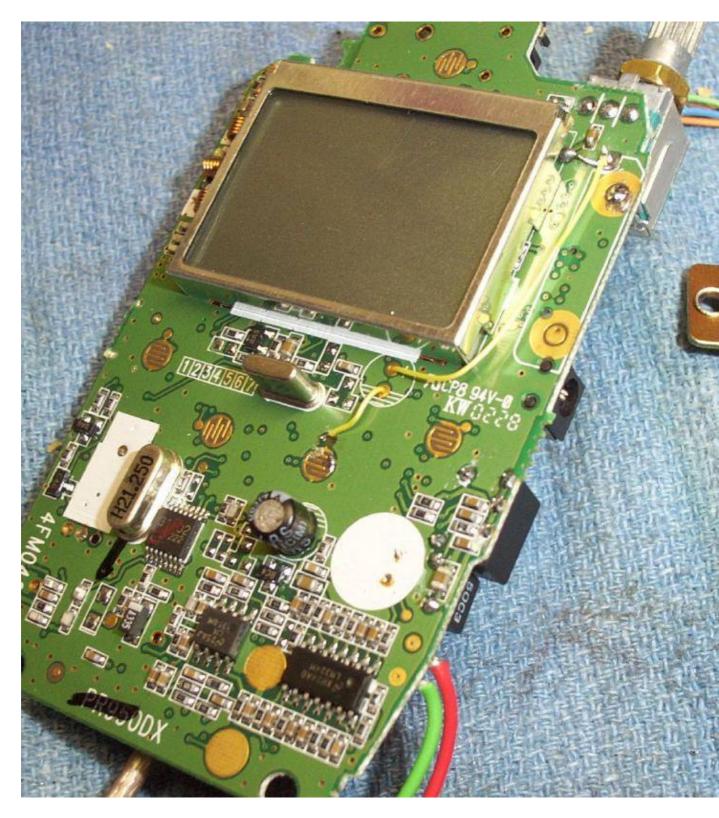


Internal rear view. The **PTT** button has been removed. The solder pad with the arrow is the new **PTT** switch. The + and - on the lower left is for the +6 VDC power supply input.

The **GREEN** dot shows where to solder the control wire from the computer interface board to control the **PTT** button.



Test setup showing the computer interface board mounted internally. The RF output (via a short BNC jumper) is taken from the stock antenna connection. The **BLACK** numbers are the parallel port's DB–25 pins. There should be a common ground between the radio and the computer. There are (optional) ferrite beads on the computer lines going into the 74LS244 buffer.



Front panel close-up.

Software

FRS/GMRS radio control software for Linux. Needs to run as root (userid = 0) for hardware port access.

Edit frs_control.c for your parallel port address. Default is 0x378, which is LPT1 on most computers.

```
----- CUT & SAVE AS frs_control.c -----
/* Linux FRS Radio Control
/* GBPPR 'Zine - Issue #13 - April 2005 - http://zine.gbppr.org */
                                                                      * /
                                                                      */
/* Compile using: gcc -02 frs_control.c -o frs_control
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <asm/io.h>
/* 0x278 is 2nd parallel port */
int c;
extern int errno;
                             /* pin 432 on the parallel port */
unsigned char data_0 = 0; /* Send 000000000 */
unsigned char data_1 = 1; /* Send 00000001b */
unsigned char data_2 = 2; /* Send 00000010b */
unsigned char data_4 /* /* Send 00000010b */
unsigned char data_4 = 4; /* Send 00000100b */
void usage (char *);
int txtime, verbose;
int.
main (int argc, char **argv)
  if (geteuid () && getuid ())
      fprintf (stderr, "\n\n%s must be run as root to access hardware ports.\n", argv[0]);
      fprintf (stderr, "Leaving...\n\n");
      exit (1);
  if (argc < 2)
     usage (argv[0]);
  while ((c = getopt (argc, argv, "t:rum")) != EOF)
      switch (c)
          case 't':
            txtime = atoi(optarg);
            TX ();
            exit (0);
          case 'r':
            RESET ();
            exit (0);
          case 'u':
```

```
UP ();
           exit (0);
          case 'm':
           MODE ();
           exit ();
          default:
           usage (argv[0]);
    }
 return (0);
int
TX (void)
 porttest ();
 outb (data_1, BASE_ADDRESS);
 sleep (txtime);
 outb (data_0, BASE_ADDRESS);
 usleep (200000);
 ioperm (BASE_ADDRESS, 3, 0);
 return (0);
}
int
RESET (void)
 porttest ();
 outb (data_0, BASE_ADDRESS);
 printf ("Parallel Port Reset: 0x%x hex\t(%d decimal)\n", BASE_ADDRESS, BASE_ADDRESS);
 printf ("Sending '00000000' ...\n");
 usleep (200000);
 ioperm (BASE_ADDRESS, 3, 0);
 return (0);
}
int
UP (void)
{
 porttest ();
 outb (data_2, BASE_ADDRESS);
 usleep (500000);
 outb (data_0, BASE_ADDRESS);
 usleep (200000);
 ioperm (BASE_ADDRESS, 3, 0);
 return (0);
}
int
MODE (void)
 porttest ();
```

```
outb (data_4, BASE_ADDRESS);
 usleep (500000);
 outb (data_0, BASE_ADDRESS);
 usleep (200000);
 ioperm (BASE_ADDRESS, 3, 0);
 return (0);
}
int
porttest (void)
  if (ioperm (BASE_ADDRESS, 3, 1) < 0)
     fprintf (stderr, "\nERROR: Can not open port 0x%x\n\n", BASE_ADDRESS);
     perror ("port");
     exit (errno);
 usleep (200000);
 return (0);
}
void
usage (char *pname)
 fprintf (stderr, "\n\nUsage: %s\n\n", pname);
 fprintf (stderr, "\t-t x\tEnable Transmit for 'x' seconds\n");
 fprintf (stderr, "\t-r\tParallel port reset\n");
 fprintf (stderr, "\t-u\tEnable Up Arrow\n");
 fprintf (stderr, "\t-m\tEnable Mode Function\n");
 exit (1);
----- CUT -----
```

The PR950DX (or whatever radio you use) should be in a "clean" state when the software first starts. That is, all subchannels should be at "00" and the starting channel should be 1. Otherwise, you'll get all confused. Have fun!

Use frs_control -t x to enable the PTT button (transmit) for "x" seconds.

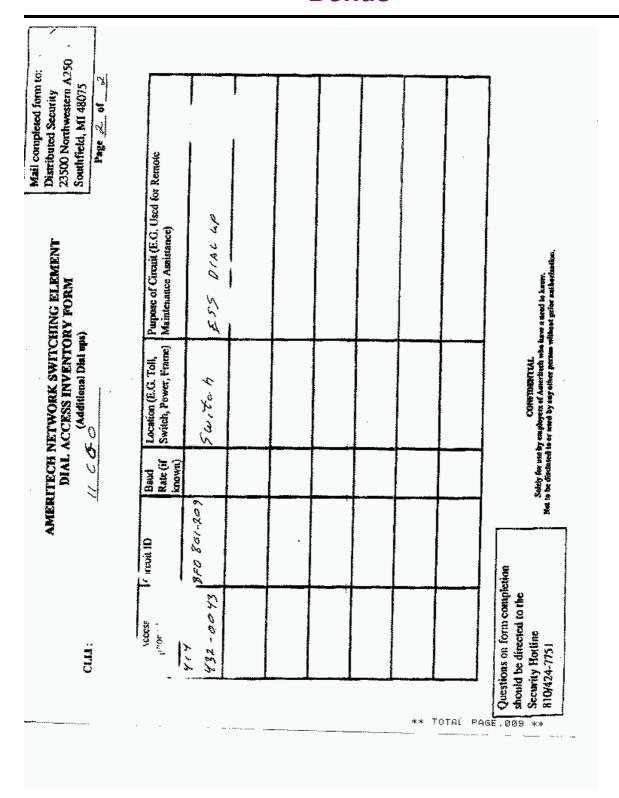
Use $frs_control -r$ to reset the parallel port to "all zeros" if the program crashes or doesn't exit right.

Use frs_control -u to enable the Up Arrow button.

Use frs control -m to enable the Mode button.

Call the frs_control program from a Perl or shell script to interface it with audio recording/playing software or other programs.

Bonus



End of Issue #13



Any Questions?

Editorial and Rants

From http://www.insightmag.com/main.cfm?include=detail&storyid=357431

Eurobiz Is Caught Arming Saddam

Posted Feb. 4, 2003

By Kenneth R. Timmerman

Disability payments are going to 161,000 U.S. veterans for what appears to have been low–level exposure to Iraqi chemical weapons during the Persian Gulf War a dozen years ago. Documents dated 1997–98 and obtained for the first time by Insight from Gary Pitts, a Houston lawyer representing the vets in a \$1 billion class–action lawsuit against the companies that supplied Iraq with precursors and production equipment in the 1980s, show that Saddam Hussein has retained much of his chemical– and biological–weapons capabilities and continues to do business with many of the same European suppliers.

Chief U.N. weapons inspector Hans Blix came to similar conclusions on Jan. 27, when he delivered an "update" to the U.N. Security Council based on his inspection efforts in Iraq. Berating Baghdad for failing to disclose and dismantle its forbidden weapons, Blix insisted that cooperation had to be "active" to meet U.N requirements. "It is not enough to open doors," he told the Security Council. "Inspection is not a game of 'catch—as—catch—can." The facts presented by Blix — who has been called "spineless" by former colleague Per Ahlmark in Sweden — belie his conclusion that the Security Council should give the inspectors more time. Security Council Resolution 1441, which was passed 15–0 in November 2002 and brought the inspectors back to Baghdad for one last try, summons Iraq to cooperate fully with U.N. Monitoring, Verification and Inspection Commission (UNMOVIC) inspectors and voluntarily to surrender and dismantle all remaining unconventional

weapons and production gear. But as Blix and Mohammad El Baradei, his replacement at the International Atomic Energy Agency, detailed to the Security Council, the Iraqi government continues to play a game of "hide-and-seek." "We're not talking about aspirin," Secretary of State Colin Powell told the World Economic Forum in Davos, Switzerland, the day before Blix made his presentation. "We're talking about the most deadly things one can imagine." Among the capabilities identified by Blix that Saddam retained are huge quantities of sarin nerve gas. The Iragis used sarin against Iranian troops during the 1980–88 Iran–Irag War with deadly effect — a single drop on the skin will kill a man. Persian Gulf War veterans believe that Saddam purposefully stockpiled munitions packed with sarin and other chemical weapons in conventional ammunition dumps in southern Iraq as he withdrew his forces from Kuwait, knowing that the United States would order the ammo dumps destroyed. "So you can say that Saddam Hussein indirectly attacked U.S. troops with chemical weapons during the 1991 war," Pitts tells Insight. An increasing number of U.S. government scientists are coming to share that view. But Saddam's 1997-98 "Full, Final and Complete Disclosure" shows that he also retains VX, a chemical-warfare agent said to be even more deadly than sarin. The Iraqis acknowledge having imported or produced, up to that date, 750 tons of chemicals needed for making VX. Except for the U.N.-controlled disposal of 170 tons at that time, the rest of these deadly chemicals are missing. Blix said his inspectors had found "indications that the [VX] agent was weaponized." Already in 1997, then-U.S. ambassador to the United Nations Bill Richardson concluded, "Iraq has perfected the technique for the large-scale production of a VX precursor that is well-suited to long-term storage." Blix's recent report to the United Nations documented many other instances in which Iraq has failed to disclose prohibited weapons. The Iraqi documents, revealed here for the first time, portray a worldwide procurement network that relied on top-drawer assistance from governments and major banks in Europe, Asia and the Middle East. Iraq's suppliers range from huge multinational corporations, such as Hoechst AG of Germany, to little-known entities, such as an outfit identified by the Iraqis as "Kim Al-Khalleej" of Singapore, which they claimed supplied more than 4,500 tons of VX-, sarin- and mustard-gas precursors and production equipment.

Until 1984, when the United States interdicted a small shipment of nerve-gas precursors from Al Haddad Trading in Nashville, there were no regulations banning commerce in chemical-weapons precursors or production equipment. But after that, at U.S. insistence, the major chemical-producing nations in the West agreed to require export licenses for a list of chemical-weapons (CW) precursors that gradually was expanded as evidence of Iraq's battlefield use against Iran and Iraqi Kurds became available. Once the Chemical Weapons Convention was negotiated in 1989 in Paris and ratified by all major producing countries, trade in banned chemicals was highly regulated. And yet, as Iraq was forced to admit in 1997-98 when U.N. inspectors found the evidence, huge quantities of those same chemicals and banned production equipment reached Iraq courtesy of complacent governments and greedy corporations, banks and middlemen beginning in the 1980s. According to the Iraqi 1997–98 declaration, one of the biggest suppliers was Exomet Plastics of Bombay, India, now part of EPC Industrie. An EPC attorney told CNN in late January that the only chemicals Exomet had shipped to Iraq were for pesticides. "There were no restrictions for exporting these chemicals at the time the exports were made." he added. But critics dismiss such corporate excuses as phony. Exomet shipped much more than mere pesticides, according to the Iraqi documents. It supplied materials usable only for CW, such as 1,000 tons of thionyl chloride, a chemical used to make mustard gas and sarin. It shipped 300 tons of phosphorus trichloride, 300 tons of dimethylamine HCl, 250 tons of phosphorus oxychloride and 250 tons of P2S5 — all for use in manufacturing sarin. It shipped another 192 tons of chlorethanol used for making VX between 1988 and 1990. At the same time these shipments were occurring, the Iraqi Ministry of Agriculture purchased huge quantities of completely manufactured pesticides for agricultural purposes from ICI, Ciba Gigy, Dow Chemical, Roussel and others. "Negotiations were carried out in 1989-1990 with a number of companies to get the know-how to produce certain pesticides," the Iraqi documents state. While Iraq frequently disguised its CW programs behind

pesticide production, few producers in Europe or elsewhere really were duped. This reporter's database, compiled from U.N. documents and his investigations during the last 15 years, includes 131 companies that provided CW assistance and another 42 that aided Irag's biological-weapons (BW) program. Iraq names 56 companies that provided chemicals and production equipment. One surprise in the Iraqi list was the appearance of Egypt's Abu Zaabal Special Chemicals Co., a state-owned conglomerate with artillery and ammunition plants run by the Egyptian Ministry for Military Production. While Abu Zaabal's involvement in Irag's programs was revealed during the Scott commission inquiry in Britain in 1993, the quantities of nerve-gas precursors delivered by Abu Zaabal, as declared by Iraq, is a mind-boggling 1,300 tons in all. The Iraqis also state that Abu Zaabal delivered 200 tons of hydrogen cyanide, a CW agent in its own right. Iraq used cyanide agents in 1988, four years after this Egyptian delivery, when it gassed thousands of Kurds in the town of Halabia. In 1989, when the New York Times accused Egypt of transforming Abu Zaabal into a CW production plant with the help of a Swiss company, President Hosni Mubarak hotly denied the charge. The Iraqi documents Insight has obtained show that Abu Zaabal already was producing chemical weapons and precursor chemicals at least five years earlier. Abu Zaabal is located in the Cairo suburb of Heliopolis and is known in Egypt as Factory 18, a carryover from the days of Col. Gamal Abdel Nasser, when all military plants were known by number and not by name. "Egypt was the first country after World War II to use chemical weapons, against rebels in Yemen, so we've known they've had a chemical-weapons capability for some time," says Shoshana Bryen of the Jewish Institute for National Security, a Washington think tank. "These undeclared sales to Iraq raise serious questions about Egypt's trustworthiness as a U.S. ally," she says. Despite the years of war and U.N. inspections, Karl Kolb GmbH, the German company that designed and built Irag's main CW production plants in the 1980s, never really left Baghdad. Prosecuted in Germany in 1984 for having delivered CW gear to Iraq, the company won its case, then turned around and successfully sued the German government for libel. In 1999, when the German government sent its first official trade mission to Iraq since the 1991 gulf war, Karl Kolb official Michael Fraenzel went along for the ride. That mission led to fresh business for the German chemical-equipment broker. In 1999 and 2000, it submitted five requests to the U.N. Sanctions Committee to sell close to \$2 million in chemical- and possibly biological-weapons-production gear. The equipment Karl Kolb wanted to sell included a \$271,000 "incubator," which was on a list of proscribed equipment because of potential weapons use. All five requests from Karl Kolb were put on hold by the U.S. government. Undeterred, Karl Kolb went back in 2001 as a prominent participant in the Baghdad International Fair, according to a list of participants posted on the fair's official Website (http://www.baghdad-international-fair.de/eng/ws04.html). They were not alone. Chemical companies and machine-tool makers abounded, as did major industrial players such as telecommunications giant Siemens AG and DaimlerChrysler AG, whose U.S.-based vice president for government affairs told a reporter at the Rev. Jesse Jackson's Wall Street Project conference last month that conservatives such as U.S. President George W. Bush "have a rather myopic view of the world."

Other major suppliers mentioned in the Iraqi documents include German chemical giant Preussag, equipment maker Schott Glasswerke, Dutch chemical suppliers Melchemie, KBS and Philips Export, Fluka Chemia AG of Switzerland and French partners De Dietrich and Protect SA. In all, the Iraqis named 14 German, three Dutch, three Swiss and two French companies as their top CW suppliers, although dozens more played supporting roles. According to U.N. databases Insight was able to access, since 1998 French companies lead the pack in applying for U.N. licenses to sell potential weapons material to Iraq, with more than 272 different license applications worth billions of dollars. The United States put 93 of those contracts, worth \$217 million, on hold. Among them was the sale as "medical equipment" of a series of lithotripsy machines for treating kidney stones without surgery manufactured by the company Karl Storz Endoscopie France SA. Perfectly normal? Think again. The lithotripter employs a high–speed krytron switch similar to those used to trigger nuclear warheads. Along with the six

medical machines, Iraq sought 120 spare krytrons, the U.N. Website reveals. After France, countries with the most applications were Jordan (184), Russia (154), the United Arab Emirates (112), Italy (97) and China (66). While German companies had made only 36 applications, the dollar value was believed to be in the billions, just behind France. "Whenever the French and the Germans criticize the U.S. for going to war against Saddam," one congressional insider tells Insight, "we should understand that it's really all about the euros."

Kenneth R. Timmerman is a senior writer for Insight magazine.





Make the World a Better Place - Kill All Europeans