

Technical Report
Research Project Agreement No. T9903, Task 70
Fuzzy Neural Seattle

**DOCUMENTATION OF TSMC SOFTWARE
THAT INTERFACES WITH
TRAFFIC ANALYSIS PROGRAMS**

by

Cynthia E. Taylor
Research Engineer

Deirdre R. Meldrum
Associate Professor

Department of Electrical Engineering
University of Washington
Seattle, Washington 98195

Washington State Transportation Center (TRAC)
University of Washington, Box 354802
University District Building
1107 NE 45th Street, Suite 535
Seattle, Washington 98105-4631

Washington State Department of Transportation
Technical Monitor
Les Jacobson
Traffic Systems Manager, Northwest Region

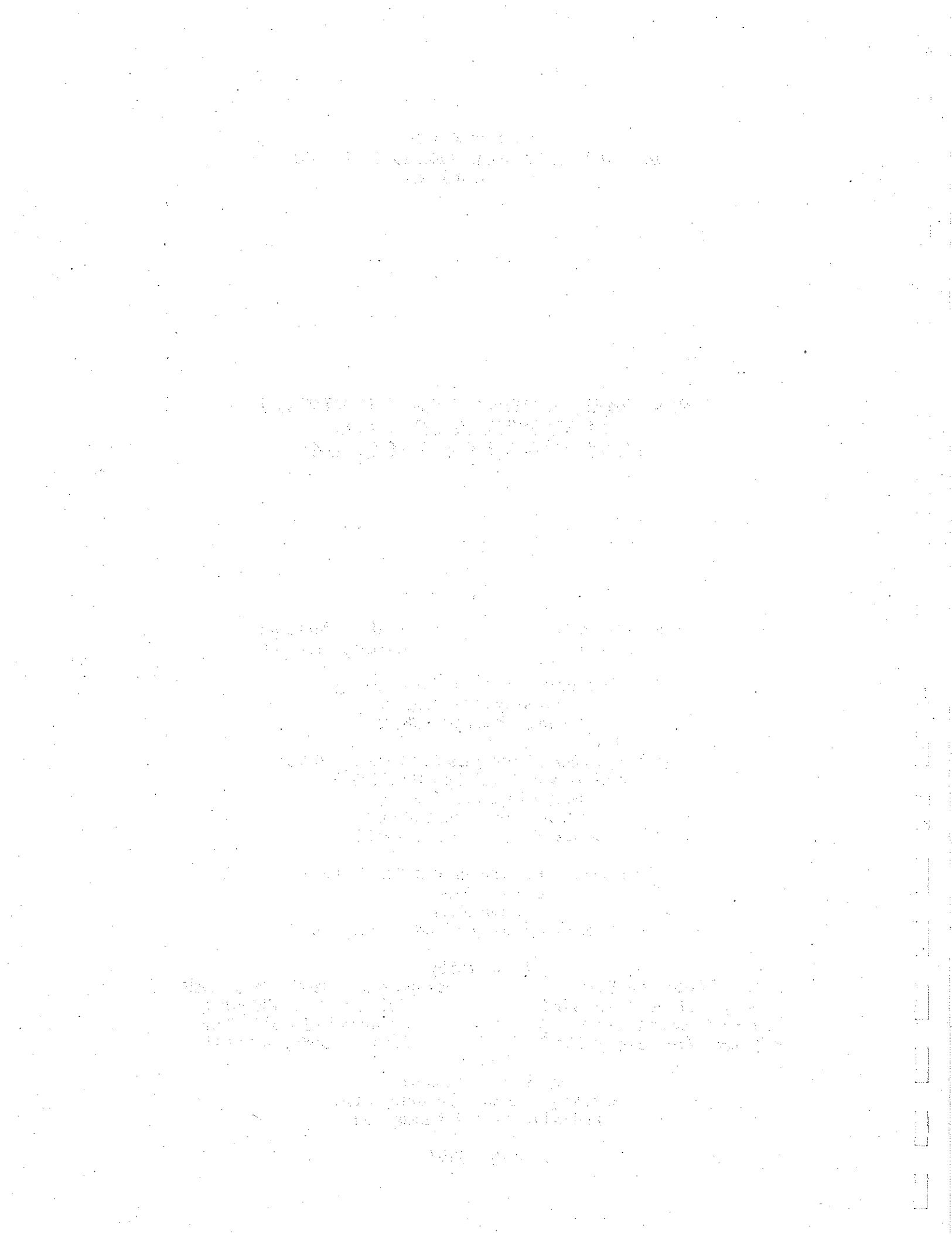
Sponsored by

**Washington State
Transportation Commission**
Department of Transportation
Olympia, Washington 98504-7370

Transportation Northwest (TransNow)
University of Washington
135 More Hall, Box 354802
Seattle, Washington 98195

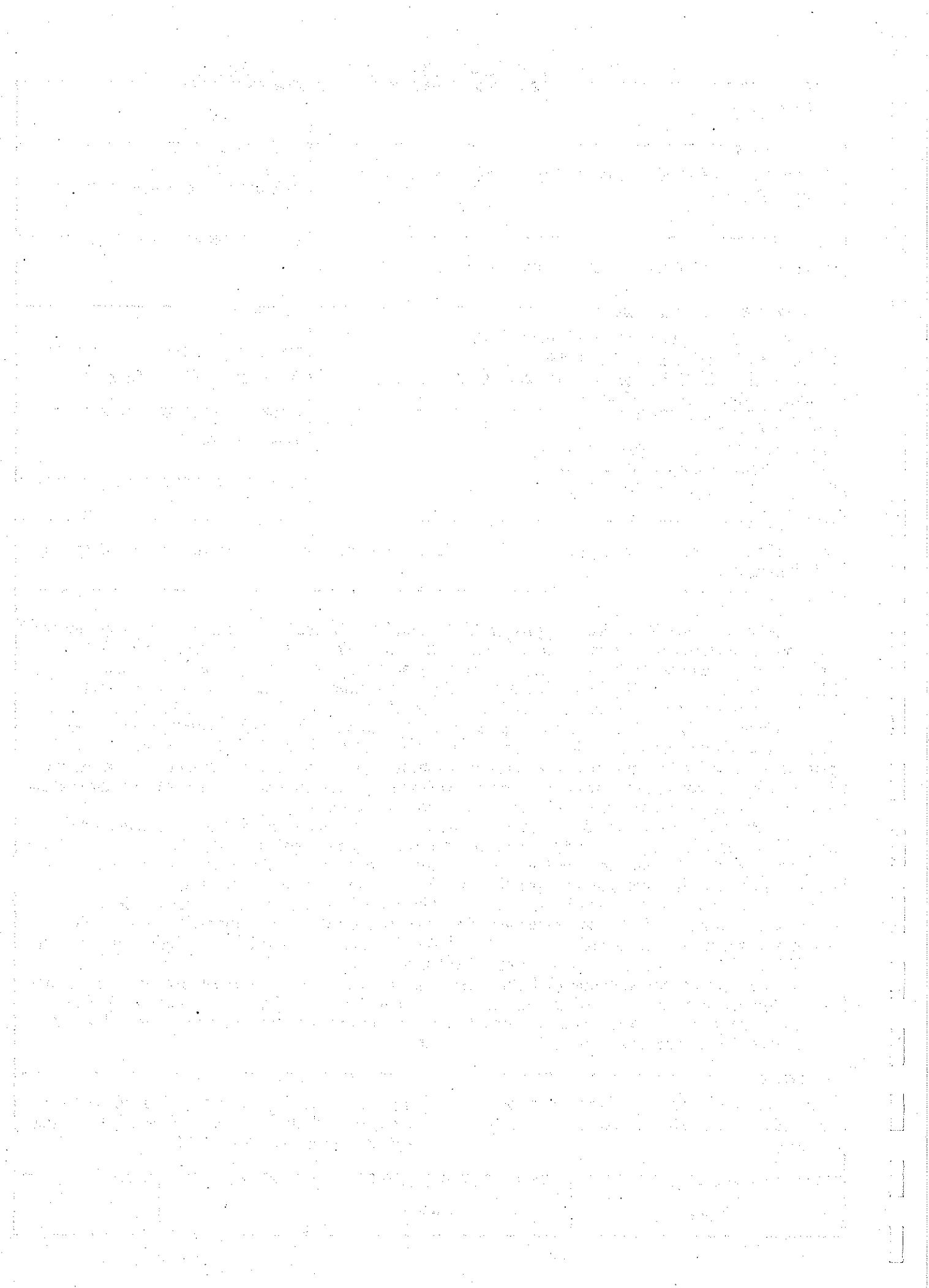
and in cooperation with
U.S. Department of Transportation
Federal Highway Administration

August 1997



TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO. WA-RD 442.2	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.	
4. TITLE AND SUBTITLE Documentation of TSMC Software That Interfaces with Traffic Analysis Programs		5. REPORT DATE August 1997	
6. PERFORMING ORGANIZATION CODE			
7. AUTHOR(S) Cynthia E. Taylor and Deirdre R. Meldrum		8. PERFORMING ORGANIZATION REPORT NO.	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Washington State Transportation Center (TRAC) University of Washington, Box 354802 University District Building; 1107 NE 45th Street, Suite 535 Seattle, Washington 98105-4631		10. WORK UNIT NO.	
		11. CONTRACT OR GRANT NO. Agreement T9903, Task 70	
12. SPONSORING AGENCY NAME AND ADDRESS Research Office Washington State Department of Transportation Transportation Building, MS 47370 Olympia, Washington 98504-7370		13. TYPE OF REPORT AND PERIOD COVERED Technical report	
		14. SPONSORING AGENCY CODE	
15. SUPPLEMENTARY NOTES This study was conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.			
16. ABSTRACT A Fuzzy Logic Ramp Metering Algorithm will address the needs of Seattle's freeway system and overcome limitations of the existing ramp metering algorithm. This project progressed toward implementing and testing a fuzzy neural ramp metering algorithm on-line at the Traffic Systems Management Center (TSMC) for the Washington State Department of Transportation's Northwest Region. Improvements were made to neural network predictors to allow better generalization. Code was written for the fuzzy ramp metering algorithm and its interface with the pre-existing TSMC code. Of the new code written, approximately 95 percent of it was for the interface, and only 5 percent of it was for the ramp metering algorithm itself. Interfacing the fuzzy controller with the existing TSMC software required modification of 16 pre-existing files related to the ramp metering database, real-time skeleton, and ramp metering and data collector communications. A method was developed and code was written to directly send metering rates from the VAX computer to the 170 computer and to implement them, whereas previously only a metering rate adjustment had been possible. The operator interface was designed and code was written to enter fuzzy tuning parameters and fuzzy equations. The specifications for each new parameter were designed. Although this code was written, it has not yet been implemented on-line because of time constraints. Preparation for on-line implementation required more time than anticipated because of the unexpected complexity of the pre-existing TSMC code. On-line implementation and testing will proceed on a WSDOT/TransNow project that begins in September 1997. In addition to software design, further planning was necessary to ensure smooth implementation and quality performance. The testing plan was developed in greater detail to include software quality testing. Primary and backup study sites were chosen, and an evaluation technique was selected. A risk assessment plan was developed to mitigate future problems.			
17. KEY WORDS Artificial neural networks (ANN), fuzzy logic controller (FLC), traffic data prediction, ramp metering		18. DISTRIBUTION STATEMENT No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22616	
19. SECURITY CLASSIF. (of this report) None	20. SECURITY CLASSIF. (of this page) None	21. NO. OF PAGES	22. PRICE



DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. This document is disseminated through the Transportation Northwest (TransNow) Regional Center under the sponsorship of the U.S. Department of Transportation UTC Grant Program and through the Washington State Department of Transportation. The U.S. Government assumes no liability for the contents or use thereof. Sponsorship for the local match portion of the research project was provided by the Washington State Department of Transportation. The contents do not necessarily reflect the official views or policies of the U.S. Department of Transportation, Washington State Transportation Commission, or Washington State Department of Transportation. This report does not constitute a standard, specification, or regulation.

TABLE OF CONTENTS

Section

I. CODE STRUCTURE

 Directory Listing
 Tag Files
 Calling Trees

II. VAX-170 COMMUNICATIONS

 Description of VAX-170 Communications
 Diagram of VAX-170 Communications
 RMDC-COMM

III. REAL-TIME SOFTWARE

 TMS_Startup
 TMS_Shutdown
 Real-Time Skeleton
 Error Handling

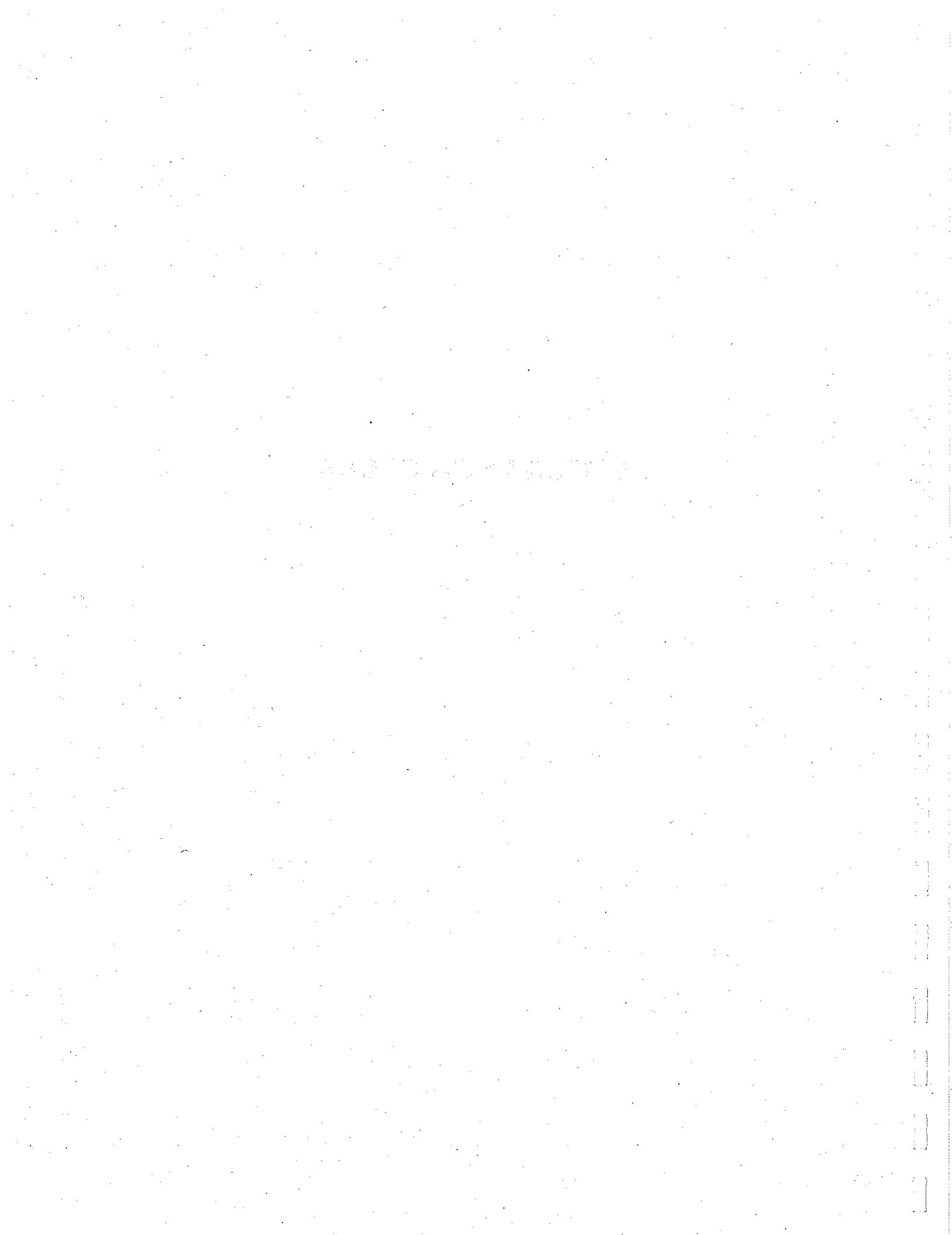
IV. TRAFFIC ANALYSIS PROGRAMS (TAPS)

 Bottleneck
 Pseudo Code
 Bottleneck Table
 Action Codes
 Equation Parsing Diagram
 Watch_bottleneck
 Incident Detection
 Pseudo Code
 Incident Table
 State Transition Diagram

V. DATABASES

 Files Associated with Building Databases
 RMDB
 Default Specifications in RMDB
 Structure of RMDB
 BUILD_RMDB
 RMDB-JOURNAL
 Databases
 RTDB
 Structure of RTDB
 BUILD_RTDB

I. CODE STRUCTURE



TMS_CODE Directories

<u>DIR</u>	<u># FILES</u>	<u>KB</u>
TMS_CODE	20	330
COMM_PROT	54	3005
OPC_COMM	177	21665
RMDC_COMM	34	2433
VMS_COMM	33	2881
FDDB	38	1235
CCTVDB	46	1788
GBLDB	34	1019
GCDB	34	966
OPRTVDB	46	1646
RMDB	72	4175
TEST	82	933
SCHEDDB	31	1145
TOKEN	15	290
VAXPORTDB	56	1965
VMSDB	45	3046
MAP_DIFFERENCES	44	2725
NOAA_MONITOR	6	363
RT_SKELETON	127	3219
TMS_INCLUDE	10	322
TMS_LIBRARY	67	2472
UPI_XMIT	7	733
TMS_CODE TOTAL: 22 DIRS	1078 FILES	58 MB
TMS_EXE TOTAL: 39 DIRS	1813 FILES	150 MB
GRAND TOTAL: 61 DIRS	2891 FILES	208 MB

TAGS FILE

```

FP_left_arrow_to_buffer tms_library/vms_lib.c    /^int  FP_left_arrow_to_buffer(buffer, n_c
hars, bfr_/
FP_position_cursor      tms_library/vms_lib.c    /^int  FP_position_cursor(buffer, row, col
)-
FP_right_arrow_to_buffer       tms_library/vms_lib.c  /^int  FP_right_arrow_to_buffer(bu
ffer, n_chars, bfr/
HM_command      comm_prot/reset_modem.c  /^int  HM_command(channel, command_str)/
HM_hangup        upi_xmit/upi_xmit.c   /^int  HM_hangup(channel)/
HM_initialize    comm_prot/reset_modem.c  /^int  HM_initialize(channel)/
HM_reset         upi_xmit/upi_xmit.c   /^int  HM_reset(channel)/
HM_turn_off_DTR upi_xmit/upi_xmit.c   /^int  HM_turn_off_DTR(channel)/
HM_view  comm_prot/reset_modem.c  /^int  HM_view(channel)/
Mactv_anal       rt_skeleton/actv_anal.c  /^main()
Mbottleneck     rt_skeleton/bottleneck.c  /^main()
Mbuild_cctvdb   fddb/cctvdb/build_cctvdb.c  /^main()
Mbuild_fmdb     rt_skeleton/build_fmdb.c   /^main()
Mbuild_gbldb    fddb/gbldb/build_gbldb.c  /^main()
Mbuild_gcdb     fddb/gcdb/build_gcdb.c   /^main()
Mbuild_oprtvdb  fddb/oprtvdb/build_oprtvdb.c /^main()
Mbuild_rmdb     fddb/rmdb/build_rmdb.c   /^main()
Mbuild_rtddb    rt_skeleton/build_rtddb.c  /^main()
Mbuild_scheddb  fddb/scheddb/build_scheddb.c /^main()
Mbuild_vaxportdb fddb/vaxport/build_vaxportdb.c /^main()
Mbuild_vmsdb    fddb/vmsdb/build_vmsdb.c  /^main()
Mcomm_stats_rpt rt_skeleton/comm_stats_rpt.c /^main()
Mcount_tms_lines count_tms_lines.c   /^main()
Mcrack_fmdb_dailyfil rt_skeleton/crack_fmdb_dailyfil.c  /^main()
Mcrack_fmdb_namefile rt_skeleton/crack_fmdb_namefile.c /^main()
Mcrack_fmdb_snapshot rt_skeleton/crack_fmdb_snapshot.c /^main()
Mdel_actvdb     rt_skeleton/del_actvdb.c   /^main()
Mdel_cctvdb    fddb/cctvdb/del_cctvdb.c  /^main()
Mdel_fmdb      rt_skeleton/del_fmdb.c   /^main()
Mdel_gbldb     fddb/gbldb/del_gbldb.c  /^main()
Mdel_gcdb      fddb/gcdb/del_gcdb.c   /^main()
Mdel_oprtvdb   fddb/oprtvdb/del_oprtvdb.c /^main()
Mdel_rmdb      fddb/rmdb/del_rmdb.c   /^main()
Mdel_rtddb     rt_skeleton/del_rtddb.c  /^main()
Mdel_scheddb   fddb/scheddb/del_scheddb.c /^main()
Mdel_vaxportdb fddb/vaxport/del_vaxportdb.c /^main()
Mdel_vmsdb    fddb/vmsdb/del_vmsdb.c  /^main()
Mdumydata      rt_skeleton/dumydata.c  /^main()
Mevent_logger   rt_skeleton/event_logger.c /^main()
Mfdb_aggr      rt_skeleton/fmdb_aggr.c  /^main()
Mfdb_archiver  rt_skeleton/fmdb_archiver.c /^main()
Minc_detect    rt_skeleton/inc_detect.c /^main()
Mmon_event_log rt_skeleton/mon_event_log.c /^main()
Mnoaa_monitor  noaa_monitor/noaa_monitor.c /^main()
Mpatch_cctvdb  fddb/cctvdb/patch_cctvdb.c /^main()
Mpatch_gbldb   fddb/gbldb/patch_gbldb.c /^main()
Mpatch_gcdb    fddb/gcdb/patch_gcdb.c  /^main()
Mpatch_oprtvdb fddb/oprtvdb/patch_oprtvdb.c /^main()
Mpatch_rmdb    fddb/rmdb/patch_rmdb.c  /^main()
Mpatch_scheddb fddb/scheddb/patch_scheddb.c /^main()
Mpatch_vaxportdb fddb/vaxport/patch_vaxportdb.c /^main()
Mpatch_vmsdb   fddb/vmsdb/patch_vmsdb.c  /^main()
Mread_rtddb    rt_skeleton/read_rtddb.c /^main()
Mreset_modem   comm_prot/reset_modem.c /^main()
Mrtskeleton    rt_skeleton/rt_skeleton.c  /^main()
Mshutdown_opc_comm comm_prot/opc_comm/shutdown_opc_comm.c /^main()
Mshutdown_rmdc_comm comm_prot/rmdc_comm/shutdown_rmdc_comm.c /^main()
Mshutdown_vms_comm comm_prot/vms_comm/shutdown_vms_comm.c /^main()
Msnap_loop_err rt_skeleton/snap_loop_err.c /^main()
Mstn_aggr      rt_skeleton/stn_aggr.c  /^main()
Mswitch_tty    comm_prot/switch_tty.c  /^main()

```

```

Mt_token          fddb/token/t_token.c    /^main()
Mtms_shutdown    rt_skeleton/tms_shutdown.c  /^main()
Mtms_startup     rt_skeleton/tms_startup.c   /^main()
Mupi_xmit        upi_xmit/upi_xmit.c    /^main()
Mwatch_actv_anal rt_skeleton/watch_actv_anal.c  /^main()
Mwatch_bottleneck rt_skeleton/watch_bottleneck.c  /^main()
Mwatch_fmdb      rt_skeleton/watch_fmdb.c    /^main()
Mwatch_rmdc      rt_skeleton/watch_rmdc.c    /^main()
_align comm_prot/opc_comm/opc_comm.c  /^struct queue_hdr _align (QUADWORD) L10_free_li
st/
abort_data_retrieval comm_prot/opc_comm/opc_comm_sub.c  /^void abort_data_retriev
al(mpno, msg_tail)/
abort_tx_at_driver comm_prot/opc_comm/opc_comm_sub.c  /^void abort_tx_at_driver
(port_no)/
abort_tx_wait_rx  comm_prot/opc_comm/opc_comm_sub.c  /^void abort_tx_wait_rx(p
ort_no)/
add_cluster       comm_prot/opc_comm/opc_vms_sub.c  /^unsigned short add_cluster(cst_
name, cst_body, bo/
add_library       comm_prot/opc_comm/opc_vms_sub.c  /^unsigned short add_library(sign
_name, lib_name, l/
add_message       comm_prot/opc_comm/opc_vms_sub.c  /^unsigned short add_message(sign
_name, msg_name, m/
add_que_to_lib    tms_library/vms_lib.c   /^unsigned short add_que_to_lib(action_code, sign
_n/
add_queue         comm_prot/opc_comm/opc_vms_sub.c  /^unsigned short add_queue(sign_n
ame, que_name, que/
add_schedule      comm_prot/opc_comm/opc_vms_sub.c  /^unsigned short add_schedule(sch
ed_ptr, unit_no, o/
add_to_list       tms_library/link_sub.c  /^void add_to_list(entry, pred)
add_to_list_head_i tms_library/intlk_queue.c  /^int add_to_list_head_i(entry, l
ist_head)/
add_to_list_tail_i tms_library/intlk_queue.c  /^int add_to_list_tail_i(entry, l
ist_head)/
add_to_partial_list comm_prot/opc_comm/opc_comm_sub.c  /^void add_to_partial_lis
t(rx_ubfr)/
add_to_port_device_table fddb/vaxport/build_vaxportdb.c /^void add_to_port_device
_table(VAXPortName, Device/
add_to_rel_list   tms_library/vms_lib.c   /^void add_to_rel_list(vmsdb_t1, entry, rel_pred)
/
add_to_scheddb_list tms_library/sched_lib.c /^void add_to_scheddb_list(scheddb_t1, en
try, rel_p/
alloc_and_queue_empty_jhub comm_prot/tms_comm_sub.c  /^void alloc_and_queue_em
pty_jhub(data_size)/
alloc_and_queue_empty_vms  comm_prot/vms_comm/vms_comm_sub.c  /^void alloc_and_
queue_empty_vms(data_size)/
alloc_vms_bfr_list    comm_prot/vms_comm/vms_comm_sub.c  /^void alloc_vms_bfr_list
()
assemble_library    tms_library/vms_lib.c   /^unsigned short assemble_library(lib_bod
y, lib_siz/
assign_initial_camera comm_prot/opc_comm/cctv_comm_sub.c  /^void assign_initial_cam
era(port_no)/
assign_joystick_to_monitor comm_prot/opc_comm/cctv_comm_sub.c  /^void assign_joy
stick_to_monitor(rx_ubfr)/
assign_sys$input     tms_library/kb_func.c  /^void assign_sys$input()
attn_ast_func      comm_prot/opc_comm/opc_comm_sub.c  /^void attn_ast_func(param)/
auto_menu          comm_prot/rmdc_comm/rmdc_comm_sub.c  /^int auto_menu()
break_scheddb_block tms_library/sched_lib.c /^void break_scheddb_block(scheddb_t1, bl
ock, req_s/
break_vmsdb_block  tms_library/vms_lib.c   /^void break_vmsdb_block(vmsdb_t1, block,
req_size)/
breakthru         tms_library/proc_cntrl.c  /^unsigned long breakthru(terminal, messa
ge)/
breakthru_with_parameter tms_library/proc_cntrl.c  /^unsigned long breakthru
_with_parameter(termid, pr/
build_DPW_DRA     comm_prot/opc_comm/opc_comm_sub.c  /^int build_DPW_DRA(tx_cmd, drcb_
ptr, list_head)/

```

build_FP_arrow_msg	tms_library/vms_lib.c	/*int build_FP_arrow_msg(buffer, bfr_size, cm)/
build_FP_flashing_msg	tms_library/vms_lib.c	/*int build_FP_flashing_msg(buffer, bfr_size, cm)/
build_FP_message	tms_library/vms_lib.c	/*unsigned short build_FP_message(buffer, bfr_size, /
build_FP_msg_header	tms_library/vms_lib.c	/*void build_FP_msg_header(buffer, auxout)/
build_FP_multiphase_msg	tms_library/vms_lib.c	/*int build_FP_multiphase_msg(buffer, bfr_size, cm, /
build_FP_queue	tms_library/vms_lib.c	/*unsigned short build_FP_queue(buffer, bfr_size, q/
build_FP_static_msg	tms_library/vms_lib.c	/*int build_FP_static_msg(buffer, bfr_size, cm)/
build_all_queue	comm_prot/vms_comm/vms_comm_sub.c	/*void build_all_queue(unit_no)/
build_and_queue_170_date_time	comm_prot/rmdc_comm/rmdc_comm_sub.c	/*void build_and_queue_170_date_time(unit_no)/
build_and_queue_170_msg	comm_prot/rmdc_comm/rmdc_comm_sub.c	/*void build_and_queue_170_msg(unit_no, command, pa/
build_and_queue_GIM	comm_prot/opc_comm/opc_comm_sub.c	/*void build_and_queue_GIM(unit_no, gim_code, data,/
build_and_queue_HDR	comm_prot/opc_comm/opc_comm_sub.c	/*void build_and_queue_HDR(rx_ubfr, command, param)/
build_and_queue_RDAT	comm_prot/opc_comm/opc_comm_sub.c	/*void build_and_queue_RDAT(elem_name, elem_type, e/
build_and_queue_REJL	comm_prot/opc_comm/opc_comm_sub.c	/*void build_and_queue_REJL(rx_ubfr, rej_code, data/
build_and_queue_SCH_reply	comm_prot/opc_comm/opc_vms_sub.c	/*void build_and_queue_SCH_reply(rx_ubfr, sched_ent/
build_and_queue_TP	comm_prot/opc_comm/opc_comm_sub.c	/*void build_and_queue_TP(rx_ubfr)/
build_and_queue_VAX_TIME	comm_prot/opc_comm/opc_comm_sub.c	/*void build_and_queue_VAX_TIME(rx_ubfr)/
build_and_queue_VMS_auxout	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_auxout(unit_no, auxout)/
build_and_queue_VMS_blank_sign	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_blank_sign(unit_no, clea/
build_and_queue_VMS_diag	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_diag(unit_no, slot_no)/
build_and_queue_VMS_display	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_display(unit_no, content/
build_and_queue_VMS_enquire	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_enquire(unit_no, config/_
build_and_queue_VMS_exit	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_exit(unit_no, exit_reque/
build_and_queue_VMS_load	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_load(unit_no, lib_en/
build_and_queue_VMS_load	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_load(unit_no, slot_no, c/
build_and_queue_VMS_recall	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_recall(unit_no, slot_no, /
build_and_queue_VMS_reply	comm_prot/opc_comm/opc_vms_sub.c	/*void build_and_queue_VMS_reply(rx_ubfr, action_co/
build_and_queue_VMS_reset	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_reset(unit_no)/
build_and_queue_VMS_view	comm_prot/vms_comm/vms_comm_sub.c	/*void build_and_queue_VMS_view(unit_no, slot_no)/
build_and_queue_load_params	comm_prot/rmdc_comm/rmdc_comm_sub.c	/*void build_and_queue_load_params(unit_no, lp_grou/
build_and_send_video_seq_cmd	comm_prot/opc_comm/cctv_comm_sub.c	/*unsigned short build_and_send_video_seq_cmd(monit/
build_and_send_video_switch_cmd	comm_prot/opc_comm/cctv_comm_sub.c	/*unsigned short build_and_send_video_switch_cmd(mo/
build_bottleneck_table	rt_skeleton/bottleneck.c	/*build_bottleneck_table(ba_table, table_size)/
build_dailyfile_header	rt_skeleton/fmdb_archiver.c	/*unsigned long build_dailyfile_h

```

eader(filename)
build_dev_port_name_string      fddb/vaxport/build_vaxportdb.c /*void build_dev_port_name
e_string(index, buffer, Dr/
build_dft_stn_aggr_eqns fddb/rmdb/rmdb_sub.c /*void build_dft_stn_aggr_eqns(tl)
build_fmdb_daily_filename     tms_library/fmdb_lib.c /*unsigned long build_fmdb_daily_
filename(start_time)
build_fmdb_snapshot_filename   tms_library/fmdb_lib.c /*unsigned long build_fmdb_snapshot_
ot_filename(start_
build_full_name fddb/fddb_sub.c /*void build_full_name(full_name, path, file_name, f/
build_inc_det_table        rt_skeleton/inc_detect.c /*build_inc_det_table(id_table, ta_
ble_size)
build_snapshot_header         rt_skeleton/fmdb_archiver.c /*unsigned long build_snapshot_he_
ader()
build_stn_aggr_table        rt_skeleton/stn_aggr.c /*build_stn_aggr_table(sa_table, table_siz_
e)
build_tap_error tms_library/tap_sub.c /*void build_tap_error(line_no, buffer, nchars, t_
ex)
build_up_arrow fddb/fddb_sub.c /*void build_up_arrow(buffer, column, n_col)
byte_to_float    tms_library/format_db.lib.c /*void byte_to_float(str, n_dec)
byte_to_table   tms_library/table_sub.c /*void byte_to_table(pointer, byte)
calc_FP_lib_slot_no       tms_library/vms.lib.c /*unsigned short calc_FP_lib_slot_no(vmsd_
b_t1, vmsdb/
calc_FP_msg_time          tms_library/vms.lib.c /*unsigned short calc_FP_msg_time(msg_ent_
ry, vmsdb/
calc_actv_anal            rt_skeleton/actv_anal.c /*void calc_actv_anal()
calc_all_lib_slot_no      comm_prot/opc_comm/opc_vms_sub.c /*unsigned short calc_all_
_lib_slot_no(sign_name, ms/
calc_bottleneck           rt_skeleton/bottleneck.c /*calc_bottleneck(ba_table)
calc_check                comm_prot/tms_comm_sub.c /*unsigned short calc_check(message, leng_
th)
calc_fletcher_checksum    comm_prot/vms_comm/vms_comm_sub.c /*unsigned short calc_fle_
tcher_checksum(msg, msg_le)
calc_inc_det              rt_skeleton/inc_detect.c /*calc_inc_det(id_table)
calc_next_20sec_time      rt_skeleton/fmdb_archiver.c /*unsigned long calc_next_20sec_t_
ime(ref_time, next/
calc_next_22sec           rt_skeleton/rt_skeleton.c /*unsigned long calc_next_22sec(next_22se_
c_time)
calc_next_due_time         comm_prot/opc_comm/opc_vms_sub.c /*unsigned short calc_nex_
t_due_time(due_time, inter/
calc_offsets               fddb/fddb_sub.c /*void calc_offsets(nt, ref_base)
calc_stn_aggr             rt_skeleton/stn_aggr.c /*calc_stn_aggr(sa_table)
calc_table_checksum        tms_library/table_sub.c /*unsigned short calc_table_checksum(tabl_
e_base, la/
cancel_tms_sched_timer   comm_prot/opc_comm/opc_vms_sub.c /*void cancel_tms_sched_t_
imer()
center_justify             tms_library/misc_func.c /*void center_justify(field_bfr, width)
change_VAXPort_flag        fddb/vaxport/patch_vaxportdb.c /*void change_VAXPort_flag(vaxport_
db_t1, port_no)
change_term_char           tms_library/kb_func.c /*unsigned long change_term_char(ttchan,
basic_char/
char_in_set                fddb/token/token.c /*char_in_set(src, table, curr_row)
check_DOW_mask              fddb/scheddb/scheddb_sub.c /*unsigned long check_DOW_mask(string)
check_ERR_REQ_bit           comm_prot/rmdc_comm/rmdc_comm_sub.c /*void check_ERR_REQ_bit(
unit_no)
check_JHUB                  comm_prot/tms_comm_sub.c /*struct jhub *check_JHUB(msg, jh_ubfr)
check_RFD                   comm_prot/opc_comm/opc_comm_sub.c /*struct opc_rfd *check_RFD(msg,
drccb_ptr)
check_SCH_action             comm_prot/opc_comm/opc_vms_sub.c /*unsigned short check_SC_
H_action(action_code, devi/
check_SCH_date_time         comm_prot/opc_comm/opc_vms_sub.c /*unsigned short check_SC_
H_date_time(year, month, d/
check_SCH_interval           comm_prot/opc_comm/opc_vms_sub.c /*unsigned short check_SC_
H_interval(day, hour, minu/
check_VPT_name               comm_prot/tms_comm_sub.c /*void check_VPT_name(msg)
check_cluster                comm_prot/opc_comm/opc_vms_sub.c /*unsigned short check_cluster(cs_
t_name, cst_body, /

```

```

check_date      tms_library/misc_func.c /*unsigned long check_date(year, month, day) /
check_date_time fddb/scheddb/scheddb_sub.c      /*unsigned long check_date_time(string, date_time) /
check_drcb_poison      comm_prot/opc_comm/opc_comm_sub.c      /*void check_drcb_poison(
drcb_ptr, msg) /
check_due_in_dow      tms_library/sched_lib.c /*unsigned long check_due_in_dow(due_time
64, dow_ma) /
check_filename tms_library/misc_func.c /*unsigned char *check_filename(file_name, buffer)
/
check_fmdb_filename_format      tms_library/fmdb_lib.c /*unsigned long check_fmdb_filename
me_format(filename) /
check_for_NO_CARRIER upi_xmit/upi_xmit.c      /*int check_for_NO_CARRIER(channel) /
check_interval fddb/scheddb/scheddb_sub.c      /*unsigned long check_interval(string, interval) /
check_library  comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short check_library(ac-
tion_code, sign_na) /
check_line      fddb/vmsdb/vmsdb_sub.c /*int check_line(vmsdb_t1, param_start, param_val-
ue) /
check_list_unit_no      comm_prot/tms_comm_sub.c      /*void check_list_unit_no(msg, li-
st_head) /
check_message  comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short check_message(si-
gn_name, msg_name) /
check_name      fddb/token/t_token.c /*check_name(table, name) /
check_number   fddb/scheddb/scheddb_sub.c      /*int check_number(string, bfr_ndx, max_d-
igits, max) /
check_opc_unit_inactive comm_prot/opc_comm/opc_comm_sub.c      /*void check_opc_unit_ina-
ctive(unit_no) /
check_oper_initials      fddb/scheddb/scheddb_sub.c      /*unsigned long check_oper_initia-
ls(string) /
check_port_inactive      comm_prot/tms_comm_sub.c      /*void check_port_inactive(port_n-
o) /
check_process_interactive      tms_library/proc_cntrl.c      /*unsigned long check_pro-
cess_interactive(interacti) /
check_queue      comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short check_queue(sign-
_name, que_name, q) /
check_reversible      tms_library/tap_sub.c /*unsigned char check_reversible (stn_loo-
p_name) /
check_rfd_lists comm_prot/opc_comm/opc_comm_sub.c      /*struct opc_rfd *check_rfd_lists
(handle, unit_no, / /
check_rmdb_changed      fddb/rmdb/patch_rmdb.c /*void check_rmdb_changed() /
check_rmdc_port_status      comm_prot/rmdc_comm/rmdc_comm_sub.c /*void check_rmdc_port_st-
atus(port_no) /
check_rpt_memory      rt_skeleton/comm_stats_rpt.c /*void check_rpt_memory() /
check_schedule  comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short check_schedule(s-
ched_ptr, unit_no, / /
check_test_packet      comm_prot/opc_comm/opc_comm_sub.c      /*void check_test_packet(
rx_ubfr) /
check_type_ahead      tms_library/kb_func.c /*unsigned long check_type_ahead(ttchan,
count, fir) /
check_valid_flag      rt_skeleton/actv_anal.c /*int check_valid_flag(n_loops, flag) /
check_vms_port_status      comm_prot/vms_comm/vms_comm_sub.c /*void check_vms_port_sta-
tus(port_no) /
chk_col_name      fddb/fddb_sub.c /*void chk_col_name(tl) /
classify_roadway      tms_library/fddb_lib.c /*int classify_roadway(stn_loop_name) /
clear_R0      tms_library/misc_func.c /*int clear_R0() /
clear_all_event_flags      tms_library/proc_cntrl.c      /*void clear_all_event_flags(ef_c-
luster) /
clear_joystick  comm_prot/opc_comm/cctv_comm_sub.c      /*unsigned short clear_joystick(m-
onitor, unit_no) /
clear_joystick_from_monitor      comm_prot/opc_comm/cctv_comm_sub.c      /*void clear_joys-
tick_from_monitor(rx_ubfr) /
clear_process_event_flag      rt_skeleton/rt_skeleton.c      /*void clear_process_even-
t_flag(proc_name, efc_name) /
close_comm_log_file      comm_prot/tms_comm_sub.c      /*unsigned long close_comm_log_fi-
le() /
close_event_log_file      rt_skeleton/event_logger.c /*void close_event_log_file() /

```

```

close_files_for_exit    comm_prot/opc_comm/opc_comm_sub.c      /*void close_files_for_ex
it()
close_operator_log_file comm_prot/opc_comm/opc_comm_sub.c      /*unsigned long close_ope
rator_log_file()
cluster_menu     fddb/vmsdb/patch_vmsdb.c      /*void cluster_menu()
cnt_data_col     fddb/fddb_sub.c /*void cnt_data_col(tl)
code_byte_to_table tms_library/table_sub.c /*void code_byte_to_table(pointer, code,
byte)
code_to_table     tms_library/table_sub.c /*void code_to_table(pointer, code)
code_ulong_to_table tms_library/table_sub.c /*void code_ulong_to_table(pointer, code,
ulong)
code_ushort_to_table tms_library/table_sub.c /*void code_ushort_to_table(pointer, code
, ushort)
col_name_fnd     fddb/fddb_sub.c /*void col_name_fnd(tl)
column_layout    rt_skeleton/watch_actv_anal.c /*int column_layout()
column_name_line fddb/fddb_sub.c /*int column_name_line(tl, table_name, equal_sign,
d/
comb_adj_scheddb_block tms_library/sched_lib.c /*void comb_adj_scheddb_block(scheddb_tl,
block)
combine_adjacent_vmsdb_block tms_library/vms_lib.c /*void combine_adjacent_vmsdb_blo
ck(vmsdb_tl, block)
compare_VAX_time   tms_library/misc_func.c /*int compare_VAX_time(time1, time2)
complete_col      fddb/cctvdb/cctvdb_sub.c /*void complete_col(cctvdb_tl)
complete_new       fddb/fddb_sub.c /*void complete_new(tl)
connect_joystick   comm_prot/opc_comm/cctv_comm_sub.c /*unsigned short connect_
joystick(monitor, new_joy)
connect_to_mailbox tms_library/mailbox.c /*unsigned long connect_to_mailbox(mbx_na
me, channel)
constant_name      fddb/token/token.c /*unsigned char *constant_name(value)
convert_DB_type   rt_skeleton/comm_stats_rpt.c /*void convert_DB_type(DBType, type_text)
/
convert_HM_result  comm_prot/reset_modem.c /*unsigned char *convert_HM_result(result
_code)
convert_VAXPort_flag fddb/vaxport/vaxportdb_sub.c /*void convert_VAXPort_flag(flag,
flag_text)
convert_action_code tms_library/sched_lib.c /*unsigned char *convert_action_code(acti
on_code, s/
convert_dow_mask   tms_library/fddb_lib.c /*int convert_dow_mask(dow_chars, dow_mas
k)
convert_loop_error fddb/rmdb/rmdb_sub.c /*void convert_loop_error(error_byte, err
or_text)
convert_loop_status tms_library/fddb_lib.c /*void convert_loop_status(status_byte, s
tatus_text)
convert_non_print_to_space fddb/token/token.c /*void convert_non_print_to_space(
begin)
convert_perm_mask  tms_library/format_db_lib.c /*int convert_perm_mask(perm_char
s, perm_mask)
convert_rpt_status rt_skeleton/comm_stats_rpt.c /*void convert_rpt_status(status,
status_text)
convert_sched_type tms_library/sched_lib.c /*unsigned char *convert_sched_type(sched
_type)
convert_start_flag  fddb/vaxport/vaxportdb_sub.c /*unsigned char convert_start fla
g(start_flag)
convert_status     tms_library/fddb_lib.c /*void convert_status(status, status_text, check_
de/
convert_time_to_int fddb/fddb_sub.c /*int convert_time_to_int(hour, min, time_string)
convert_to_0x_form  fddb/vaxport/vaxportdb_sub.c /*void convert_to_0x_form(VAXPortN
ame)
convert_to_x_form   fddb/vaxport/vaxportdb_sub.c /*void convert_to_x_form(VAXPortNa
me)
copy_char         fddb/token/token.c /*copy_char(dest, src, table, curr_row, cnt)
copy_chk_cnt      fddb/token/token.c /*copy_chk_cnt(dest, src, table, curr_row, cnt)
copy_defaults     fddb/cctvdb/cctvdb_sub.c /*void copy_defaults(tl, default_index)
copy_driver_stats rt_skeleton/comm_stats_rpt.c /*void copy_driver_stats()
copy_esc_new_st   fddb/token/token.c /*copy_esc_new_st(dest, src, table, curr_row, cnt)
/

```

```

copy_flash      tms_library/vms_lib.c    /*void copy_flash(cm)/*^
copy_lib_add_slot_no   comm_prot/opc_comm/opc_vms_sub.c        /*void copy_lib_add_slot_
no(entry_bfr, mlq_body, n_/
copy_lib_del_slot_no   comm_prot/opc_comm/opc_vms_sub.c        /*void copy_lib_del_slot_
no(data_bfr; entry_bfr, n_/
copy_new_st     fddb/token/token.c      /*copy_new_st(dest, src, table, curr_row, cnt)/*^
copy_param_lines   fddb/fddb_sub.c /^int copy_param_lines /
copy_port_24hr_stats rt_skeleton/comm_stats_rpt.c  /*void copy_port_24hr_stats()/*^
copy_same_st     fddb/token/token.c      /*copy_same_st(dest, src, table, curr_row, cnt)/*^
copy_str         fddb/token/token.c      /*copy_str(dest, src, table, curr_row, cnt)/*^
copy_str_new_st  fddb/token/token.c      /*copy_str_new_st(dest, src, table, curr_row, cnt)/*^
/
copy_unit_24hr_stats rt_skeleton/comm_stats_rpt.c  /*void copy_unit_24hr_stats()/*^
copyl_char       fddb/token/token.c      /*copyl_char(dest, src, table, curr_row, cnt)/*^
copyl_chk_cnt    fddb/token/token.c      /*copyl_chk_cnt(dest, src, table, curr_row, cnt)/*^
copyl_new_st     fddb/token/token.c      /*copyl_new_st(dest, src, table, curr_row, cnt)/*^
copyl_same_st    fddb/token/token.c      /*copyl_same_st(dest, src, table, curr_row, cnt)/*^
copyu_char       fddb/token/token.c      /*copyu_char(dest, src, table, curr_row, cnt)/*^
copyu_chk_cnt    fddb/token/token.c      /*copyu_chk_cnt(dest, src, table, curr_row, cnt)/*^
copyu_new_st     fddb/token/token.c      /*copyu_new_st(dest, src, table, curr_row, cnt)/*^
copyu_same_st    fddb/token/token.c      /*copyu_same_st(dest, src, table, curr_row, cnt)/*^
count_active_loops fddb/rmdb/rmdb_sub.c  /*void count_active_loops(tl)
/
count_actv_anal_eqns rt_skeleton/actv_anal.c /^int count_actv_anal_eqns(eqn_file, n_eq
ns, sum_st/
count_lines      count_tms_lines.c      /^int count_lines(file_name, n_bytes, checksum, o
ut/
count_metered_lanes fddb/rmdb/rmdb_sub.c  /*void count_metered_lanes(tl)
/
count_port_entries fddb/vaxport/build_vaxportdb.c /*int count_port_entries(n_unit_e
ntries)/*^
count_speed_traps fddb/rmdb/rmdb_sub.c  /*void count_speed_traps(tl)
/
crack_FP_msg     tms_library/crack_fp_msg.c  /*int crack_FP_msg(buffer, max_chars, msg
_bfr, msg_/
crack_vms_message tms_library/vms_lib.c  /*unsigned short crack_vms_message(vmsdb_
msg, cm, v/
crc_opc_tx_msg   comm_prot/opc_comm/opc_comm_sub.c  /*void crc_opc_tx_msg(jh_ubfr)/*^
crc_rmdc_tx_msg  comm_prot/rmdc_comm/rmdc_comm_sub.c /*void crc_rmdc_tx_msg(jh_ubfr)/*^
create_comm_log_file comm_prot/tms_comm_sub.c  /*unsigned long create_comm_log_f
ile()/*^
create_event_log_file rt_skeleton/event_logger.c  /*void create_event_log_file()/*^
create_fmdb_daily_file rt_skeleton/fmdb_archiver.c /*unsigned long create_fmdb_daily
_file(filename, pa/
create_fmdb_snapshot_file rt_skeleton/fmdb_archiver.c /*unsigned long create_fm
db_snapshot_file(filename, /^
create_global_section tms_library/global_sub.c  /*unsigned long create_global_sec
tion(name, n_bytes)/*^
create_global_section_ident tms_library/global_sub.c  /*unsigned long create_glo
bal_section_ident(name, n/
create_journal_files  comm_prot/opc_comm/opc_comm_sub.c  /*void create_journal_fil
es()/*^
create_logical_name   tms_library/logical_name.c  /*unsigned long create_logical_na
me(logical_name, e/
create_mailbox      tms_library/mailbox.c  /*unsigned long create_mailbox(mbx_name, channel,
b/
create_operator_log_file comm_prot/opc_comm/opc_comm_sub.c  /*unsigned long c
reate_operator_log_file()/*^
create_rpt_file rt_skeleton/comm_stats_rpt.c  /*void create_rpt_file()/*^
cs_titles        rt_skeleton/watch_rmdc.c  /*cs_titles()/*^
date_time_to_table tms_library/table_sub.c /*void date_time_to_table(pointer)/*^
deallocate_rfd_blocks comm_prot/opc_comm/opc_comm_sub.c  /*void deallocate_rfd blo
cks(drccb_ptr, debug_id)/*^
del_que_from_lib   tms_library/vms_lib.c  /*unsigned short del_que_from_lib(sign_na
me, que_na)/*^
delete_all_global_section tms_library/global_sub.c  /*unsigned long delete_al

```

```

l_global_section(name)
delete_cluster comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short delete_cluster(c
st_name, unit_no, /
delete_from_rel_list tms_library/vms_lib.c   /*void delete_from_rel_list(vmsdb_t1, ent
ry)
delete_from_scheddb_list    tms_library/sched_lib.c /*void delete_from_scheddb_list(s
cheddb_t1, entry)
delete_global_section tms_library/global_sub.c      /*unsigned long delete_global_sec
tion(name, ident)
delete_library comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short delete_library(s
ign_name, lib_name)
delete_logical_name   tms_library/logical_name.c    /*unsigned long delete_logical_na
me(logical_name, t/
delete_mailbox tms_library/mailbox.c   /*unsigned long delete_mailbox(channel)
delete_message comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short delete_message(s
ign_name, msg_name)
delete_queue   comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short delete_queue(sig
n_name, que_name, /
delete_scheddb_entry comm_prot/opc_comm/opc_vms_sub.c  /*int delete_scheddb_entr
y(sched_entry)
delete_schedule comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short delete_schedule(
sched_ptr, unit_no/
dial_HM_number upi_xmit/upi_xmit.c      /*int dial_HM_number(channel, phone_number, tone_
pu/
dial_upi_number upi_xmit/upi_xmit.c      /*int dial_upi_number()
dial_upi_send_msg upi_xmit/upi_xmit.c      /*int dial_upi_send_msg()
disable_enable_loop fddb/rmdb/patch_rmdb.c /*void disable_enable_loop(disable_flag)
disassemble_library tms_library/vms_lib.c  /*unsigned short disassemble_library(lib_
body, lib_
display_comm_stats rt_skeleton/watch_rmdc.c      /*display_comm_stats()
display_dynamic_params rt_skeleton/watch_actv_anal.c /*display_dynamic_params(EqnOffset
, eqn_win_id)/
display_eqn_data rt_skeleton/watch_actv_anal.c      /*display_eqn_data(EqnOffset, eqn_
win_id)/
display_error_data rt_skeleton/watch_rmdc.c      /*display_error_data()
display_history_data rt_skeleton/watch_bottleneck.c /*display_history_data()
display_loop_curr_data rt_skeleton/watch_fmdb.c   /*display_loop_curr_data(loop_win,
first_loop, last_/
display_loop_data rt_skeleton/watch_rmdc.c      /*display_loop_data()
display_loop_names rt_skeleton/watch_fmdb.c   /*display_loop_names(loop_win, fir
st_loop, last_loop/
display_loop_work_data rt_skeleton/watch_fmdb.c   /*display_loop_work_data(loop_win,
first_loop, last_/
display_meter_rate_data rt_skeleton/watch_rmdc.c /*display_meter_rate_data()
display_sc_labels rt_skeleton/watch_bottleneck.c /*display_sc_labels(sc)
display_static_labels rt_skeleton/watch_actv_anal.c /*display_static_labels(EqnOffset,
eqn_win_id, win_r/
display_station_curr_data rt_skeleton/watch_fmdb.c /*display_station_curr_dat
a()/
display_station_work_data rt_skeleton/watch_fmdb.c /*display_station_work_dat
a()/
display_status rt_skeleton/watch_rmdc.c      /*display_status()
display_stn_data rt_skeleton/watch_rmdc.c      /*display_stn_data()
display_stn_names rt_skeleton/watch_fmdb.c   /*display_stn_names()
display_trap_curr_data rt_skeleton/watch_fmdb.c   /*display_trap_curr_data()
display_trap_data rt_skeleton/watch_rmdc.c      /*display_trap_data()
display_trap_names rt_skeleton/watch_fmdb.c   /*display_trap_names()
display_trap_work_data rt_skeleton/watch_fmdb.c   /*display_trap_work_data()
display_vol_data rt_skeleton/watch_bottleneck.c /*display_vol_data(sc)
do_noth_same_st fddb/token/token.c   /*do_noth_same_st(dest, src, table, curr_row, cnt)
/
do_nothing fddb/token/token.c   /*do_nothing(dest, src, table, curr_row, cnt)
do_nothing_new fddb/token/token.c   /*do_nothing_new(dest, src, table, curr_row, cnt)
dow_mask_to_byte fddb/fddb_sub.c /*void dow_mask_to_byte(dow_mask, dow_chars)
dump_GMS_globals comm_prot/opc_comm/opc_comm_sub.c /*void dump_GMS_globals()
/

```

```

dump_UnitPdStats      fddb/vaxport/vaxportdb_sub.c    /*void dump_UnitPdStats(vaxportdb_
tl, UnitListDB)/
dump_UnitTable        fddb/vaxport/vaxportdb_sub.c    /*void dump_UnitTable(vaxportdb_tl, flag_
bits_colum/
dump_VAXPortPdStats  fddb/vaxport/vaxportdb_sub.c    /*void dump_VAXPortPdStats(vaxport
db_tl) /
dump_VAXPortTable    fddb/vaxport/vaxportdb_sub.c    /*void dump_VAXPortTable(vaxportdb
_tl) /
dump_actvdb_eqn      rt_skeleton/actv_lib.c        /*void dump_actvdb_eqn(EqnOffset) /
dump_actvdb_name_table rt_skeleton/actv_lib.c        /*void dump_actvdb_name_table() /
dump_actvdb_offsets   rt_skeleton/actv_lib.c        /*void dump_actvdb_offsets(offsets_address
s) /
dump_actvdb_params   rt_skeleton/actv_lib.c        /*void dump_actvdb_params() /
dump_actvdb_t1        rt_skeleton/actv_lib.c        /*void dump_actvdb_t1() /
dump_cctvdb_offsets   tms_library/fddb_lib.c       /*void dump_cctvdb_offsets(offsets_address
) /
dump_cctvdb_params   tms_library/fddb_lib.c       /* void dump_cctvdb_params(cctvdb_t1) /
dump_cctvdb_t1        tms_library/fddb_lib.c       /*void dump_cctvdb_t1(t1) /
dump_cctvdb_to_table_files fddb/cctvdb/cctvdb_sub.c /*void dump_cctvdb_to_table_
e_files(tl, col_index) /
dump_chk_table        fddb/token/t_token.c        /*void dump_chk_table(table, table_name, dump_mode
) /
dump_df_elem_data    tms_library/fmdb_lib.c       /*void dump_df_elem_data(buffer, name_lis
t, nt_inde/
dump_disassembled_library tms_library/vms_lib.c    /*void dump_disassembled_library(
) /
dump_drcb            comm_prot/opc_comm/opc_comm_sub.c /*void dump_drcb(msg, drcb_ptr, r
x_bfr) /
dump_drcb_hdr        comm_prot/opc_comm/opc_comm_sub.c /*void dump_drcb_hdr(msg, drcb_pt
r) /
dump_fddb_col_list   tms_library/fddb_lib.c       /*void dump_fddb_col_list(tl) /
dump_fddb_elements   fddb/fddb_sub.c             /*void dump_fddb_elements(tl, out_file, col_index
) /
dump_fddb_group_table tms_library/fddb_lib.c       /*void dump_fddb_group_table(tl) /
dump_fddb_line_buffer tms_library/fddb_lib.c       /*void dump_fddb_line_buffer(tl) /
dump_fddb_name_table tms_library/fddb_lib.c       /*void dump_fddb_name_table(tl) /
dump_fddb_offsets    tms_library/fddb_lib.c       /*void dump_fddb_offsets(offsets_address)
) /
dump_fddb_params     tms_library/fddb_lib.c       /*void dump_fddb_params(tl) /
dump_fmdb_col_offsets tms_library/fmdb_lib.c      /*void dump_fmdb_col_offsets() /
dump_fmdb_data_col   tms_library/fmdb_lib.c      /*int dump_fmdb_data_col(name_list, n_nam
es, data_c) /
dump_fmdb_name_list  tms_library/fmdb_lib.c      /*int dump_fmdb_name_list(name_list, n_na
mes, cab_n) /
dump_fmdb_name_table tms_library/fmdb_lib.c      /*void dump_fmdb_name_table() /
dump_fmdb_offsets    tms_library/fmdb_lib.c      /*void dump_fmdb_offsets(offsets_address)
) /
dump_fmdb_params     tms_library/fmdb_lib.c      /*void dump_fmdb_params() /
dump_fmdb_t1          tms_library/fmdb_lib.c      /*void dump_fmdb_t1() /
dump_gbldb_offsets   tms_library/fddb_lib.c       /*void dump_gbldb_offsets(offsets_address
) /
dump_gbldb_params    tms_library/fddb_lib.c       /*void dump_gbldb_params(tl) /
dump_gbldb_t1          tms_library/fddb_lib.c      /*void dump_gbldb_t1(tl) /
dump_gbldb_to_table_files fddb/gbldb/gbldb_sub.c /*void dump_gbldb_to_table_files(t
l, col_index) /
dump_gcdb_offsets    tms_library/fddb_lib.c       /*void dump_gcdb_offsets(offsets_address)
) /
dump_gcdb_params     tms_library/fddb_lib.c       /* void dump_gcdb_params(gcdb_t1) /
dump_gcdb_t1          tms_library/fddb_lib.c      /*void dump_gcdb_t1(gcdb_t1) /
dump_gcdb_to_table_files fddb/gcdb/gcdb_sub.c    /*void dump_gcdb_to_table_files(gc
db_t1, col_index) /
dump_jhub            comm_prot/tms_comm_sub.c    /*void dump_jhub(msg, jh_ubfr) /
dump_jhub_hdr        comm_prot/tms_comm_sub.c    /*void dump_jhub_hdr(msg, jh_ubfr) /
dump_list            comm_prot/tms_comm_sub.c    /*void dump_list(list_head) /
dump_list_rel        comm_prot/tms_comm_sub.c    /*void dump_list_rel(list_head) /

```

```

dump_loop_name_list      fddb/rmdb/rmdb_sub.c    /*void dump_loop_name_list(tl)
/
dump_loop_table_to_file fddb/rmdb/rmdb_sub.c    /*void dump_loop_table_to_file(tl, out_file,
                                                 col_in/
dump_mem                tms_library/dump_mem.c  /*void dump_mem(memptr, nbytes, addr_offset, base
                                                 _a/
dump_oprtvdb_offsets    tms_library/fddb_lib.c  /*void dump_oprtvdb_offsets(offsets_address
                                                 s)/
dump_oprtvdb_params     tms_library/fddb_lib.c  /* void dump_oprtvdb_params(tl) /
dump_oprtvdb_t1         tms_library/fddb_lib.c  /*void dump_oprtvdb_t1(tl) /
dump_oprtvdb_to_table_files   fddb/oprtvdb/oprtvdb_sub.c  /*void dump_oprtvdb_to_table_files(tl, col_index) /
dump_port_device_table  fddb/vaxport/build_vaxportdb.c /*void dump_port_device_table() /
dump_rmdb_offsets        tms_library/fddb_lib.c  /*void dump_rmdb_offsets(offsets_address)
/
dump_rmdb_params         tms_library/fddb_lib.c  /*void dump_rmdb_params(tl) /
dump_rmdb_t1             tms_library/fddb_lib.c  /*void dump_rmdb_t1(tl) /
dump_rmdb_to_table_files   fddb/rmdb/rmdb_sub.c  /*void dump_rmdb_to_table_files(tl
                                                 , col_index) /
dump_rtdb_col_offsets    tms_library/rtdb_lib.c  /*void dump_rtdb_col_offsets() /
dump_rtdb_name_table     tms_library/rtdb_lib.c  /*void dump_rtdb_name_table() /
dump_rtdb_offsets         tms_library/rtdb_lib.c  /*void dump_rtdb_offsets(offsets_address)
/
dump_rtdb_params          tms_library/rtdb_lib.c  /*void dump_rtdb_params() /
dump_rtdb_t1              tms_library/rtdb_lib.c  /*void dump_rtdb_t1() /
dump_scheddb_offsets      tms_library/fddb_lib.c  /*void dump_scheddb_offsets(offsets_address)
/
dump_scheddb_params       tms_library/fddb_lib.c  /*void dump_scheddb_params(scheddb_t1) /
dump_scheddb_t1            tms_library/fddb_lib.c  /*void dump_scheddb_t1(scheddb_t1) /
dump_schedule              fddb/scheddb/scheddb_sub.c  /*void dump_schedule() /
dump_speed_trap_table     fddb/rmdb/rmdb_sub.c  /*void dump_speed_trap_table(tl)
/
dump_speed_traps_to_file  fddb/rmdb/rmdb_sub.c  /*void dump_speed_traps_to_file(tl
                                                 , out_file, col_in/
dump_stn_aggr_list        fddb/rmdb/rmdb_sub.c  /*void dump_stn_aggr_list(tl)
/
dump_vaxportdb_offsets    tms_library/fddb_lib.c  /*void dump_vaxportdb_offsets(offsets_address)
/
dump_vaxportdb_params     tms_library/fddb_lib.c  /*void dump_vaxportdb_params(tl) /
dump_vaxportdb_t1          tms_library/fddb_lib.c  /*void dump_vaxportdb_t1(tl) /
dump_vaxportdb_to_file    fddb/vaxport/build_vaxportdb.c /*void dump_vaxportdb_to_file() /
dump_video_switch_mon_list fddb/cctvdb/patch_cctvdb.c  /*void dump_video_switch_
mon_list(print_unassigned) /
dump_vms_msg_struct       fddb/vmsdb/vmsdb_sub.c  /*void dump_vms_msg_struct(msg_name, cm) /
dump_vmsdb_offsets         tms_library/fddb_lib.c  /*void dump_vmsdb_offsets(offsets_address)
/
dump_vmsdb_params          tms_library/fddb_lib.c  /*void dump_vmsdb_params(vmsdb_t1) /
dump_vmsdb_t1              tms_library/fddb_lib.c  /*void dump_vmsdb_t1(tl) /
dump_vmsdb_to_table_files   fddb/vmsdb/vmsdb_sub.c  /*void dump_vmsdb_to_table_files(tl
                                                 , col_index) /
enable_disable_unit        comm_prot/opc_comm/opc_comm_sub.c  /*unsigned char enable_di
nable_unit(unit_no, enable/
err_titles                 rt_skeleton/watch_rmdc.c  /*err_titles()
establish_process_name    tms_library/proc_cntrl.c  /*unsigned long establish_process
_name(new_process_/
event_log_int_write       rt_skeleton/event_logger.c  /*void event_log_int_write(event_
code, str1, str2, /
event_logger_write_func    rt_skeleton/event_logger.c  /*void event_logger_write_func(bu
ffer, n_char) /
exit_multi_opc_comm       comm_prot/opc_comm/opc_comm_sub.c  /*void exit_multi_opc_com
m()/
exit_multi_rmdc_comm      comm_prot/rmdc_comm/rmdc_comm_sub.c  /*void exit_multi_rmdc_co
mm()/
exit_multi_vms_comm       comm_prot/vms_comm/vms_comm_sub.c  /*void exit_multi_vms_com
m()/
exit_test_opc_comm        comm_prot/opc_comm/opc_comm_sub.c  /*void exit_test_opc_com
m()

```

```

()/
exit_test_rmdc_comm      comm_prot/rmdc_comm/rmdc_comm_sub.c      /*void exit_test_rmdc_com
m()
exit_test_vms_comm      comm_prot/vms_comm/vms_comm_sub.c      /*void exit_test_vms_comm
()
extract_name   count_tms_lines.c      /*int extract_name(source, dest)
fddb_error     fddb/fddb_sub.c /*void fddb_error(tl, src_ptr, field_start, err_code)
fill_in_date_time   comm_prot/rmdc_comm/rmdc_comm_sub.c      /*void fill_in_date_time(
tx_ubfr)
find_actvdb_eqn_name   rt_skeleton/actv_lib.c /*int find_actvdb_eqn_name(eq_name)
find_col_name   fddb/fddb_sub.c /*void find_col_name(tl)
find_db_names    comm_prot/opc_comm/opc_comm_sub.c      /*int find_db_names(drcb_ptr)
find_duplicate_dev_addr fddb/vaxport/build_vaxportdb.c /*void find_duplicate_dev_addr()
find_err_text    fddb/fddb_sub.c /*unsigned char *find_err_text(err_code)
find_fddb_cl_name   tms_library/fddb.lib.c /*int find_fddb_cl_name(tl, data_col_name
)
find_fddb_gt_name   fddb/fddb_sub.c /*int find_fddb_gt_name(tl, group_name)
find_fddb_nt_name   tms_library/fddb.lib.c /*int find_fddb_nt_name(tl, element_name,
type, siz)
find_first_last_port_unit tms_library/find_first_last.c /*unsigned long find_firs
t_last_port_unit(db_type, /
find_gms_names   comm_prot/opc_comm/opc_comm_sub.c      /*int find_gms_names(drcb_ptr)
find_grp_name    fddb/fddb_sub.c /*void find_grp_name(tl)
find_msg_in_cluster   tms_library/vms.lib.c /*unsigned short find_msg_in_cluster(cst_
entry, sig)
find_msg_in_library   tms_library/vms.lib.c /*unsigned short find_msg_in_library(lib_
entry, msg)
find_msg_in_queue    tms_library/vms.lib.c /*unsigned short find_msg_in_queue(que_en
try, msg_n)
find_ptl_names    comm_prot/opc_comm/opc_comm_sub.c      /*int find_ptl_names(rx_ubfr, drcb
_ptr)
find_que_in_library   tms_library/vms.lib.c /*unsigned short find_que_in_library(lib_
entry, que)
find_slot_no_in_library tms_library/vms.lib.c /*unsigned short find_slot_no_in_library(
lib_entry, /
find_string      tms_library/misc_func.c /*unsigned char *find_string(string, buffer, buff
er)
find_usage_in_multiple_dbs fddb/vaxport/build_vaxportdb.c /*void find_usage_in_mult
iple_dbs()
find_vmsdb_entry_by_index   comm_prot/vms_comm/vms_comm_sub.c      /*void *find_vmsd
b_entry_by_index(list_head, index)
find_vmsdb_entry_by_name   tms_library/vms.lib.c /*void *find_vmsdb_entry_by_name(
list_head, name)
fit_eqn_in_win_col   rt_skeleton/watch_bottleneck.c /*int fit_eqn_in_win_col(st_row,
win_col, line_at_b)
fit_largest_unassigned   rt_skeleton/watch_bottleneck.c /*int fit_largest_unassigned(st_r
ow, win_col, line_/
fl_delta_time   tms_library/misc_func.c /*float fl_delta_time(start_time, curr_time)
flush_input      tms_library/kb_func.c /*void flush_input()
flush_mailbox    tms_library/mailbox.c /*unsigned long flush_mailbox(channel, buffer, bu
f_/
flush_opc_comm_queues   comm_prot/opc_comm/opc_comm_sub.c      /*void flush_opc_comm_que
ues()
flush_rmdc_queues   comm_prot/rmdc_comm/rmdc_comm_sub.c      /*void flush_rmdc_queues(
)
flush_type_ahead   tms_library/kb_func.c /*unsigned long flush_type_ahead(ttchan,
count, ios)
flush_vms_comm_queues   comm_prot/vms_comm/vms_comm_sub.c      /*void flush_vms_comm_que
ues()
fmt_write_comm_msg   comm_prot/tms_comm_sub.c /*void fmt_write_comm_msg(event_c
ode, mpu_name, msg)
follow_linked_list_fwd  comm_prot/opc_comm/opc_comm_sub.c /*follow_linked_list_fwd(l
ist_head, msg)
follow_linked_list_rev  comm_prot/opc_comm/opc_comm_sub.c /*follow_linked_list_rev(l
ist_head, msg)
follow_scheddb_list   tms_library/sched.lib.c /*void follow_scheddb_list(scheddb_tl, li

```

```

st_head) /
follow_vmsdb_list tms_library/vms_lib.c /*void follow_vmsdb_list(vmsdb_t1, list_h
ead) /
format_ascii_hex comm_prot/tms_comm_sub.c /*int format_ascii_hex(buffer, dat
a, n_bytes) /
format_db_element_for_output tms_library/format_db_lib.c /*int format_db_element_f
or_output(tl, buffer, colu/
format_event_log_msg tms_library/format_el_msg.c /*void format_event_log_msg(elm,
write_func) /
format_unit_name comm_prot/tms_comm_sub.c /*void format_unit_name(unit_no,
buffer) /
general_process_startup tms_library/proc_cntrl.c /*unsigned long general_process_s
tartup(process_nam/
get_1_char tms_library/kb_func.c /*unsigned long get_1_char(ttchan, ch, iosb) /
get_HM_input comm_prot/reset_modem.c /*int get_HM_input(channel, buffer, length, timeo
ut/
get_VMS_cluster fddb/vmsdb/vmsdb_sub.c /*int get_VMS_cluster(vmsdb_t1) /
get_VMS_library fddb/vmsdb/vmsdb_sub.c /*int get_VMS_library(vmsdb_t1) /
get_VMS_message fddb/vmsdb/vmsdb_sub.c /*int get_VMS_message(vmsdb_t1) /
get_VMS_queue fddb/vmsdb/vmsdb_sub.c /*int get_VMS_queue(vmsdb_t1) /
get_actv_anal_eqn fddb/rmdb/rmdb_sub.c /*int get_actv_anal_eqn(tl) /
get_actv_params_from_RMDB rt_skeleton/actv_lib.c /*unsigned long get_actv_params_f
rom_RMDB(RMDBColIn/
get_aggr_vol_occ comm_prot/opc_comm/opc_comm_sub.c /*void get_aggr_vol_occ(r
tdb_offset, first_col, las/
get_btl_neck_eqn fddb/rmdb/rmdb_sub.c /*int get_btl_neck_eqn(tl) /
get_cab_loop_name fddb/rmdb/rmdb_sub.c /*int get_cab_loop_name(dest, src) /
get_cluster comm_prot/opc_comm/opc_vms_sub.c /*unsigned short get_cluster(cst_
name, unit_no, cst/
get_cpu_time tms_library/proc_cntrl.c /*unsigned long get_cpu_time(cpu_time_adr
) /
get_device_name tms_library/proc_cntrl.c /*unsigned long get_device_name(logical_n
ame, dev_n/
get_fletcher_checksum comm_prot/vms_comm/vms_comm_sub.c /*void get_fletcher_check
sum(msg, msg_len, fp_check) /
get_group_name tms_library/uaf_sub.c /*unsigned long get_group_name(user_uic, group_na
me) /
get_hex fddb/token/token.c /*get_hex(dest, src, table, curr_row, cnt) /
get_inc_det_eqn fddb/rmdb/rmdb_sub.c /*int get_inc_det_eqn(tl) /
get_iochan tms_library/kb_func.c /*unsigned long get_iochan(device_name, iochan) /
get_library comm_prot/opc_comm/opc_vms_sub.c /*unsigned short get_library(sign
_name, lib_name, v/
get_message comm_prot/opc_comm/opc_vms_sub.c /*unsigned short get_message(sign
_name, msg_name, v/
get_mpu_name comm_prot/tms_comm_sub.c /*unsigned long get_mpu_name(mpu_no, mpu_
name) /
get_next_actv_line fddb/rmdb/rmdb_sub.c /*int get_next_actv_line(tl) /
get_next_btl_line fddb/rmdb/rmdb_sub.c /*int get_next_btl_line(tl) /
get_next_line fddb/fddb_sub.c /*unsigned char get_next_line(tl, stream) /
get_next_loop_name fddb/rmdb/rmdb_sub.c /*int get_next_loop_name(dest, src, defau
lt_cab) /
get_nibble comm_prot/rmdc_comm/rmdc_comm_sub.c /*unsigned char get_nibble(byte,
index) /
get_number_from_header tms_library/misc_func.c /*int get_number_from_header(label_string
, buffer) /
get_octal fddb/token/token.c /*get_octal(dest, src, table, curr_row, cnt) /
get_opc_buffer comm_prot/opc_comm/opc_comm_sub.c /*struct jhub *get_opc_buffer(dat
a_size) /
get_opc_comm_version comm_prot/opc_comm/opc_comm_sub.c /*void get_opc_comm_versi
on(rx_ubfr) /
get_other_process_id tms_library/proc_cntrl.c /*unsigned long get_other_process
_id(process_name, /
get_param fddb/fddb_sub.c /*void get_param(tl) /
get_param_value tms_library/kb_func.c /*int get_param_value(prompt, buffer, max_chars) /
get_pid tms_library/proc_cntrl.c /*unsigned long get_pid(pidadr) /
get_pin_assignments fddb/rmdb/rmdb_sub.c /*get_pin_assignments(tl) /

```

```

get_port_device_memory fddb/vaxport/build_vaxportdb.c /*void get_port_device_memory()
get_port_name comm_prot/reset_modem.c /*get_port_name(port_name)
get_proc_info tms_library/proc_cntrl.c /*unsigned long get_proc_info(item_code,
buffer_add/
get_process_name tms_library/proc_cntrl.c /*unsigned long get_process_name(
process_name)
get_process_priv tms_library/proc_cntrl.c /*unsigned long get_process_priv(
priv_quad_word)
get_pv fddb/fddb_sub.c /*int get_pv(bfr, b_offset, /
get_queue comm_prot/opc_comm/opc_vms_sub.c /*unsigned short get_queue(sign_n
ame, que_name, vms/
get_rmdc_buffer comm_prot/rmdc_comm/rmdc_comm_sub.c /*struct jhub *get_rmdc_buffer(da
ta_size) /
get_road_location comm_prot/opc_comm/opc_comm_sub.c /*void get_road_location(t
l, name_ptr, cl_index)
get_rpt_memory rt_skeleton/comm_stats_rpt.c /*void get_rpt_memory()
get_scheddb_block tms_library/sched_lib.c /*struct rel_list_hdr *get_scheddb_block(
scheddb_t1/
get_schedule comm_prot/opc_comm/opc_vms_sub.c /*unsigned short get_schedule(sch
ed_ptr, unit_no, s/
get_speed_param_name fddb/rmdb/rmdb_sub.c /*int get_speed_param_name(tl, dest, spee
d_ndx)
get_speed_trap_params fddb/rmdb/rmdb_sub.c /*int get_speed_trap_params(tl)/
get_stn_aggr_eqn fddb/rmdb/rmdb_sub.c /*int get_stn_aggr_eqn(tl)/
get_term_char tms_library/kb_func.c /*unsigned long get_term_char(ttchan, basic_char,
e/
get_terminal_name tms_library/proc_cntrl.c /*unsigned long get_terminal_name(
terminal_name)
get_tms_uaf_info tms_library/uaf_sub.c /*unsigned long get_tms_uaf_info(user_nam
e, acct, e/
get_tod_entry fddb/rmdb/rmdb_sub.c /*int get_tod_entry(tl)
get_token fddb/token/token.c /*get_token(dest, src, table)
get_tty_bit tms_library/fmdb_lib.c /*int get_tty_bit(efc_name, first_tty_bit, num_tt
y_
get_unit_name comm_prot/tms_comm_sub.c /*unsigned long get_unit_name(mpdu_no, opt
ions, unit/
get_unit_status comm_prot/opc_comm/opc_comm_sub.c /*unsigned char get_unit_status(u
nit_no)
get_user_name tms_library/proc_cntrl.c /*unsigned long get_user_name(user_name)
get_vms_buffer comm_prot/vms_comm/vms_comm_sub.c /*struct jhub *get_vms_buffer(dat
a_size) /
get_vmsdb_block tms_library/vms_lib.c /*struct rel_list_hdr *get_vmsdb_block(vmsdb_t1,
re/
getall_HM_SRegisters upi_xmit/upi_xmit.c /*int getall_HM_SRegisters(channel)
grp_name_fnd fddb/fddb_sub.c /*void grp_name_fnd(tl)
grp_name_special_case fddb/cctvdb/cctvdb_sub.c /*void grp_name_special_case(cctv
db_t1)
hash_password tms_library/uaf_sub.c /*unsigned long hash_password(user_name, password
,
highlight_max_rate rt_skeleton/watch_bottleneck.c /*highlight_max_rate()
idle_all comm_prot/opc_comm/opc_comm_sub.c /*int idle_all(unit_no)
idle_one comm_prot/opc_comm/opc_comm_sub.c /*short idle_one(unit_no, handle)
/
if comm_prot/opc_comm/opc_comm.c /* if ((cond_code != SS$_WASSET) && (cond_code !=
S/
in_order_layout rt_skeleton/watch_bottleneck.c /*int in_order_layout()
incident_detect rt_skeleton/inc_detect.c /*int incident_detect /* Modified Cali
fornia Algo/
init_actvdb_eqn rt_skeleton/actv_anal.c /*void init_actvdb_eqn(EqnOffset, NameTableNdx, R
MD/
init_actvdb_params rt_skeleton/actv_lib.c /*void init_actvdb_params(n_eqns)
init_actvdb_t1 rt_skeleton/actv_lib.c /*void init_actvdb_t1(base_address, offsets, t1)
/
init_attn_ast_jhub comm_prot/tms_comm_sub.c /*struct jhub *init_attn_ast_jhub(
jh_ubfr, port_no)
init_bfr_tracking_params comm_prot/vms_comm/vms_comm_sub.c /*void init_bfr_t

```

```

racking_params()
init_cctv_rx_jhub comm_prot/tms_comm_sub.c /*^struct jhub *init_cctv_rx_jhub(
jh_ubfr, rx_size, / comm_prot/tms_comm_sub.c /*^struct jhub *init_cctv_tx_jhub(
jh_ubfr, tx_size, / init_cctvdb_data_col_list fddb/cctvdb/cctvdb_sub.c /*^void init_cctvdb_data_co
l_list(tl, cctvdb_db_base)/
init_cctvdb_params fddb/cctvdb/cctvdb_sub.c /*void init_cctvdb_params(cctvdb_
tl, n_data_columns/
init_cctvdb_t1 tms_library/fddb_lib.c /*void init_cctvdb_t1(base_address, offsets, tl)/
init_cluster fddb/vmsdb/vmsdb_sub.c /*void init_cluster()
init_empty_jhub comm_prot/tms_comm_sub.c /*void init_empty_jhub(jh_ubfr, jhub_size
)*/
init_fddb_params fddb/fddb_sub.c /*void init_fddb_params(tl, n_data_columns, num_nt
_e/
init_fmdb_t1 tms_library/fmdb_lib.c /*void init_fmdb_t1(base_address, offsets)
/
init_gbldb_data_col_list fddb/gbldb/gbldb_sub.c /*void init_gbldb_data_col_list(tl
, gbldb_db_base)/
init_gbldb_params fddb/gbldb/gbldb_sub.c /*void init_gbldb_params(gbldb_t1, n_data
_columns, /
init_gbldb_t1 tms_library/fddb_lib.c /*void init_gbldb_t1(base_address, offsets, tl)/
init_gc_unit comm_prot/rmdc_comm/rmdc_comm_sub.c /*void init_gc_unit(unit_no, clea
r_dev_status)/
init_gcdb_data_col_list fddb/gcdb/gcdb_sub.c /*void init_gcdb_data_col_list(gcdb_t1, g
cdb_db_bas/
init_gcdb_params fddb/gcdb/gcdb_sub.c /*void init_gcdb_params(gcdb_t1, n_data_c
olumns, nu/
init_gcdb_t1 tms_library/fddb_lib.c /*void init_gcdb_t1(base_address, offsets, gcdb_t
1)/
init_group_table fddb/fddb_sub.c /*void init_group_table(tl, table_name)/
init_hex fddb/token/token.c /*init_hex(dest, src, table, curr_row, cnt)/
init_jhdriver comm_prot/tms_comm_sub.c /*void init_jhdriver()
init_library fddb/vmsdb/vmsdb_sub.c /*void init_library()
init_list_heads comm_prot/opc_comm/opc_comm_sub.c /*void init_list_heads()
init_message fddb/vmsdb/vmsdb_sub.c /*void init_message()
init_octal fddb/token/token.c /*init_octal(dest, src, table, curr_row, cnt)/
init_opc_event_flags comm_prot/opc_comm/opc_comm_sub.c /*unsigned long init_opc_
event_flags()
init_opc_unit comm_prot/opc_comm/opc_comm_sub.c /*void init_opc_unit(unit_no, cle
ar_dev_status)/
init_oprtvdb_data_col_list fddb/oprtvdb/oprtvdb_sub.c /*void init_oprtvdb_data_c
ol_list(tl, oprtvdb_db_bas/
init_oprtvdb_params fddb/oprtvdb/oprtvdb_sub.c /*void init_oprtvdb_params(oprtvd
b_t1, n_data_colum/
init_oprtvdb_t1 tms_library/fddb_lib.c /*void init_oprtvdb_t1(base_address, offsets, tl)
/
init_queue fddb/vmsdb/vmsdb_sub.c /*void init_queue()
init_rmdb_data_col_list fddb/rmdb/rmdb_sub.c /*void init_rmdb_data_col_list(tl, rmdb_db
_base)/
init_rmdb_params fddb/rmdb/rmdb_sub.c /*void init_rmdb_params(rmdb_t1, n_data_c
olumns, nu/
init_rmdb_t1 tms_library/fddb_lib.c /*void init_rmdb_t1(base_address, offsets, tl)/
init_rmdc_gc_unit comm_prot/rmdc_comm/rmdc_comm_sub.c /*void init_rmdc_gc_unit(
unit_no, clear_dev_status)/
init_rmdc_list_heads comm_prot/rmdc_comm/rmdc_comm_sub.c /*void init_rmdc_list_he
ads()
init_rmdc_unit comm_prot/rmdc_comm/rmdc_comm_sub.c /*void init_rmdc_unit(unit_no, cl
ear_dev_status)/
init_rtdb_t1 tms_library/rtdb_lib.c /*void init_rtdb_t1(base_address, offsets, tl)
/
init_rtrv_params comm_prot/opc_comm/opc_comm_sub.c /*void init_rtrv_params()
/
init_rx_jhub comm_prot/tms_comm_sub.c /*^struct jhub *init_rx_jhub(jh_ubfr, rx_s
ize, unit_/
init_scheddb_free_list fddb/scheddb/scheddb_sub.c /*void init_scheddb_free_list(sch

```

```

eddb_t1, base_addr/
init_scheddb_params fddb/scheddb/scheddb_sub.c /*void init_scheddb_params(schedd
b_t1, start_link_1/
init_scheddb_t1 tms_library/fddb_lib.c /*void init_scheddb_t1(base_address, offsets, sch
ed/
init_schedule fddb/scheddb/scheddb_sub.c /*void init_schedule()/
init_stn_aggr_list fddb/rmdb/rmdb_sub.c /*void init_stn_aggr_list(tl)
/
init_tx_jhub comm_prot/tms_comm_sub.c /*struct jhub *init_tx_jhub(jh_ubfr, tx_s
ize, unit_/
init_tx_rx_jhub comm_prot/tms_comm_sub.c /*struct jhub *init_tx_rx_jhub(jh_ubfr, t
x_size, rx/
init_vaxportdb_params fddb/vaxport/vaxportdb_sub.c /*void init_vaxportdb_params(tl, n
_VAXPorts, n_unit_/
init_vaxportdb_t1 tms_library/fddb_lib.c /*void init_vaxportdb_t1(base_address, off
sets, tl)-
init_video_tx_jhub comm_prot/tms_comm_sub.c /*struct jhub *init_video_tx_jhub
(jh_ubfr, tx_size)/
init_vms_list_heads comm_prot/vms_comm/vms_comm_sub.c /*void init_vms_list_head
s()/
init_vms_msg_struct tms_library/vms_lib.c /*void init_vms_msg_struct(cm)-
init_vms_tx_rx_jhub comm_prot/tms_comm_sub.c /*struct jhub *init_vms_tx_rx_jhu
b(jh_ubfr, tx_size/
init_vms_unit comm_prot/vms_comm/vms_comm_sub.c /*void init_vms_unit(unit_no, cle
ar_dev_status)-
init_vmsdb_data_col_list fddb/vmsdb/vmsdb_sub.c /*void init_vmsdb_data_col_list(t
l, vmsdb_db_base)-
init_vmsdb_free_list tms_library/vms_lib.c /*void init_vmsdb_free_list(vmsdb_t1, bas
e_address, /
init_vmsdb_params fddb/vmsdb/vmsdb_sub.c /*void init_vmsdb_params(vmsdb_t1, n_data
_columns, /
init_vmsdb_t1 tms_library/fddb_lib.c /*void init_vmsdb_t1(base_address, offsets, t1)-
initial_camera_setup comm_prot/opc_comm/cctv_comm_sub.c /*void initial_camera_set
up(oprtv_unit_no, monitor,/
ins_copy_new_st fddb/token/token.c /*ins_copy_new_st(dest, src, table, curr_row, cnt)
/
ins_new_state fddb/token/token.c /*ins_new_state(dest, src, table, curr_row, cnt)-
insert_in_scheddb_by_time tms_library/sched_lib.c /*int insert_in_scheddb_by_time(n
ew_entry)-
interpolate_metering_curve rt_skeleton/watch_bottleneck.c /*double interpolate_mete
ring_curve(rmdb_col_ptr, l/
joystick_monitor_to_camera comm_prot/opc_comm/cctv_comm_sub.c /*void joystick_m
onitor_to_camera(rx_ubfr)-
leading_zero_pad tms_library/format_db_lib.c /*void leading_zero_pad(str, leng
th)-
left_justify tms_library/misc_func.c /*void left_justify(field_bfr, width)-
link_rmdb_to_rtdb rt_skeleton/tms_startup.c /*void link_rmdb_to_rtdb()-
load_HM_Sregister comm_prot/reset_modem.c /*int load_HM_Sregister(channel, reg, val
ue)-
load_WRT_B_cmd comm_prot/opc_comm/cctv_comm_sub.c /*void load_WRT_B_cmd(tx_bfr, nex
t_char, wrt_addr, /
load_WRT_R_cmd comm_prot/opc_comm/cctv_comm_sub.c /*void load_WRT_R_cmd(tx_bfr, nex
t_char, wrt_addr, /
load_actv_anal_eqns rt_skeleton/actv_anal.c /*int load_actv_anal_eqns(eqn_file)-
load_camera_into_switch_cmd comm_prot/opc_comm/cctv_comm_sub.c /*void load_camer
a_into_switch_cmd(switch_cmd, next)-
load_delay_into_bfr comm_prot/opc_comm/cctv_comm_sub.c /*void load_delay_into_bf
r(tx_bfr, next_char, delay)-
load_dwell_into_switch_cmd comm_prot/opc_comm/cctv_comm_sub.c /*void load_dwell
_into_switch_cmd(switch_cmd, next)-
load_fmdb_name_table rt_skeleton/build_fmdb.c /*unsigned long load_fmdb_name_ta
ble(filename, num_-
load_monitor_into_switch_cmd comm_prot/opc_comm/cctv_comm_sub.c /*void load_monit
or_into_switch_cmd(switch_cmd, next)-
load_name_table_ndx fddb/fddb_sub.c /*void load_name_table_ndx(tl)-
load_param fddb/fddb_sub.c /*int load_param(tl, db_table_base)-

```

```

load_rtdb_name_table    rt_skeleton/build_rtdb.c           /*void load_rtdb_name_table(filename,
ame, num_entries, /
load_snapshot_namelist rt_skeleton/build_fmdb.c          /*unsigned long load_snapshot_nam
elist(namefile_name/
load_tables      fddb/vaxport/build_vaxportdb.c  /*void load_tables()
log_170_data_poll   comm_prot/rmdc_comm/rmdc_comm_sub.c /*void log_170_data_poll(
unit_no, lane_rate, lane_s/
log_170_error_response comm_prot/rmdc_comm/rmdc_comm_sub.c /*void log_170_error_resp
onse(tx_ubfr)/
log_24Hr_port_stats   comm_prot/tms_comm_sub.c        /*void log_24Hr_port_stats(port_n
o)/
log_24Hr_unit_stats   comm_prot/tms_comm_sub.c        /*void log_24Hr_unit_stats(unit_n
o)/
log_VMS_msg_in_hex    comm_prot/vms_comm/vms_comm_sub.c /*void log_VMS_msg_in_hex
(jh_ubfr, flag_bits)/
log_comm_bfr          comm_prot/tms_comm_sub.c        /*void log_comm_bfr(msg, jh_ubfr)/
log_comm_event         comm_prot/tms_comm_sub.c        /*void log_comm_event(event_code, mpu_no,
msg)/
log_comm_event_cc     comm_prot/tms_comm_sub.c        /*void log_comm_event_cc(event_co
de, mpu_no, msg, e/
log_comm_msg          comm_prot/tms_comm_sub.c        /*void log_comm_msg(event_code, mpu_no, m
sg)/
log_comm_msg_cc       comm_prot/tms_comm_sub.c        /*void log_comm_msg_cc(event_code, mpu_no
, msg, ext/
log_drcb              comm_prot/opc_comm/opc_comm_sub.c /*void log_drcb(msg, drcb_ptr, rx
_bfr)/
log_drcb_hdr          comm_prot/opc_comm/opc_comm_sub.c /*void log_drcb_hdr(msg, drcb_ptr
)*/
log_driver_stats      comm_prot/tms_comm_sub.c        /*void log_driver_stats(port_no)/
log_iobs_fields       comm_prot/tms_comm_sub.c        /*void log_iobs_fields(jh_ubfr)/
log_jhub_hdr          comm_prot/tms_comm_sub.c        /*void log_jhub_hdr(msg, jh_ubfr)/
log_list               comm_prot/tms_comm_sub.c        /*void log_list(list_head)/
log_list_rel          comm_prot/tms_comm_sub.c        /*void log_list_rel(list_head)/
log_name_list         comm_prot/opc_comm/opc_comm_sub.c /*void log_name_list(name_list, n
_names, name_size, /
log_tms_common         tms_library/event_log_sub.c   /*void log_tms_common(msg_type, event_cod
e, msg_text/
log_tms_event          tms_library/event_log_sub.c   /*void log_tms_event(event_code, msg_text
)*/
log_tms_event_cc      tms_library/event_log_sub.c   /*void log_tms_event_cc(event_cod
e, msg_text, cond_/
log_tms_mpu_name      tms_library/event_log_sub.c   /*void log_tms_mpu_name(msg_type,
event_code, mpu_n/
log_upi_event          upi_xmit/upi_xmit.c        /*void log_upi_event(event_code, msg_text)/
log_upi_event_cc      upi_xmit/upi_xmit.c        /*void log_upi_event_cc(event_code, msg_t
ext, cond_/
log_vms_buffer_stats   comm_prot/vms_comm/vms_comm_sub.c /*void log_vms_buffer_sta
ts()
loop_function          fddb/rmdb/rmdb_sub.c        /* unsigned char loop_function(loop_name)/
loop_titles            rt_skeleton/watch_fmdb.c      /*loop_titles(window)
mail_cmd_to_rmdc_comm  comm_prot/opc_comm/opc_comm_sub.c /*unsigned short mail_cmd
_to_rmdc_comm(rmdc_comm_mb/
mail_list_to_rmdc_comm comm_prot/opc_comm/opc_comm_sub.c /*unsigned short mail_lis
t_to_rmdc_comm(rmdc_comm_m/
mail_msg_to_vms_comm   comm_prot/opc_comm/opc_vms_sub.c /*unsigned char mail_msg_
to_vms_comm(action_code, v/
mail_patch_cmd_to_rmdc_comm fddb/rmdb/patch_rmdb.c /*unsigned short mail_patch_cmd_t
o_rmdc_comm(command/
mail_rmdc_updates      comm_prot/opc_comm/opc_comm_sub.c /*unsigned short mail_rmd
c_updates(opc_unit_no)/
manual_menu            comm_prot/rmdc_comm/rmdc_comm_sub.c /*int manual_menu()/
map_to_ACTVDB          rt_skeleton/actv_lib.c      /*unsigned long map_to_ACTVDB()
map_to_CCTVDB          tms_library/fddb_lib.c     /*unsigned long map_to_CCTVDB()
map_to_FMDB            tms_library/fmdb_lib.c     /*unsigned long map_to_FMDB()
map_to_GBLDB           tms_library/fddb_lib.c     /*unsigned long map_to_GBLDB()
map_to_GCDB            tms_library/fddb_lib.c     /*unsigned long map_to_GCDB()

```

```

map_to_OPRTVDB tms_library/fddb_lib.c /*unsigned long map_to_OPRTVDB()
map_to_RMDB tms_library/fddb_lib.c /*unsigned long map_to_RMDB()
map_to_RTDB tms_library/rtdb_lib.c /*unsigned long map_to_RTDB()
map_to_SCHEDDB tms_library/fddb_lib.c /*unsigned long map_to_SCHEDDB()
map_to_VAXPORTDB tms_library/fddb_lib.c /*unsigned long map_to_VAXPORTDB()
map_to_VMSDB tms_library/fddb_lib.c /*unsigned long map_to_VMSDB()
map_to_global_section tms_library/global_sub.c /*unsigned long map_to_global_section(name, write_a/
match_speed_loops fddb/rmdb/rmdb_sub.c /*void match_speed_loops(tl)
/
mem_sort_rtfmdb_names fddb/rmdb/rmdb_sub.c /*void mem_sort_rtfmdb_names(srt_file)
memcmpi tms_library/misc_func.c /*int memcmpi(buffer1, buffer2, count)
menu2 fddb/cctvdb/patch_cctvdb.c /*void menu2(tl)
menu3 fddb/cctvdb/patch_cctvdb.c /*void menu3(tl)
menu4 fddb/cctvdb/patch_cctvdb.c /*void menu4(tl)
mon_el_write_func rt_skeleton/mon_event_log.c /*void mon_el_write_func(buffer,
n_char)
move_fmdb_data_to_buffer rt_skeleton/fmdb_archiver.c /*int move_fmdb_data_to_b
uffer(df_bfr, nt_index, fi)
mr_titles rt_skeleton/watch_rmdc.c /*mr_titles()
msleep tms_library/misc_func.c /*void msleep(msleep)
multi_menu comm_prot/opc_comm/opc_comm_sub.c /*int multi_menu()
mvwrtstr rt_skeleton/watch_actv_anal.c /*mvwrtstr(window, row, col, string)
mvwrtstr_attrib rt_skeleton/watch_actv_anal.c /*mvwrtstr_attrib(window, row, col, string
, attribut/
mvwrtstr_underline rt_skeleton/watch_fmdb.c /*mvwrtstr_underline(window, row,
col, string)
mvwrtstr_vertical rt_skeleton/watch_bottleneck.c /*mvwrtstr_vertical(window, start_
row, col, string)
name_time_titles rt_skeleton/watch_fmdb.c /*name_time_titles(cabinet_name)
nd_compare fddb/vaxport/build_vaxportdb.c /*int nd_compare(entry1, entry2)
new_column fddb/fddb_sub.c /*void new_column(tl)
new_column_special_case fddb/cctvdb/cctvdb_sub.c /*void new_column_special_case(tl,
column_name)
nt_compare fddb/vaxport/build_vaxportdb.c /*int nt_compare(entry1, entry2)
nul_to_dest fddb/token/token.c /*nul_to_dest(dest, src, table, curr_row, cnt)
one_bit_mask tms_library/skel_sub.c /*unsigned long one_bit_mask(bit_number)
output_cctvdb_out_fil fddb/cctvdb/cctvdb_sub.c /*void output_cctvdb_out_fil(tl,
check_default)
output_gbl_db_out_fil fddb/gbldb/gbldb_sub.c /*void output_gbldb_out_fil(tl, check_def
ault)
output_gcdb_out_fil fddb/gcdb/gcdb_sub.c /*void output_gcdb_out_fil(gcdb_t1, check
_default)
output_oprvdb_out_fil fddb/oprvdb/oprvdb_sub.c /*void output_oprvdb_out_fil(tl,
check_default)
output_rmdb_out_fil fddb/rmdb/rmdb_sub.c /*void output_rmdb_out_fil(tl, check_defa
ult)
output_scheddb_out_fil fddb/scheddb/scheddb_sub.c /*void output_scheddb_out_fil()
output_vmsdb_out_fil fddb/vmsdb/vmsdb_sub.c /*void output_vmsdb_out_fil(vmsdb_t1, che
ck_default)
pack_fmdb_loop tms_library/pack_lib.c /*void pack_fmdb_loop(fmdb_loop, vol, occ, flag,
n/
pack_fmdb_spd_trap tms_library/pack_lib.c /*void pack_fmdb_spd_trap/
pack_fmdb_station tms_library/pack_lib.c /*void pack_fmdb_station(fmdb_stn, vol, o
cc, flag, /
pack_fp_checksum comm_prot/vms_comm/vms_comm_sub.c /*void pack_fp_checksum(f
letcher_checksum, fp_check/
pack_hi_lo tms_library/pack_lib.c /*void pack_hi_lo(param, buffer, lo_byte, hi_byte,
h)
pack_rtdb_inc_det tms_library/pack_lib.c /*void pack_rtdb_inc_det(rtdb_loop_stn, i
nc_det)
pack_rtdb_loop tms_library/pack_lib.c /*void pack_rtdb_loop(rtdb_loop, vol, scan_cnt, f
la/
pack_rtdb_spd_trap tms_library/pack_lib.c /*void pack_rtdb_spd_trap/
pack_rtdb_station tms_library/pack_lib.c /*void pack_rtdb_station(rtdb_stn, vol, s
can_cnt, f/

```

```
pack_volocc_summation tms_library/pack_lib.c /*void pack_volocc_summation(packed_summa-
tion, n_ti/
pad_FP_number tms_library/vms_lib.c /*void pad_FP_number(number)/
pad_end tms_library/tap_sub.c /*void pad_end(buffer, pad_length)/
pad_trailing_blanks tms_library/utility_func.c /*void pad_trailing_blanks(buffer,
, padded_length)/
pelco_init comm_prot/opc_comm/cctv_comm_sub.c /*void pelco_init(unit_no, joysti-
ck_mode, drop_addr/
perm_mask_to_byte fddb/fddb_sub.c /*void perm_mask_to_byte(perm_mask, perm_chars)/
port_control fddb/vaxport/patch_vaxportdb.c /*void port_control(vaxportdb_t1, port_no
)*/
print_Unit_list fddb/vaxport/vaxportdb_sub.c /*int print_Unit_list(vaxportdb_t1)/
print_VAXPortStatus fddb/vaxport/vaxportdb_sub.c /*void print_VAXPortStatus(vaxport
db_t1)/
print_VAXPort_list fddb/vaxport/vaxportdb_sub.c /*int print_VAXPort_list(vaxportdb
_t1)/
print_cabinet_list rt_skeleton/watch_bottleneck.c /*print_cabinet_list()
print_camera_report fddb/cctvdb/patch_cctvdb.c /*void print_camera_report()
print_ccbs tms_library/ccb_subs.c /*void print_ccbs(first_channel, number)
print_data_col_list rt_skeleton/watch_fmdb.c /*void print_data_col_list(tl)
print_db_table fddb/fddb_sub.c /*void print_db_table(tl, fddb_file, column_index, /
print_db_table_special_case fddb/cctvdb/cctvdb_sub.c /*void print_db_table_spe
cial_case (tl, fddb_input/
print_driver_stats rt_skeleton/comm_stats_rpt.c /*void print_driver_stats(out_fil
e)
print_element_name_list fddb/cctvdb/patch_cctvdb.c /*void print_element_name_list(tl,
first_element, la/
print_eqn_list rt_skeleton/watch_actv_anal.c /*int print_eqn_list()
print_file_comments fddb/fddb_sub.c /*int print_file_comments/
print_gc_report fddb/gcdb/patch_gcdb.c /*void print_gc_report()
print_group_name_list fddb/cctvdb/patch_cctvdb.c /*void print_group_name_list(tl)
print_hex_ascii_line tms_library/dump_mem.c /*void print_hex_ascii_line(start, count,
addr_offset/
print_n_skipped tms_library/dump_mem.c /*void print_n_skipped(n_zero, addr_offset)
print_one_ccb tms_library/ccb_subs.c /*void print_one_ccb(channel)
print_opr_report fddb/oprtvdb/patch_oprvdb.c /*void print_opr_report()
print_port_24hr_stats rt_skeleton/comm_stats_rpt.c /*void print_port_24hr_stats(out_
file)
print_port_list comm_prot/tms_comm_sub.c /*int print_port_list(msg, first_VAXPort,
last_VAXP/
print_reload_progress rt_skeleton/build_fmdb.c /*void print_reload_progress(sn_f
ilename, tag)
print_rmrdc_report fddb/rmdb/patch_rmdb.c /*void print_rmrdc_report()
print_scheddb_entry tms_library/sched_lib.c /*void print_scheddb_entry(out_file, sche
d_entry, a/
print_scheddb_entry_by_index fddb/scheddb/patch_scheddb.c /*void print_scheddb_entr
y_by_index(index, option)
print_scheddb_list fddb/scheddb/patch_scheddb.c /*void print_scheddb_list(option)
/
print_unit_24hr_stats rt_skeleton/comm_stats_rpt.c /*void print_unit_24hr_stats(out_
file, UnitListDB)
print_unit_list comm_prot/tms_comm_sub.c /*int print_unit_list(msg, first_unit, la
st_unit)
print_vms_auxout tms_library/vms_lib.c /*void print_vms_auxout(out_file, auxout)
/
print_vms_justify tms_library/vms_lib.c /*void print_vms_justify(out_file, justif
y, justify)
print_vms_line tms_library/vms_lib.c /*void print_vms_line(out_file, phase, line, line
_t/
print_vms_msg_type fddb/vmsdb/vmsdb_sub.c /*void print_vms_msg_type(text, msg_type)
/
print_vms_repeats tms_library/vms_lib.c /*void print_vms_repeats(out_file, label,
repeat_te/
print_vms_struct_field fddb/vmsdb/vmsdb_sub.c /*void print_vms_struct_field(label, fiel
d, max_len/
print_vms_time tms_library/vms_lib.c /*void print_vms_time(out_file, label, time_text)
```

```

/
print_vmsdb_cluster_entry      tms_library/vms_lib.c    /*void print_vmsdb_cluster_entry(
out_file, vmsdb_en/
print_vmsdb_entry_by_index     fddb/vmsdb/patch_vmsdb.c   /*void print_vmsdb_entry_
by_index(list_head, index,/
print_vmsdb_library_entry     tms_library/vms_lib.c    /*void print_vmsdb_library_entry(
out_file, vmsdb_en/
print_vmsdb_list_names        fddb/vmsdb/patch_vmsdb.c   /*void print_vmsdb_list_names(lis
t_head, mlq_option/
print_vmsdb_message_entry     tms_library/vms_lib.c    /*void print_vmsdb_message_entry(
out_file, vmsdb_en/
print_vmsdb_queue_entry       tms_library/vms_lib.c    /*void print_vmsdb_queue_entry(out_file,
vmsdb_entr/
print_vmsdb_report            fddb/vmsdb/patch_vmsdb.c   /*void print_vmsdb_report()
proc_term                     fddb/token/token.c    /*proc_term(dest, src, table, curr_row, cnt)/
process_ENQ_config            comm_prot/vms_comm/vms_comm_sub.c  /*void process_ENQ_config
(tx_ubfr)/
process_ENQ_status            comm_prot/vms_comm/vms_comm_sub.c  /*void process_ENQ_status
(tx_ubfr)/
process_RLI                   comm_prot/opc_comm/opc_comm_sub.c  /*void process_RLI(rx_ubfr)/
process_SCH_message           comm_prot/opc_comm/opc_vms_sub.c  /*void process_SCH_message
e(rx_ubfr)/
process_VMS_message           comm_prot/opc_comm/opc_vms_sub.c  /*void process_VMS_message
e(rx_ubfr)/
process_VMS_print              comm_prot/opc_comm/opc_vms_sub.c  /*unsigned char process_V
MS_print(sign_name, vmsdb_/
process_VMS_sign_cmd          comm_prot/opc_comm/opc_vms_sub.c  /*unsigned char process_V
MS_sign_cmd(action_code, v/
process_cluster                fddb/vmsdb/vmsdb_sub.c   /*void process_cluster(vmsdb_t1)/
process_data_response          comm_prot/rmdc_comm/rmdc_comm_sub.c /*void process_data_respo
nse(tx_ubfr)/
process_data_retrieval         comm_prot/opc_comm/opc_comm_sub.c  /*void process_data_retri
eval(rx_ubfr)/
process_enable_disable_loop    comm_prot/opc_comm/opc_comm_sub.c  /*void process_en
able_disable_loop(rx_ubfr)/
process_equation_file          rt_skeleton/watch_bottleneck.c /*process_equation_file(cab_index)
/
process_gc_err_resp            comm_prot/rmdc_comm/rmdc_comm_sub.c /*void process_gc_err_res
p(tx_ubfr)/
process_gc_status              comm_prot/rmdc_comm/rmdc_comm_sub.c /*void process_gc_status(
tx_ubfr)/
process_get_data               comm_prot/opc_comm/opc_comm_sub.c  /*void process_get_data(r
x_ubfr)/
process_idle_data              comm_prot/opc_comm/opc_comm_sub.c  /*void process_idle_data(
rx_ubfr)/
process_input_special_case     fddb/cctvdb/cctvdb_sub.c   /*process_input_special_ca
se(t1)/
process_invalid_partial        comm_prot/opc_comm/opc_comm_sub.c  /*void process_invalid_pa
rtial(rx_ubfr)/
process_library                 fddb/vmsdb/vmsdb_sub.c   /*void process_library(vmsdb_t1)/
process_mailbox_command        comm_prot/rmdc_comm/rmdc_comm_sub.c /*void process_mailbox_co
mmand(mbx_command, unit_no/
process_message                fddb/vmsdb/vmsdb_sub.c   /*void process_message(vmsdb_t1)/
process_meter_control          comm_prot/opc_comm/opc_comm_sub.c  /*void process_meter_cont
rol(rx_ubfr)/
process_monitor_request        comm_prot/opc_comm/cctv_comm_sub.c /*void process_monitor_re
quest(rx_ubfr)/
process_opc_good_comm           comm_prot/opc_comm/opc_comm_sub.c  /*void process_opc_good_c
omm(jh_ubfr)/
process_opc_qio_fdt_errors     comm_prot/opc_comm/opc_comm_sub.c  /*void process_op
c_qio_fdt_errors(msg, jh_ubfr, err/
process_opc_rx_available_errors comm_prot/opc_comm/opc_comm_sub.c  /*void process_op
c_rx_available_errors(msg, jh_ubfr/
process_opc_tx_errors          comm_prot/opc_comm/opc_comm_sub.c  /*void process_opc_tx_err
ors(msg, jh_ubfr, err_code/
process_opc_tx_wait_rx_errors  comm_prot/opc_comm/opc_comm_sub.c  /*void process_op
c_tx_wait_rx_errors(msg, jh_ubfr, err_code/

```

```

c_tx_wait_rx_errors(msg, jh_ubfr, /
process_operator_log_entry      comm_prot/opc_comm/opc_comm_sub.c      /*void process_op
erator_log_entry(rx_ubfr)/
process_operator_loginout       comm_prot/opc_comm/opc_comm_sub.c      /*void process_op
erator_loginout(rx_ubfr)/
process_output_special_case    comm_prot/opc_comm/opc_comm_sub.c      /*int process_out
put_special_case(tl, type, size, p/
process_partial_completion     comm_prot/opc_comm/opc_comm_sub.c      /*struct jhub *pro
cess_partial_completion(rx_ubfr)/
process_queue      fddb/vmsdb/vmsdb_sub.c /*void process_queue(vmsdb_t1)/
process_read_write_update      comm_prot/opc_comm/opc_comm_sub.c      /*void process_re
ad_write_update(rx_ubfr)/
process_request_for_data       comm_prot/opc_comm/opc_comm_sub.c      /*void process_re
quest_for_data(rx_ubfr)/
process_restart_repeat         comm_prot/opc_comm/opc_comm_sub.c      /*void process_restart_re
peat (rx_ubfr)/
process_rmdc_err_resp          comm_prot/rmdc_comm/rmdc_comm_sub.c      /*void process_rmdc_err_r
esp(tx_ubfr)/
process_rmdc_good_comm         comm_prot/rmdc_comm/rmdc_comm_sub.c      /*void process_rmdc_good_
comm(jh_ubfr)/
process_rmdc_tx_wait_rx_errors comm_prot/rmdc_comm/rmdc_comm_sub.c      /*void process_rm
dc_tx_wait_rx_errors(msg, jh_ubfr, /
process_schedule      fddb/scheddb/scheddb_sub.c /*void process_schedule()
process_terminate_data        comm_prot/opc_comm/opc_comm_sub.c      /*void process_terminate_
data(rx_ubfr)/
process_vms_good_comm         comm_prot/vms_comm/vms_comm_sub.c      /*void process_vms_good_c
omm(jh_ubfr)/
process_vms_tx_wait_rx_errors comm_prot/vms_comm/vms_comm_sub.c      /*void process_vm
s_tx_wait_rx_errors(msg, jh_ubfr, /
process_wild_card_name        comm_prot/opc_comm/opc_comm_sub.c      /*int process_wild_card_n
ame(rx_ubfr, ques_mark, tx/
prompt_for_unit_enable       comm_prot/opc_comm/opc_comm_sub.c      /*unsigned char prompt_fo
r_unit_enable(unit_no, dis/
prompt_for_yes_no            tms_library/kb_func.c /*int prompt_for_yes_no(msg1, msg2, def)/
queue_read_upi_mbx           upi_xmit/upi_xmit.c /*unsigned long queue_read_upi_mbx()
queued_get_1_char            tms_library/kb_func.c /*unsigned long queued_get_1_char(ttchan,
efn, ch, /
queued_read_from_mailbox     tms_library/mailbox.c /*unsigned long queued_read_from_
mailbox(channel, e/
range_check      fddb/fddb_sub.c /*int range_check(tl, param, offset, type, size, sr/
read_HM_Sregister           upi_xmit/upi_xmit.c /*int read_HM_Sregister(channel, reg)/
read_JH_modem_status         comm_prot/tms_comm_sub.c /*unsigned long read_JH_modem_sta
tus(channel, modem/
read_TT_modem_status         tms_library/tt_func.c /*unsigned long read_TT_modem_status(chan
nel, modem/
read_data_file   rt_skeleton/dumydata.c /*read_data_file(stream, buffer, size)/
read_df_elem_data           tms_library/fmdb_lib.c /*unsigned long read_df_elem_data(elem_na
me, channel/
read_df_hdr_namelist         tms_library/fmdb_lib.c /*unsigned long read_df_hdr_namelist(file
name, path/
read_fddb_file   fddb/fddb_sub.c /*int read_fddb_file(tl, fddb_file, fst)/
read_fmdb_namefile          tms_library/fmdb_lib.c /*unsigned long read_fmdb_namefile(filena
me, pathna/
read_fmdb_snapshot          tms_library/fmdb_lib.c /*unsigned long read_fmdb_snapshot(filena
me, pathna/
read_from_mailbox           tms_library/mailbox.c /*unsigned long read_from_mailbox(channel
, buffer, /
read_from_mailbox_nowait    tms_library/mailbox.c /*unsigned long read_from_mailbox
_nowait(channel, b/
read_report_list             noaa_monitor/noaa_monitor.c /*void read_report_list()
read_rtfdmdb_name_file      tms_library/fddb_lib.c /*int read_rtfdmdb_name_file(stream, curr_
record, le/
read_type_ahead tms_library/kb_func.c /*unsigned long read_type_ahead(ttchan, buffer, b
uf/
remove_from_list             tms_library/link_sub.c /*void *remove_from_list(entry)/
remove_from_list_head_i      tms_library/intlk_queue.c /*void *remove_from_list_head_i(1

```

```

ist_head) /
remove_from_list_tail_i tms_library/intlk_queue.c      /*void *remove_from_list_tail_i(l
ist_head) /
replace_cluster comm_prot/opc_comm/opc_vms_sub.c      /*unsigned short replace_cluster(
cst_name, cst_body/
replace_library comm_prot/opc_comm/opc_vms_sub.c     /*unsigned short replace_library(
sign_name, lib_nam/
replace_message comm_prot/opc_comm/opc_vms_sub.c     /*unsigned short replace_message(
sign_name, msg_nam/
replace_queue   comm_prot/opc_comm/opc_vms_sub.c     /*unsigned short replace_queue(si
gn_name, que_name,/
replace_schedule    comm_prot/opc_comm/opc_vms_sub.c  /*unsigned short replace_
schedule(sched_ptr, unit_n/
reschedule_tms_event  comm_prot/opc_comm/opc_vms_sub.c /*void reschedule_tms_eve
nt()/
restore_comm_process_name  comm_prot/tms_comm_sub.c  /*void restore_comm_proce
ss_name()/
restore_tty_process_name  tms_library/proc_cntrl.c  /*void restore_tty_proces
s_name(old_process_name)/
return_HM_to_cmd_mode  comm_prot/reset_modem.c /*int return_HM_to_cmd_mode(channel) /
return_scheddb_block  tms_library/sched_lib.c /*struct rel_list_hdr *return_scheddb_blo
ck(scheddb/
return_to_free_list  comm_prot/opc_comm/opc_comm_sub.c /*void return_to_free_lis
t(jh_ubfr) /
return_vmsdb_block  tms_library/vms_lib.c  /*struct rel_list_hdr *return_vmsdb_block
(vmsdb_tl,/
right_justify  tms_library/misc_func.c /*void right_justify(field_bfr, width) /
rtfm_compare  fddb/rmdb/rmdb_sub.c  /*int rtfm_compare(name1, name2) /
run_polling_processes  rt_skeleton/rt_skeleton.c  /*void run_polling_processes() /
run_process_alt_bit  rt_skeleton/rt_skeleton.c  /*void run_process_alt_bit(proc_n
ame, efc_name, sta/
run_process_one_bit  rt_skeleton/rt_skeleton.c  /*void run_process_one_bit(proc_n
ame, efc_name, sta/
run_process_wait  rt_skeleton/rt_skeleton.c  /*void run_process_wait(proc_name
, start_event_flag/
run_watch_actv_anal  rt_skeleton/actv_anal.c /*void run_watch_actv_anal(efc_name, star
t_event_f1/
rx_ast_func  comm_prot/opc_comm/opc_comm_sub.c  /*void rx_ast_func(param) /
scan_rtfmdb_name_file  fddb/rmdb/rmdb_sub.c  /*int scan_rtfmdb_name_file(filename, n_na
mes, n_loo/
scheddb_error  fddb/scheddb/scheddb_sub.c  /*void scheddb_error(err_msg, print_line,
first_col/
scroll_fmdb_col_offsets tms_library/fmdb_lib.c /*void scroll_fmdb_col_offsets() /
scroll_port_stats  comm_prot/tms_comm_sub.c  /*void scroll_port_stats(port_no)
/
scroll_rtldb_col_offsets tms_library/rtldb_lib.c /*void scroll_rtldb_col_offsets() /
scroll_unit_stats  comm_prot/tms_comm_sub.c  /*void scroll_unit_stats(unit_no)
/
search_all_cst_for_msg  tms_library/vms_lib.c /*unsigned short search_all_cst_for_msg(s
ign_name, /
search_all_lib_for_msg  tms_library/vms_lib.c /*unsigned short search_all_lib_for_msg(m
sg_name, v/
search_all_que_for_msg  tms_library/vms_lib.c /*unsigned short search_all_que_for_msg(m
sg_name, v/
search_fddb_offset_list tms_library/fddb_lib.c /*int search_fddb_offset_list(tl, offset)
/
search_fmdb_name_table  tms_library/fmdb_lib.c /*int search_fmdb_name_table(search_name)
/
search_rtldb_name_table tms_library/rtldb_lib.c /*int search_rtldb_name_table(search_name)
/
search_rtldb_offset_list tms_library/rtldb_lib.c /*int search_rtldb_offset_list(offset) /
search_sch_for_create_time  tms_library/sched_lib.c /*void *search_sch_for_create_tim
e(create_time) /
search_sch_for_dev_item tms_library/sched_lib.c /*void *search_sch_for_dev_item(device_na
me, action/
search_sch_for_start_time  tms_library/sched_lib.c /*void *search_sch_for_start_time

```

```

(device_name, acti/
select_DB_for_UnitList fddb/vaxport/patch_vaxportdb.c /*unsigned char select_DB_for_UnitList()
select_entry tms_library/kb_func.c /*int select_entry(min, max)
select_vmsdb_ll_entry comm_prot/vms_comm/vms_comm_sub.c /*void *select_vmsdb_ll_entry(list_head)
send_HM_cmd comm_prot/reset_modem.c /*int send_HM_cmd(channel, cmd_str)
send_HM_data comm_prot/reset_modem.c /*int send_HM_data(channel, data_bfr, n_bytes)
send_active_rfd_blocks comm_prot/opc_comm/opc_comm_sub.c /*void send_active_rfd_blocks(opc_unit_no)
send_data_response comm_prot/opc_comm/opc_comm_sub.c /*int send_data_response(drcb_ptr, flow_control_fla)
send_gc_start_msgs comm_prot/rmdc_comm/rmdc_comm_sub.c /*unsigned char send_gc_start_msgs(unit_no)
send_msg_to_oper rt_skeleton/actv_anal.c /*void send_msg_to_oper()
send_report upi_xmit/upi_xmit.c /*int send_report(channel, report, n_bytes)
send_rmdc_gc_start_msgs comm_prot/rmdc_comm/rmdc_comm_sub.c /*unsigned char send_rmdc_gc_start_msgs(unit_no)
send_rmdc_start_msgs comm_prot/rmdc_comm/rmdc_comm_sub.c /*unsigned char send_rmdc_start_msgs(unit_no)
send_to_mon_event_log rt_skeleton/event_logger.c /*void send_to_mon_event_log(msg_type, event_code, /)
send_vms_start_msgs comm_prot/vms_comm/vms_comm_sub.c /*unsigned char send_vms_start_msgs(unit_no)
set_2min_timer rt_skeleton/event_logger.c /*void set_2min_timer()
set_JH_speed_parity comm_prot/tms_comm_sub.c /*unsigned long set_JH_speed_parity(channel, speed, /)
set_TT_speed_parity tms_library/tt_func.c /*unsigned long set_TT_speed_parity(channel, speed, /)
set_joystick_status comm_prot/opc_comm/cctv_comm_sub.c /*void set_joystick_status(unit_no, new_joy_status)
set_port_partial tms_library/kb_func.c /*unsigned long set_port_partial(ttchan)
set_port_raw tms_library/kb_func.c /*unsigned long set_port_raw(ttchan)
set_priority tms_library/proc_cntrl.c /*unsigned long set_priority(new_priority, prev_pri)
set_process_name tms_library/proc_cntrl.c /*unsigned long set_process_name(process_name)
set_roadway_type_bits fddb/rmdb/rmdb_sub.c /*void set_roadway_type_bits(tl)
set_term_char tms_library/kb_func.c /*unsigned long set_term_char(ttchan, basic_char, e)
setup_port_for_protocol comm_prot/tms_comm_sub.c /*unsigned long setup_port_for_protocol(port_no, pr)
simple_write_to_comm_log comm_prot/tms_comm_sub.c /*void simple_write_to_comm_log(record, n_bytes, err)
skip_char fddb/token/token.c /*skip_char(dest, src, table, curr_row, cnt)
skip_chk_cnt fddb/token/token.c /*skip_chk_cnt(dest, src, table, curr_row, cnt)
skip_new_st fddb/token/token.c /*skip_new_st(dest, src, table, curr_row, cnt)
skip_same_st fddb/token/token.c /*skip_same_st(dest, src, table, curr_row, cnt)
sort_list tms_library/sort_lib.c /*void sort_list(list, record_length, num_records
./
sort_list_by_db_type fddb/vaxport/build_vaxportdb.c /*void sort_list_by_db_type(list, record_length, num_records)
sort_rtfmdb_names fddb/rmdb/rmdb_sub.c /*unsigned long sort_rtfmdb_names(src_file 1, src_fil)
spawn_DCL_cmd_nowait tms_library/proc_cntrl.c /*unsigned long spawn_DCL_cmd_nowait(command)
spawn_noaa_del_cmd noaa_monitor/noaa_monitor.c /*void spawn_noaa_del_cmd()
squeeze_layout rt_skeleton/watch_bottleneck.c /*int squeeze_layout(line_at_bottom)
start_cctv_port comm_prot/opc_comm/cctv_comm_sub.c /*unsigned long start_cctv_port(port_no)
start_cctv_unit comm_prot/opc_comm/cctv_comm_sub.c /*void start_cctv_unit(unit_no, clear_dev_status)
start_comm_process rt_skeleton/tms_startup.c /*unsigned long start_comm_process(image_name, proc)
start_next_tms_sched comm_prot/opc_comm/opc_vms_sub.c /*void start_next_tms_sched(check_curr)

```

```

start_noaa_port noaa_monitor/noaa_monitor.c      /*void start_noaa_port()
start_noaa_timer      noaa_monitor/noaa_monitor.c      /*void start_noaa_timer()
start_opc_port  comm_prot/opc_comm/opc_comm_sub.c      /*unsigned long start_opc_port(po
rt_no, n_short_to_/
start_process   tms_library/proc_cntrl.c      /*unsigned long start_process(image_name,
process_n/
start_processx  rt_skeleton/tms_startup.c      /*unsigned long start_processx(image_name
, process_/
start_reodial_timer    upi_xmit/upi_xmit.c      /*int start_reodial_timer(redial_wait_time
)*/
start_rmdc_port comm_prot/rmdc_comm/rmdc_comm_sub.c      /*unsigned long start_rmdc_port(po
rt_no)/
start_rpt_timer rt_skeleton/comm_stats_rpt.c      /*void start_rpt_timer()
start_tms_process     rt_skeleton/tms_startup.c      /*void start_tms_process(filename
, tms_control_mbx,/
start_upi_port upi_xmit/upi_xmit.c      /*void start_upi_port()
start_vms_port  comm_prot/vms_comm/vms_comm_sub.c      /*unsigned long start_vms_port(po
rt_no)/
start_vs_port   comm_prot/opc_comm/cctv_comm_sub.c      /*unsigned long start_vs_port()
stn_titles      rt_skeleton/watch_fmdb.c      /*stn_titles()
stop_cctv_port  comm_prot/opc_comm/cctv_comm_sub.c      /*unsigned long stop_cctv_port(po
rt_no)/
stop_cctv_unit  comm_prot/opc_comm/cctv_comm_sub.c      /*void stop_cctv_unit(unit_no)/
stop_gc_unit    comm_prot/rmdc_comm/rmdc_comm_sub.c      /*void stop_gc_unit(unit_no)/
stop_noaa_port  noaa_monitor/noaa_monitor.c      /*void stop_noaa_port()
stop_opc_port   comm_prot/opc_comm/opc_comm_sub.c      /*unsigned long stop_opc_port(po
rt_no)/
stop_opc_unit   comm_prot/opc_comm/opc_comm_sub.c      /*void stop_opc_unit(unit_no)/
stop_rmdc_gc_unit comm_prot/rmdc_comm/rmdc_comm_sub.c      /*void stop_rmdc_gc_unit(
unit_no)/
stop_rmdc_port  comm_prot/rmdc_comm/rmdc_comm_sub.c      /*unsigned long stop_rmdc_port(po
rt_no)/
stop_rmdc_unit  comm_prot/rmdc_comm/rmdc_comm_sub.c      /*void stop_rmdc_unit(unit_no)/
stop_upi_port   upi_xmit/upi_xmit.c      /*void stop_upi_port()
stop_vms_port   comm_prot/vms_comm/vms_comm_sub.c      /*unsigned long stop_vms_port(po
rt_no)/
stop_vms_unit   comm_prot/vms_comm/vms_comm_sub.c      /*void stop_vms_unit(unit_no)/
stop_vs_port    comm_prot/opc_comm/cctv_comm_sub.c      /*unsigned long stop_vs_port()
strcmpi tms_library/misc_func.c /*int strcmpi(string1, string2)/
strip_leading_blanks  tms_library/format_db_lib.c      /*void strip_leading_blanks(ptr)/
strip_trailing_blanks tms_library/format_db_lib.c      /*void strip_trailing_blanks(ptr)
/
switch_monitor_to_camera  comm_prot/opc_comm/cctv_comm_sub.c      /*void switch_mon
itor_to_camera(rx_ubfr)/
switch_monitor_to_sequence  comm_prot/opc_comm/cctv_comm_sub.c      /*void switch_mon
itor_to_sequence(rx_ubfr)/
switch_port      comm_prot/switch_tty.c /*switch_port(port_name, driver_location, switch_a
ll/
talk_through    comm_prot/reset_modem.c /*int talk_through(channel)/
terminate_all    comm_prot/opc_comm/opc_comm_sub.c      /*int terminate_all(unit_no)/
terminate_one    comm_prot/opc_comm/opc_comm_sub.c      /*int terminate_one(unit_no, hand
le)/
test_menu       comm_prot/opc_comm/opc_comm_sub.c      /*int test_menu()
test_unit_select  comm_prot/rmdc_comm/rmdc_comm_sub.c      /*int test_unit_select()
time_stamp_rtdb rt_skeleton/rt_skeleton.c      /*unsigned long time_stamp_rtdb(curr_time
)*/
timed_wait_for_1_char  tms_library/kb_func.c      /*unsigned long timed_wait_for_1_char(ttc
han, timeo/
tnl_compare     fddb/vaxport/build_vaxportdb.c /*int tnl_compare(entry1, entry2)/
trans_chk_cnt   fddb/token/token.c      /*^trans_chk_cnt(dest, src, table, curr_row, cnt)/
translate_char   fddb/token/token.c      /*^translate_char(dest, src, table, curr_row, cnt)/
translate_logical_name tms_library/logical_name.c      /*unsigned long translate_logical
_name(logical_name)
trap_titles      rt_skeleton/watch_fmdb.c      /*trap_titles()
trim_trailing_blanks tms_library/utility_func.c /*void trim_trailing_blanks(buffer
r)/

```

```

turn_off_JH_DTR comm_prot/tms_comm_sub.c      /*^unsigned long turn_off_JH_DTR(channel)//
turn_off_JH_RTS comm_prot/tms_comm_sub.c      /*^unsigned long turn_off_JH_RTS(channel)//
turn_off_TT_DTR tms_library/tt_func.c        /*^unsigned long turn_off_TT_DTR(channel)//
turn_off_TT_RTS tms_library/tt_func.c        /*^unsigned long turn_off_TT_RTS(channel)//
turn_on_JH_DTR comm_prot/tms_comm_sub.c      /*^unsigned long turn_on_JH_DTR(channel)//
turn_on_JH_RTS comm_prot/tms_comm_sub.c      /*^unsigned long turn_on_JH_RTS(channel)//
turn_on_TT_DTR tms_library/tt_func.c        /*^unsigned long turn_on_TT_DTR(channel)//
turn_on_TT_RTS tms_library/tt_func.c        /*^unsigned long turn_on_TT_RTS(channel)//
two_bit_mask   tms_library/skel_sub.c       /*^unsigned long two_bit_mask(bit_number)//
two_group_layout rt_skeleton/watch_bottleneck.c /*int two_group_layout()//
tx_ast_func    comm_prot/opc_comm/opc_comm_sub.c /*void tx_ast_func(param)//
tx_done_rx_ast_func comm_prot/rmdc_comm/rmdc_comm_sub.c /*void tx_done_rx_ast_fun
c(param)//
tx_wait_rx_ast_func comm_prot/opc_comm/opc_comm_sub.c /*void tx_wait_rx_ast_fun
c(param)//
ulong_to_table tms_library/table_sub.c /*void ulong_to_table(pointer, ulong)//
unit_control   fddb/vaxport/patch_vaxportdb.c /*void unit_control(vaxportdb_tl, unit_no
)//
unmap_global_section tms_library/global_sub.c /*unsigned long unmap_global_sect
ion(gbl_addr)//
unpack_170_speed_trap  comm_prot/rmdc_comm/rmdc_comm_sub.c /*void unpack_170_speed_t
rap//
unpack_170_vol_occ   comm_prot/rmdc_comm/rmdc_comm_sub.c /*void unpack_170_vol_occ
(volume, scan_cnt, vs_byte)//
unpack_fmdb_loop    tms_library/pack_lib.c /*void unpack_fmdb_loop(fmdb_loop, vol, o
cc, flag, ///
unpack_fmdb_spd_trap tms_library/pack_lib.c /*void unpack_fmdb_spd_trap//
unpack_fmdb_station tms_library/pack_lib.c /*void unpack_fmdb_station(fmdb_stn, vol,
occ, flag)//
unpack_rtbd_inc_det tms_library/pack_lib.c /*void unpack_rtbd_inc_det(rtbd_loop_stn,
inc_det)//
unpack_rtbd_loop    tms_library/pack_lib.c /*void unpack_rtbd_loop(rtbd_loop, vol, s
can_cnt, f)//
unpack_rtbd_loop_stn tms_library/pack_lib.c /*void unpack_rtbd_loop_stn(rtbd_loop_stn
, vol, sca)//
unpack_rtbd_spd_trap tms_library/pack_lib.c /*void unpack_rtbd_spd_trap//
unpack_rtbd_station tms_library/pack_lib.c /*void unpack_rtbd_station(rtbd_stn, vol,
scan_cnt)//
up_arrow         tms_library/fmdb_lib.c /*void up_arrow(first_col, num_cols)//
ushort_to_table tms_library/table_sub.c /*void ushort_to_table(pointer, ushort)//
vaxport_event_msg  fddb/vaxport/vaxportdb_sub.c /*void vaxport_event_msg(module_c
ode, msg)//
vaxport_event_msg_cc fddb/vaxport/vaxportdb_sub.c /*void vaxport_event_msg_cc(modul
e_code, msg, cond)//
vms_c_getch     tms_library/kb_func.c  /*int vms_c_getch()//
vms_c_grabch    tms_library/kb_func.c  /*int vms_c_grabch()//
vms_c_kbhit     tms_library/kb_func.c  /*int vms_c_kbhit()//
vms_msg_length  tms_library/vms_lib.c /*int vms_msg_length(vmsdb_msg)//
wait_HM_number  comm_prot/reset_modem.c /*int wait_HM_number(channel, buffer, length, tim
eo)//
wait_HM_result   comm_prot/reset_modem.c /*int wait_HM_result(channel, buffer, length, tim
eo)//
wait_for_1_char tms_library/kb_func.c  /*unsigned long wait_for_1_char(ttchan, ch, iosb)
/
wc_search_dev_name comm_prot/opc_comm/opc_comm_sub.c /*int wc_search_dev_name(
tl, dev_name, drcb_ptr, rx)//
wc_search_elem_name comm_prot/opc_comm/opc_comm_sub.c /*int wc_search_elem_name(
tl, elem_name, drcb_ptr, r)//
wc_search_loop_error comm_prot/opc_comm/opc_comm_sub.c /*int wc_search_loop_error(
tl, name_bfr, drcb_ptr, r)//
wc_search_loop_name  comm_prot/opc_comm/opc_comm_sub.c /*int wc_search_loop_name(
tl, name_bfr, drcb_ptr, rx)//
wc_search_mlq_name  comm_prot/opc_comm/opc_comm_sub.c /*int wc_search_mlq_name(
tl, name_bfr, drcb_ptr, rx)//
wc_search_param_tuning_list comm_prot/opc_comm/opc_comm_sub.c /*int wc_search_pa
ram_tuning_list(tl, name_bfr, drcb)

```

```
wc_search_rtdb_name      comm_prot/opc_comm/opc_comm_sub.c      /*int wc_search_rtdb_name(  
tl, rtdb_name, drcb_ptr, r/  
wc_search_scheddb      comm_prot/opc_comm/opc_comm_sub.c      /*int wc_search_scheddb(n-  
ame_bfr, drcb_ptr, rx_ubfr/  
white_space_break      tms_library/misc_func.c /*unsigned char *white_space_break(text_p-  
tr)/  
write_dft_stn_aggr_eqns_to_file fddb/rmdb/rmdb_sub.c    /*void write_dft_stn_aggr_eqns_to_  
file(tl)/  
write_eof_to_mailbox     tms_library/mailbox.c   /*unsigned long write_eof_to_mailbox(chan-  
nel, iosb)/  
write_fmdb_daily_files  rt_skeleton/fmdb_archiver.c  /*unsigned long write_fmdb_daily_  
files/  
write_fmdb_name_files   rt_skeleton/build_fmdb.c    /*unsigned long write_fmdb_name_f-  
iles(fmdb_pm, n_na/  
write_fmdb_snapshot_file rt_skeleton/fmdb_archiver.c  /*unsigned long write_fmd-  
b_snapshot_file(channel, f/  
write_loop_name_list_to_file fddb/rmdb/rmdb_sub.c    /*void write_loop_name_list_to_file(  
tl)/  
write_loop_table_to_file fddb/rmdb/rmdb_sub.c    /*void write_loop_table_to_file(t-  
l, table_index, gr/  
write_noaa_file noaa_monitor/noaa_monitor.c    /*void write_noaa_file(rpt_type_ndx, noaa-  
_time, sta/  
write_rec_to_op_log     comm_prot/opc_comm/opc_comm_sub.c  /*unsigned long write_rec-  
_to_op_log(buffer, first_c/  
write_speed_traps_to_file fddb/rmdb/rmdb_sub.c    /*void write_speed_traps_to_file(  
tl)/  
write_string_to_mailbox tms_library/mailbox.c  /*unsigned long write_string_to_mailbox(c-  
hannel, st/  
write_string_to_mailbox_nowait tms_library/mailbox.c  /*unsigned long write_string_to_m-  
ailbox_nowait(chan/  
write_to_crash_log     tms_library/proc_cntrl.c   /*unsigned long write_to_crash_lo-  
g(process_name, te/  
write_to_mailbox       tms_library/mailbox.c  /*unsigned long write_to_mailbox(channel,  
message, /  
write_to_mailbox_nowait tms_library/mailbox.c  /*unsigned long write_to_mailbox_nowait(c-  
hannel, me/  
write_to_operator_log   comm_prot/opc_comm/opc_comm_sub.c  /*unsigned long write_to_  
operator_log(msg, msg_size/  
write_to_sort_bfr      fddb/rmdb/rmdb_sub.c    /*void write_to_sort_bfr(name, type)/  
write_vmsdb_cst comm_prot/opc_comm/opc_vms_sub.c  /*void write_vmsdb_cst(out_file)/  
write_vmsdb_mlq  comm_prot/opc_comm/opc_vms_sub.c  /*void write_vmsdb_mlq(vmsdb_col-  
ptr, out_file)/
```

actv_anal.c

```
429 main [actv_anal.c]
430     map_to_RMDB [./tms_library/fddb_lib.c]
431         map_to_global_section [./tms_library/global_sub.c]
432             strlen
433                 MGBMSC
434             init_rmdb_t1 [./tms_library/fddb_lib.c]
435
436         STOP
437             map_to_GMDB [./tms_library/fddb_lib.c]
438                 map_to_global_section [see line 431]
439                     init_gldb_t1 [./tms_library/fddb_lib.c]
440                         map_to_RMDB [./tms_library/rxdb.lib.c]
441                             map_to_global_section [see line 431]
442                                 init_rxdb_t1 [./tms/library/rxdb.lib.c]
443                                     ASCEFC
444                                         two_bit_mask [./tms_library/skel_sub.c]
445                                             clear_all_event_flags [see line 356]
446                                                 fopen
447                                                     exit
448             count_actv_anal_eqns [actv_anal.c]
449                 fgets
450                     pad_end [./tms_library/tap_sub.c]
451                         strlen
452
453             map_to_global_section [see line 431]
454                 sprintf
455                     unmap_global_section [./tms_library/global_sub.c]
456                         DEFLTA
457                             delete_all_global_section [./tms_library/global_sub.c]
458
459             create_global_section [./tms_library/global_sub.c]
460                 strlen
461                     CRMPSC
462             init_actvdb_t1
463                 memcp
464             init_actvdb_params
465                 load_actv_anal_eqns [actv_anal.c]
466
467             fgets
468                 pad_end [see line 450]
469                 memcmp
470             printf
471                 find_fddb_c1_name [./tms_library/fddb_lib.c]
472                     memcmp
473                         classify_roadway [./tms_library/fddb_lib.c]
474                             memcmp
475             memcp
476             search_rtdb_name_table [./tms_library/rtdb.lib.c]
477                 memcmp
478                     init_actvdb_eqn [actv_anal.c]
479
480             fclose
481                 find_Fddb_c1_name [see line 471]
482                     WFLOR
483             READDEF
484                 calc_actv_anal [actv_anal.c]
485                     get_actv_params_from_RMDB
486                         printf
487                             unpack_rtdb_loop_stm [./tms_library/pack.lib.c]
488                                 check_valid_flag [actv_anal.c]
489                                     send_msg_to_oper [actv_anal.c]
490                                         strcpy
491                                         strlen
492                                         memcp
```

96/15/3
17:59:57

1

bottleneck.c

```

395 main [bottleneck.c]
396     general_process_startup [./tms_library/proc_ctrl.c]
397     connect_to_mailbox [./tms_library/mailbox.c]
398         strlen
399             ASSIGN
400                 strcpy
401                 printf
402                 write_to_crash_log [see line 21]
403                 STOP
404                 ASPECTFC
405                 clear_all_event_flags [./tms_library/proc_ctrl.c]
406                     CLREF
407                         printf
408                             STOP
409                                 sprintf
410                                     write_string_to_mailbox_nowait [./tms_library/mailbox.c]
411                                         strlen
412                                             QIOW
413                                                 log_tms_event [./tms_library/event_log_sub.c]
414                                                 log_tms_common [./tms_library/event_log_sub.c]
415                                                 strlen
416                                                 write_to_crash_log [see line 21]
417                                                 GETTIM
418                                                 memcpy
419                                                 memset
420                                                 write_to_mailbox_nowait [see line 393]
421                                         map_to_RTDDB [./tms_library/rtdb_lib.c]
422                                         map_to_global_section [./tms_library/global_sub.c]
423                                         strlen
424                                         MGBLSC
425                                         init_rtdb_t1 [./tms_library/rtdb_lib.c]
426                                         log_tms_event_cc [./tms_library/event_log_sub.c]
427                                         log_tms_common [see line 414]
428                                         PRINTF
429                                         STOP
430                                         map_to_RMDB [./tms_library/fddb_lib.c]
431                                         map_to_global_section [see line 422]
432                                         init_rmdb_t1 [./tms_library/fddb_lib.c]
433                                         build_bottleneck_table [bottleneck.c]
434                                         malloc
435                                         build_tap_error [./tms_library/tap_sub.c]
436                                         sprintf
437                                         memory
438                                         PRINTF
439                                         log_tms_event [see line 413]
440                                         exit
441                                         code_to_table [./tms_library/table_sub.c]
442                                         ushort_to_table [./tms_library/table_sub.c]
443                                         ulong_to_table [./tms_library/table_sub.c]
444                                         date_time_to_table [./tms_library/table_sub.c]
445                                         GETTIM
446                                         PRINTF
447                                         memory
448                                         fopen
449                                         fgets
450                                         pad_end [./tms_library/tap_sub.c]
451                                         strlen
452                                         memcmp
453                                         find_fddb_cl_name [./tms_library/fddb_lib.c]
454                                         memory
455                                         classify_roadaway [./tms_library/fddb_lib.c]
456                                         memory
457                                         check_reversible [./tms_library/tap_sub.c]
458                                         realloc

```

build_cctvdbcall

```
555 main [build_cctvdb.c]
556     printf
557     fopen
558     exit
559     time
560     ctime
561     fprintf
562     read_fddb_file [../../tms_library/fddb_lib.c]
563     get_next_line [../../fddb_sub.c]
564     ftelli
565     fgets
566     strlen
567     fseek
568     convert_non_print_to_space
569     strip_trailing_blanks [see line 141]
570     strip_leading_blanks [../../tms_library/format_db_lib.c]
571     setjmp
572     fddb_error [see line 129]
573     sprintf
574     exit
575     rewind
576     map_to_global_section [../../tms_library/global_sub.c]
577     strlen
578     MGBASIC
579     sprintf
580     fddb_error [see line 129]
581     STOP
582     unmap_global_section [../../tms_library/global_sub.c]
583     DELTIVA
584     delete_all_global_section [../../tms_library/global_sub.c]
585     DBALSC
586     create_global_section [../../tms_library/global_sub.c]
587     strlen
588     CRPSC
589     init_cctvdb_t1 [../../tms_library/fddb_lib.c]
590     memcpy
591     init_cctvdb_params [cctvdb_sub.c]
592     init_fddb_params [../../fddb_sub.c]
593     memset
594     GETTRIM
595     fddb_error [see line 129]
596     init_cctvdb_data_col_list [cctvdb_sub.c]
597     memset
598     calc_offsets [../../fddb_sub.c]
599     load_name_table_idx [../../fddb_sub.c]
600     find_fddb_nt_name [see line 479]
601     fclose

129     fddb_error [../../fddb_sub.c]
130     sprintf
131     strcpy
132     printf
133     fprintf
134     build_up_arrow [../../fddb_sub.c]
135     find_err_text [../../fddb_sub.c]
136     strlen
137     memcpy
138     _filbuf
139     strlen

141     strip_trailing_blanks [../../tms_library/format_db_lib.c]
142     strlen
```

build_fmdb.c

```
413 main [build_fmdb.c]
414     general_process_startup [...] /tms_library/proc_ctrl.c]
415         connect_to_mailbox [...] /tms_library/mailbox.c]
416             strlen
417                 ASSIGN
418                     strcpy
419                         printf
420                             write_to_crash_log [see line 21]
421                                 STOP
422                                     ASCEFC
423                                         clear_all_event_flags [...] /tms_library/proc_ctrl.c)
424                                             CLRNF
425                                                 printf
426                                                     STOP
427                                                         sprintf
428                                                             write_string_to_mailbox_nowait [...] /tms_library/mailbox.c]
429                                                                 strlen
430                                                                     QIOW
431             log_tms_event [see line 91]
432                 map_to_KRVA [...] /tms_library/rtdb.lib.c]
433                     map_to_global_section [...] /tms_library/global_sub.c]
434                         strlen
435                             MGRMSC
436                               init_rtdb_ttl [...] /tms_library/rtdb.lib.c]
437                                   printf
438                                       STOP
439                                           map_to_RMDB [...] /tms_library/fddb.lib.c]
440                                               map_to_global_section [see line 433]
441                                                   init_rmdb_ttl [...] /tms_library/fddb.lib.c]
442                                                       map_to_global_section [see line 433]
443                                                           log_tms_event_cc [...] /tms_library/event_log_sub.c]
444                                                               log_tms_common [see line 92]
445           exit
446             unmap_global_section [...] /tms_library/global_sub.c]
447               DEMITVA
448                 delete_all_global_section [...] /tms_library/global_sub.c]
449                     strlen
450                         DBLSC
451                           create_global_section [...] /tms_library/global_sub.c]
452                               strlen
453                                 CRMPSC
454                                   sprintf
455                                     memory
456                                       GETTIM
457                                         memset
458                                           load_fmdb_name_table [build_fmdb.c]
459                                             fopen
460                                               printf
461                                                 memset
462                                                   read_rtdb_name_file [...] /tms_library/rtdb.lib.c]
463                                                     fgets
464                                                       strlen
465                                                         memory
466                                                             strcpy
467                                                               search_rtdb_name_table [...] /tms_library/rtdb.lib.c]
468                     memory
469                         memcmp
470                           fclose
471                             exit
472                               write_fmdb_name_files [build_fmdb.c]
473                                 malloc
474                                   printf
475                                     exit
476                                         memset
477                                           GETTIM
478                                             NUMTIM
479                                               sprintf
480                                                 strcpy
481                                                   strlen
482                                                     strcat
483                                                       memcpy
484                                                         CREATE
485                                                           QIOW
486                                                             DASSCN
487                                                               NUMTIM
488                     CVT_VECTIM
489                         SUB_TIMES
490                           build_fmdb_snapshot_filename [...] /tms_library/fmdb.lib.c]
491                               read_fmdb_snapshot [...] /tms_library/fmdb.lib.c]
492                                 strlen
493                                   read_fmdb_snapshot [...] /tms_library/fmdb.lib.c]
494                                     strlen
495                                       QIOW
496                                         DASSCN
497                                           OPEN
498                                             printf
499                                               PUTMSG
500                                                 malloc
501                                                   QIOW
502                                                     DASSCN
503                                                       strchr
504                                                         memcmp
505                                                           dump_mem [see line 187]
506                                                             _filbuf
507                                                               check_filename [...] /tms_library/misc_func.c]
508                     strlen
509                         find_string [...] /tms_library/misc_func.c]
510                           strlen
511                             memcmp
512                               find_string [see line 509]
513                                 memory
514                                   get_number_from_header [...] /tms_library/misc_func.c]
515                     strlen
516                         find_string [see line 509]
517                           clear_R0 [...] /tms_library/misc_func.c]
518                             strlen
519                               strtonl
520                                 load_snapshot_name_list [build_fmdb.c]
521                                   strcmp
522                                     read_fmdb_namefile [...] /tms_library/fmdb.lib.c]
523                                         strcpy
524                                           strcat
525                                             OPEN
526                                               printf
527                                                 strlen
528                                                   malloc
529                                                     QIOW
530                                                       DASSCN
531                                                         strchr
532                                               memcmp
533                                                 dump_mem [see line 187]
534                                                   _filbuf
535                                                     check_filename [see line 507]
536                                                       get_number_from_header [see line 515]
537                                         free
538                                           malloc
539                                             printf
```

920544
154700

2

build_fndb.c

```
540     search_fndb_name_table [./tms_library/fndb.lib.c]
541     memcmp
542
543     _MOVQ3
544     print_reload_progress [build_fndb.c]
545     PrintF
546     write_string_to_mailbox [./tms_library/mailbox.c]
547     strlen
548     QIOW
549     write_to_crash_log [see line 21]
```

```
21     write_to_crash_log [./tms_library/proc_cntrl.c]
22
23     GETRTIM
24     fopen
25     fprintf
26     fclose
```

```
92     log_tms_common [./tms_library/event_log_sub.c]
93     strlen
94     write_to_crash_log [see line 21]
95     GETRTIM
96     memcopy
97     memset
98     write_to_mailbox_nowait [./tms_library/mailbox.c]
99     QIOW
```

```
187     dump_mem [./tms_library/dump_mem.c]
188     print_hex_ascii_line [./tms_library/dump_mem.c]
189     sprintf
190     printf
191     print_n_skipped [./tms_library/dump_mem.c]
192     printf
```

```
433     map_to_global_section [./tms_library/global_sub.c]
434     strlen
435     MGBUSC
```

```
507     check_filename [./tms_library/misc_func.c]
508     strlen
509     find_string [./tms_library/misc_func.c]
510     strlen
511     memcopy
```

```
515     get_number_from_header [./tms_library/misc_func.c]
516     strlen
517     find_string [see line 509]
518     clear_R0 [./tms_library/misc_func.c]
519     strtoul
```

16/07/18
9:07:58

built_gdb.c

```
540 main [build_gdb.c]
541     printf
542         fopen
543             exit
544                 time
545                     ctime
546                         fprintf
547                             read_fddb_file [./fddb_sub.c]
548                                 get_next_line [./fddb_sub.c]
549                                     ftell
550                                         fgets
551                                             strlen
552                                                 seek
553                                                     convert_non_print_to_space
554                                                         strip_trailing_blanks [see line 144]
555                                                             strip_leading_blanks [./tms_library/format_db_lib.c]
556
557                                         fddb_error [see line 132]
558                                         sprintf
559                                         exit
560                                         rewind
561                                         map_to_global_section [./tms_library/global_sub.c]
562                                             strlen
563                                                 MCBLSC
564                                         sprintf
565                                             fddb_error [see line 132]
566                                                 STOP
567                                                 ummap_global_section [./tms_library/global_sub.c]
568                                                     DMLVA
569                                         delete_all_global_section [./tms_library/global_sub.c]
570                                             strlen
571                                                 DBLSC
572                                         create_global_section [./tms_library/global_sub.c]
573                                             strlen
574                                                 CRMPSC
575                                         init_gblib_t1 [./tms_library/fddb.lib.c]
576                                         memcpy
577                                         init_gblib_params
578                                         init_gblib_data_col1list
579                                         calc_offsets [./fddb_sub.c]
580                                         load_name_table_idx [./fddb_sub.c]
581                                         find_fddb_nt_name [see line 461]
582                                         fclose
583
584
585                                         fddb_error [./fddb_sub.c]
586                                         sprintf
587                                             strcpy
588                                         printf
589                                         fprintf
590                                             build_up_arrow [./fddb_sub.c]
591                                         find_err_text [./fddb_sub.c]
592                                         sprintf
593                                             strlen
594                                         memcpy
595                                         memcp
596                                         -fillbuf
597                                         strcpy
598                                         strip_trailing_blanks [./tms_library/format_db_lib.c]
599                                         find_fddb_nt_name [./tms_library/fddb.lib.c]
600                                         memcp
```

```

550 main [build_gcdb.c]
551     printf
552         fopen
553             exit
554             time
555                 ctime
556                     fprintf
557                         read_fddb_file [../../tms_library/fddb_lib.c]
558                             get_next_line [../../fddb_sub.c]
559                                 fteil1
560                                     fgets
561                                         strlen
562                                             fseek
563                                                 convert_non_print_to_space
564                                                     strip_trailing_blanks [see line 141]
565                                                         strip_leading_blanks [../../tms_library/format_db_lib.c]
566                                                             setjmp
567                                                                 fddb_error [see line 129]
568                                                                     sprintf
569             exit
570             rewind
571                 map_to_global_section [../../tms_library/global_sub.c]
572                     strlen
573                         MCBLSC
574                             sprintf
575                                 fddb_error [see line 129]
576                                     STOP
577                                         unmap_global_section [../../tms_library/global_sub.c]
578                                             DBLTVA
579                                                 delete_all_global_section [../../tms_library/global_sub.c]
580                                     DBLSC
581                                         create_global_section [../../tms_library/global_sub.c]
582                                             strlen
583                                                 CRWPSC
584                                         init_gcdb_t1 [../../tms_library/fddb_lib.c]
585                                             memcpy
586                                                 init_gcdb_params [gcdb_sub.c]
587                                                     init_fddb_params [../../fddb_sub.c]
588                                                         GETTIM
589                                                             fddb_error [see line 129]
590                                         init_gcdb_data_col_list [gcdb_sub.c]
591                                             memset
592                                                 calc_offsets [../../fddb_sub.c]
593                                                     load_name_table_idx [../../fddb_sub.c]
594                                                         find_fddb_nt_name [see line 474]
595                                                 fclose
596
597                                         fddb_error [../../fddb_sub.c]
598                                         sprintf
599                                         strcpy
600                                         printf
601                                         fprintf
602                                         build_up_arrow [../../fddb_sub.c]
603                                         find_err_text [../../fddb_sub.c]
604                                         sprintf
605                                         strlen
606                                         memcpy
607                                         _filbuf
608
609                                         strip_trailing_blanks [../../tms_library/format_db_lib.c]
610                                         strlen
611
612                                         474
613                                         475
614                                         memcmp
615                                         maincmp
616                                         1

```

build_oprtvdb call

```
550 main [build_oprtvdb.c]
551     printf
552     open
553     exit
554     time
555     ctime
556     sprintf
557     read_fddb_file [./fddb_sub.c]
558         get_next_line [./fddb_sub.c]
559             ftell
560                 fgets
561                     strlen
562                         fseek
563                             convert_non_print_to_space
564                                 strip_trailing_blanks [see line 141]
565                                     strip_leading_blanks [./tms_library/format_db_lib.c]
566                                         fddb_error [see line 129]
567                                             sprintf
568                                                 exit
569
570     rewind
571     map_to_global_section [./tms_library/global_sub.c]
572         strlen
573             MGBLSC
574             sprintf
575                 fddb_error [see line 129]
576                     STOP
577                     unmmap_global_section [./tms_library/global_sub.c]
578                         DELTVA
579                         delete_all_global_section [./tms_library/global_sub.c]
580                         strlen
581                         DGBLSC
582                         create_global_section [./tms_library/global_sub.c]
583                         strlen
584                         CRMPSC
585                         init_oprtvdb_EL [./tms_library/fddb_lib.c]
586                         memcpy
587                         init_oprtvdb_params [oprtvdb_sub.c]
588                         init_fddb_params [./fddb_sub.c]
589                         GETTIM
590                         fddb_error [see line 129]
591                         init_oprtvdb_data_col_list [oprtvdb_sub.c]
592                         memset
593                         calc_offsets [./fddb_sub.c]
594                         load_name_table_idx [./fddb_sub.c]
595                         find_fddb_nt_name [see line 474]
596                         fclose
597
598                         fddb_error [./fddb_sub.c]
599                         sprintf
600                         strcpy
601                         printf
602                             fprintf
603                             build_up_arrow [./fddb_sub.c]
604                             find_err_text [./fddb_sub.c]
605                                 sprintf
606                                 strlen
607                                 memcpy
608                                 _fillbuf
609
610                         strip_trailing_blanks [./tms_library/format_db_lib.c]
611                                         strlen
612                                         memcmp
613                                         find_fddb_nt_name [./tms_library/fddb_lib.c]
```

Build Info

```
1  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300 main [build_rmdb.c]  
301     printf  
302     fopen  
303     exit  
304     time  
305     ctime  
306     fprintf  
307     read_fddb_file [../../fddb_sub.c]  
308     get_next_line [see line 281]  
309     setjmp  
310     fddb_error [see line 5]  
311     sprintf  
312     sprintf  
313     exit  
314     rewind  
315     _fillbuf  
316     malloc  
317     init_rmdb_t1 [../../rmsl_library/fddb_lib.c]  
318     memcpys  
319     init_rmdb_params [rmdb_sub.c]  
320     init_fddb_params [..../fddb_sub.c]  
321     GETTIME  
322     fddb_error [see line 5]  
323     find_rmdb_data_col_list [rmdb_sub.c]  
324     memset  
325     calc_offsets [..../fddb_sub.c]  
326     fddb_error [see line 5]  
327     load_name_table_ndx [..../fddb_sub.c]  
328     find_fddb_nt_name [see line 216]  
329     fclose  
330     mem_sort_rtmfdb_names [rmdb_sub.c]  
331     printf  
332     qsort  
333     fopen  
334     sprintf  
335     perror  
336     exit  
337     fprintf  
338     fclose  
339     delete  
340     fddb_error [..../fddb_sub.c]  
341     sprintf  
342     strlen  
343     memcpys  
344     _fillbuf  
345     memcpys  
346     init_rmdb_t1 [../../rmsl_library/fddb_lib.c]  
347     memcpys  
348     init_rmdb_params [rmdb_sub.c]  
349     init_fddb_params [..../fddb_sub.c]  
350     GETTIME  
351     fddb_error [see line 5]  
352     init_rmdb_data_col_list [rmdb_sub.c]  
353     memset  
354     calc_offsets [..../fddb_sub.c]  
355     fddb_error [see line 5]  
356     load_name_table_ndx [..../fddb_sub.c]  
357     find_fddb_nt_name [see line 216]  
358     fclose  
359     mem_sort_rtmfdb_names [rmdb_sub.c]  
360     printf  
361     qsort  
362     fopen  
363     sprintf  
364     perror  
365     exit  
366     fprintf  
367     fclose  
368     delete  
369     fddb_error [..../fddb_sub.c]  
370     sprintf  
371     strcpy  
372     printf  
373     fprintf  
374     build_up_arrow [..../fddb_sub.c]  
375     find_err_text [..../fddb_sub.c]  
376     sprintf  
377     strlen  
378     memcpys  
379     _fillbuf  
380     get_next_line [..../fddb_sub.c]  
381     ftell  
382     fgets  
383     strlen  
384     fseek  
385     convert_non_print_to_space  
386     strip_trailing_blanks  
387     strip_leading_blanks  
388     fddb_error [see line 5]  
longjmp
```

96/15/4
12:21:55

build_rtdb.c

```
416 main [build_rtdb.c]
417     general_process_startup [./tms_library/proc_cntrl.c]
418         connect_to_mailbox [./tms_library/mailbox.c]
419             strlen
420                 ASSIGN
421                     strcpy
422                         printf
423                             write_to_crash_log [see line 21]
424                             STOP
425                             ASCEFC
426                             clear_all_event_flags [./tms_library/proc_cntrl.c]
427                                 CLRREF
428                                     printf
429                                         STOP
430                                         sprintf
431                                         write_string_to_mailbox_nowait [./tms_library/mailbox.c]
432                                             strlen
433                                                 QIOW
434                                                 log_tms_event [see line 94]
435                                                 map_to_RMDB [./tms_library/fddb.lib.c]
436                                                 map_to_global_section [./tms_library/global_sub.c]
437                                                     strlen
438                                                         MGBLSC
439                                                         init_rmdb_t1 [./tms_library/fddb.lib.c]
440                                         printf
441                                         STOP
442                                         map_to_global_section [see line 436]
443                                         log_tms_event_cc [./tms_library/event_log_sub.c]
444                                         log_tms_common [see line 95]
445                                         exit
446                                         ummap_global_section [./tms_library/global_sub.c]
447                                         DFRVA
448                                         delete_all_global_section [./tms_library/global_sub.c]
449                                         strlen
450                                         DCBLSC
451                                         create_global_section [./tms_library/global_sub.c]
452                                             strlen
453                                             CRMPSC
454                                         sprintf
455                                         memcpy
456                                         GETRIM
457                                         load_rtdb_name_table [build_rtdb.c]
458                                             fopen
459                                         memset
460                                         read_rtdb_name_file [./tms_library/fddb.lib.c]
461                                             fgets
462                                             strlen
463                                             memcp
464                                             strcpy
465                                         memset
466                                         memcpy
467                                         fclose
468                                         exit
469                                         memset
470                                         strcpy
471                                         write_string_to_mailbox [./tms_library/mailbox.c]
472                                             strlen
473                                             memcp
474                                         strcpy
475                                         write_to_crash_log [see line 21]
476                                         STOP
477                                         GETRIM
478                                         NUMRIM
479                                         fopen
480                                         exit
```

build_scheddb.c

```
533 main [build_scheddb.c]
534     printf
535     map_to_vmsdb [../../tms_library/fddb.lib.c]
536     map_to_global_section [../../tms_library/global_sub.c]
537         strlen
538             MGBLSC
539                 init_vmsdb_t1 [../../tms_library/fddb.lib.c]
540                     STOP
541                         fopen
542                             exit
543                                 fgets
544                                     strspn
545                                         memcmp [../../tms_library/misc_func.c]
546                                             tolower
547                                                 rewind
548                                                     map_to_global_section [see line 536]
549                                                         umap_global_section [../../tms_library/global_sub.c]
550                                                             DELTVA
551
552             delete_all_global_section [../../tms_library/global_sub.c]
553                 DABLSC
554                     create_global_section [../../tms_library/global_sub.c]
555                         strlen
556                             ORMPSC
557                                 init_scheddb_t1 [../../tms_library/fddb.lib.c]
558                                     memcpy
559                                         init_scheddb_params [scheddb_sub.c]
560                                             GERTIM
561                     init_scheddb_free_list [scheddb_sub.c]
562                         memset
563                             add_to_scheddb_list [../../tms_library/sched_lib.c]
564                                 init_scheddb [scheddb_sub.c]
565                                     memset
566                                         get_schedule [scheddb_sub.c]
567                                             PRINTF
568                                                 strlen
569                                                     memset
570                                                         get_pv [./fddb_sub.c]
571                                                             get_token
572                                                               strchr
573                       scheddb_error [scheddb_sub.c]
574                           printf
575                               build_up_arrow
576                                   FIBUF
577                                       memcmp [see line 545]
578                                           process_schedule [scheddb_sub.c]
579                                               scheddb_error [see line 574]
580                                                   sprintf
581                                                       convert_sched_type [see line 339]
582                                               GERTIM
583                                               CVT_TO_INTERNAL_TIME
584                                               check_due_in_dow [../../tms_library/sched_lib.c]
585                                                   DAY_OF_WEEK
586                                                       get_scheddb_block [../../tms_library/sched_lib.c]
587                                                         break_scheddb_block [../../tms_library/sched_lib.c]
588                                                 check_due_in_dow [../../tms_library/sched_lib.c]
589                                         compare_VAX_time [../../tms_library/format_db.lib.c]
590                                             insert_in_scheddb_by_time [../../tms_library/vms_1.lib.c]
591                                                 compare_VAX_time [../../tms_library/misc_func.c]
592                                                     add_to_scheddb_list [see line 563]
593
594         convert_sched_type [see line 339]
595             check_date [../../tms_library/misc_func.c]
596                 strlen
597                     scheddb_error [see line 574]
598                         check_number [scheddb_sub.c]
599                             strlen
600                               sprintf
601                                   scheddb_error [see line 574]
602                                     atol
603                                         check_date [../../tms_library/misc_func.c]
604                                             CVT_VECTIM
605                                                 check_number [see line 598]
606                         check_interval [scheddb_sub.c]
607                             strlen
608                                 scheddb_error [see line 574]
609                                     check_number [see line 598]
610                                         sprintf
611                                         check_DOW_mask [scheddb_sub.c]
612                                             toupper
613                                                 scheddb_error [see line 574]
614                                     dow_mask_to_byte [./fddb_sub.c]
615                                         compare_VAX_time
616                                         check_oper_initials [scheddb_sub.c]
617                                             strlen
618                                 sprintf
619                                     scheddb_error [see line 574]
620                                         toupper
621                                             memcp
622                                                 fddb_c1_name [see line 164]
623                                     find_fddb_c1_name [see line 164]
624                                         pad_trailing_blanks [see line 474]
625                                             find_vmsdb_entry_by_name [see line 73]
626                                         memcp
627                                         process_schedule [see line 579]
628                                             follow_scheddb_list [../../tms_library/sched_lib.c]
629                                                 PRINTF
630                                         NUNTM
631                                         STOP
632                                             convert_action_code [see line 344]
633                                         read_fddb_file [./fddb_sub.c]
634                                             get_next_line [./fddb_sub.c]
635                                                 ftell
636                                                 fgets
637                                         strlen
638                                             fseek
639                                         convert_non_print_to_space
640                                         strip_trailing_blanks [see line 142]
641                                         setting
642                                             fddb_error [see line 130]
643                                                 sprintf
644                                         exit
645
646             find_vmsdb_entry_by_name [../../tms_library/vms_1.lib.c]
647                 memcmp
648                     fddb_error [./fddb_sub.c]
649                         sprintf
650                             strcpy
651                               printf
652                                 build_up_arrow [./fddb_sub.c]
653                                     find_err_text [./fddb_sub.c]
654                                         sprintf
655
656             init_schedule [see line 563]
```

26/15/14
6:10:17

138 strlen
139 memory
140 _fillbuf
142 strip_trailing_blanks [././tms_library/format_db.lib.c]
143 strlen
164 find_fddb_cl_name [././tms_library/fddb.lib.c]
165 memcmp
339 convert_sched_type [././tms_library/sched.lib.c]
340 memset
341 strcpy

build_vaxportdb.c

```
572 main [build_vaxportdb.c]
573     printf
574         get_other_process_id [../../tms_library/proc_cntrl.c]
575             strlen
576                 GETJPIW
577             exit
578                 STOP
579             get_port_device_memory [build_vaxportdb.c]
580                 malloc
581                     vaxport_event_msg [vaxportdb_sub.c]
582                         strlen
583                             strcpy
584                                 memcp
585                                     GETTIM
586                                         printf
587                                             NTNTIM
588                                         exit
589                                         realloc
590                                             map_to_RMDB [../../tms_library/fddb_lib.c]
591                                                 map_to_global_section [../../tms_library/global_sub.c]
592                                                     strlen
593                                                         MGBMSC
594                                                         init_rmdb_t1 [../../tms_library/fddb_lib.c]
595                 memcmd
596                     memcmd
597                         vaxport_event_msg [see line 581]
598                         add_to_port_device_table [build_vaxportdb.c]
599                             get_port_device_memory [see line 579]
600                             memcp
601                             convert_to_0x_form [vaxportdb_sub.c]
602                             map_to_VMSDB [../../tms_library/fddb_lib.c]
603                             map_to_global_section [see line 591]
604                             init_vmsdb_t1 [../../tms_library/fddb_lib.c]
605                             map_to_CCTNDB [../../tms_library/fddb_lib.c]
606                             map_to_global_section [see line 591]
607                             init_cctndb_t1 [../../tms_library/fddb_lib.c]
608                             map_to_GBDDB [../../tms_library/fddb_lib.c]
609                             map_to_global_section [see line 591]
610                             init_gbdedb_t1 [../../tms_library/fddb_lib.c]
611                             find_fddb_cl_name [see line 167]
612                             map_to_OPRVDB [../../tms_library/fddb_lib.c]
613                             map_to_global_section [see line 591]
614                             init_oprvdb_t1 [../../tms_library/fddb_lib.c]
615                             map_to_GCDB [../../tms_library/fddb_lib.c]
616                             map_to_global_section [see line 591]
617                             init_gcdb_t1 [../../tms_library/fddb_lib.c]
618                             qsort
619                             find_usage_in_multiple_dbs [build_vaxportdb.c]
620             memset
621                 memcmp
622             build_dev_port_name_string [see line 374]
623                 memset
624                     printf
625                         _fillbuf
626                         find_duplicate_dev_addr [build_vaxportdb.c]
627             memset
628                 memcmp
629             build_dev_port_name_string [see line 374]
630                 memset
631                     count_port_entries [build_vaxportdb.c]
632             memset
633                 memcmp
634             memcp
635             memcp
```

build_vmsdb.caff

```

718 main [build_vmsdb.c]
719     printf
720     fopen
721     exit
722     time
723     ctime
724     fprintf
725     read_fddb_file [./fddb_sub.c]
726     get_next_line [./fddb_sub.c]
727     ftell
728     fgets
729     strlen
730     fseek
731     convert_non_print_to_space
732     strip_trailing_blanks [see line 75]
733     strip_leading_blanks [./tms_library/format_db.lib.c]
734     setjmp
735     fddb_error [see line 63]
736     sprintf
737     exit
738     rewind
739     map_to_global_section [./tms_library/global_sub.c]
740     strlen
741     MGBLSC
742     sprintf
743     fddb_error [see line 63]
744     STOP
745     umap_global_section [./tms_library/global_sub.c]
746     DELTVA
747     delete_all_global_section [./tms_library/global_sub.c]
748     strlen
749     DGBUSC
750     create_global_section [./tms_library/global_sub.c]
751     strlen
752     CRMPSC
753     init_vmsdb_t1 [./tms_library/fddb_lib.c]
754     memcpys
755     init_vmsdb_params [vmsdb_sub.c]
756     init_fddb_params [./fddb_sub.c]
757     GERTIM
758     fddb_error [see line 63]
759     init_vmsdb_data_col_list [vmsdb_sub.c]
760     memset
761     calc_offsets [./fddb_sub.c]
762     init_vmsdb_free_list [./tms_library/vms_lib.c]
763     memset
764     add_to_rel_list
765     load_name_table_ndx [./fddb_sub.c]
766     find_fddb_nt_name [see line 646]
767     fclose
768     process_message [see line 115]
769     process_library [see line 167]
770     process_queue [see line 183]
771     process_cluster [see line 104]

71     strlen
72     memcpys
73     _fillbuf
74     strip_trailing_blanks [./tms_library/format_db.lib.c]
75     process_cluster [vmsdb_sub.c]
76     get_vmsdb_block [./tms_library/vms_lib.c]
77     break_vmsdb_block [./tms_library/vms_lib.c]
78     sprintf
79     fddb_error [see line 63]
80     init_cluster [vmsdb_sub.c]
81     memset
82     memcpys
83     byte_to_table [./tms_library/table_sub.c]
84     strlen
85     add_to_rel_list [./tms_library/vms_lib.c]
86     process_message [vmsdb_sub.c]
87     fddb_error [see line 63]
88     init_message [vmsdb_sub.c]
89     memset
90     memset
91     byte_to_table
92     sprintf
93     center_justify [./tms_library/misc_func.c]
94     strlen
95     memcpys
96     memset
97     memcpys
98     get_vmsdb_block [see line 105]
99     build_fp_message [./tms_library/vms_lib.c]
100    crack_vms_message [./tms_library/vms_lib.c]
101    init_vms_msg_struct [./tms_library/vms_lib.c]
102    copy_flash [./tms_library/vms_lib.c]
103    build_fp_static_msg [./tms_library/vms_lib.c]
104    build_fp_header [./tms_library/vms_lib.c]
105    memset
106    memset
107    memset
108    memset
109    memset
110    memset
111    memset
112    memset
113    memset
114    memset
115    memset
116    fddb_error [see line 63]
117    init_message [vmsdb_sub.c]
118    memset
119    memset
120    memset
121    memset
122    memset
123    memset
124    memset
125    memset
126    memset
127    memset
128    memset
129    build_fp_message [./tms_library/vms_lib.c]
130    init_vms_msg_struct [./tms_library/vms_lib.c]
131    s.lib.c]
132    s.lib.c]
133    s.lib.c]
134    s.lib.c]
135    s.lib.c]
136    s.lib.c]
137    s.lib.c]
138    s.lib.c]
139    toupper
140    toupper
141    toupper
142    toupper
143    toupper
144    toupper
145    toupper
146    toupper
147    toupper
148    build_fp_arrow_msg [./tms_library/vms_lib.c]
149    build_fp_msg_header
150    strlen
151    toupper
152    toupper
153    toupper
154    toupper
155    toupper
156    toupper
157    toupper
158    toupper
159    toupper
160    toupper
161    toupper
162    toupper
163    toupper
164    toupper
165    strcpy
166    printff
167    printff
168    build_up_arrow [./fddb_sub.c]
169    find_err_text [./fddb_sub.c]
170    sprintf
171    strlen

```

build_vmsdbcall

```

154     ] FP_position_cursor [see line 142
155   ] FP_left_arrow_to_buffer [...] /tms_library/vms.lib.c
156   ]   strlen
157   ]     FP_Position_cursor [see line 142
158   ]       calc_FP_msg_time [...] /tms_library/vms.lib.c
159   ]         crack_vms_message [see line 130]
160   ]           strlen
161   ]             atof
162   ]               atoi
163   ]                 return_vmsdb_block [...] /tms_library/vms.lib.c
164   ]                   memset
165   ]                     combine_adjacent_vmsdb_block [...] /tms_library/vms.lib.c
166   ]                         add_to_rel_list
167   ]                           process_library [vmsdb_sub.c]
168   ]                             get_vmsdb_block [see line 105]
169   ]                               sprintf
170   ]                                 fddb_error [see line 63]
171   ]                                   init_library [vmsdb_sub.c]
172   ]                                     memset
173   ]                                       memcpy
174   ]                                         byte_to_table
175   ]                                           strlen
176   ]                                             add_to_rel_list
177   ]                                               calc_FP_lib_slot_no [...] /tms_library/vms.lib.c
178   ]                                                 find_vmsdb_entry_by_name [...] /tms_library/vms.lib.c
179   ]                                                   memcmp
180   ]                                                     sprintf
181   ]                                                       delete_from_rel_list [...] /tms_library/vms.lib.c
182   ]                                                         return_vmsdb_block [see line 163]
183   ]     process_queue [vmsdb_sub.c]
184   ]       get_vmsdb_block [see line 105]
185   ]         sprintf
186   ]           fddb_error [see line 63]
187   ]             init_queue [vmsdb_sub.c]
188   ]               memset
189   ]                 memcpy
190   ]                   byte_to_table
191   ]                     strlen
192   ]                       add_to_rel_list
193   ]                         find_vmsdb_entry_by_name [see line 178]
194   ]                           calc_FP_lib_slot_no [see line 177]
195   ]                             delete_from_rel_list
196   ]                               return_vmsdb_block [see line 163]
197   ]                                 build_FP_queue [...] /tms_library/vms.lib.c
198   ]                                   find_vmsdb_entry_by_name [see line 178]
199   ]                                     find_msg_in_library [...] /tms_library/vms.lib.c
200   ]                                       memcmp
201   ]                                         calc_FP_msg_time [see line 158]
202   ]                                           atoi
203   ]                                             sprintf

```

COMMISSIONS INDICALES

90/05/14
16:52:36

2

comm.stats ioctl

```
534     sprintf
535         log_tms_event [see line 94]
536     CANTIM
537     restore_tty_process_name [...]tms_library/proc_cntrl.c]
538     strlen
539     set_process_name [see line 438]
540     printf
541     get_terminal_name [see line 435]
542     get_pid [...]tms_library/proc_cntrl.c]
543     GETUPI
544     sprintf

21     write_to_crash_log [...]tms_library/proc_cntrl.c]
22     GERRTM
23     NORMTM
24     fopen
25     fprintf
26     fclose

94     log_tms_event [...]tms_library/event_log_sub.c]
95     log_tms_common [...]tms_library/event_log_sub.c]
96     strlen
97     write_to_crash_log [see line 21]
98     GERRTM
99     memcpy
100    memset
101    write_to_mailbox_nowait [...]tms_library/mailbox.c]
102    QIOW
```

96/5/14
16:55:18

1

Count_tms_lines.call

```
1 main [count_tms_lines.c]
2   open
3   printf
4   exit
5   time
6   ctime
7   sprintf
8   fgets
9   memcmp
10  extract_name [count_tms_lines.c]
11  strcpy
12  strlen
13  strcat
14  count_lines [count_tms_lines.c]
15  fopen
16  printf
17  fprintf
18  fgets
19  strlen
20  fclose
21  fclose
```

```

421 main [crack_fmdb_dailyfil.c]
422 Printf
423 scanf
424 flush_input [./tms_library/kb_func.c]
425 _filbuf
426 strlen
427 toupper
428 check_fmdb_filename_format [./tms_library/fmdb_lib.c]
429 printf
430 up_arrow [./tms_library/fmdb_lib.c]
431 _filbuf
432
433 read_df_hdr_name_list [./tms_library/fmdb_lib.c]
434 strcpy
435 strcat
436 strlen
437 OPEN
438 sprintf
439 malloc
440 QIOW
441 free
442 strchr
443 memcmp
444 check_filename [./tms_library/misc_func.c]
445 strlen
446 find_string [./tms_library/misc_func.c]
447 strlen
448 memcmp
449 get_number_from_header [./tms_library/misc_func.c]
450 strlen
451 find_string [see line 446]
452 _filbuf
453 clear_R0 [./tms_library/misc_func.c]
454 strtok
455 memset
456 exit
457 SSTOP
458 find_string [see line 446]
459 _filbuf
460 vms_c_getch [see line 375]
461 fprintf
462 fputc
463 dump_fmdb_name_list [./tms_library/fmdb_lib.c]
464 printf
465 strlen
466 fprintf
467 memset
468 fclose
469 memcmp
470 read_df_elm_data [./tms_library/fmdb_lib.c]
471 memset
472 sprintf
473 QIOW
474 memmove
475 memset
476 dump_df_elm_data [./tms_library/fmdb_lib.c]
477 printf
478 fprintf
479 unpack_fmdb_loop
480 unpack_fmdb_station
481 unpack_fmdb_spd_trap
482 free
483 DASSGN

```

```

427 main [crack_fmdb_namefile.c]
428     printf
429     scanf
430     flush_input [./tms_library/kb_func.c]
431     _filbuf
432     strlen
433     toupper
434     check_fmdb_filename_format [./tms_library/fmdb_lib.c]
435     printf
436     ug_arrow [./tms_library/fmdb_lib.c]
437     _filbuf
438     strlen
439     read_fmdb_namefile [./tms_library/fmdb_lib.c]
440     strcpy
441     strcat
442     strlen
443     OPEN
444     printf
445     malloc
446     QIOW
447     DASSGN
448     strchr
449     memcmp
450     dump_mem [see line 191]
451     _filbuf
452     check_filename [./tms_library/misc_func.c]
453     strlen
454     find_string [./tms_library/misc_func.c]
455     strlen
456     memcmp
457     get_number_from_header [./tms_library/misc_func.c]
458     strlen
459     find_string [see line 454]
460     clear_R0 [./tms_library/misc_func.c]
461     strtoul
462     exit
463     SROP
464     find_string [see line 454]
465     _filbuf
466     vms_c_Setch [see line 381]
467     fprintf
468     fputc
469     dump_fmdb_name_list [./tms_library/fmdb_lib.c]
470     printf
471     strlen
472     fprintf
473     memcmp
474     fclose
475     strcpy
476     fopen
477     perror
478

        dump_mem [./tms_library/dump_mem.c]
        print_hex_ascii_line [./tms_library/dump_mem.c]
        sprintf
        print_n_skipped [./tms_library/dump_mem.c]
        printf

```

crack_fmdb_snapshot.c

```
424 main [crack_fmdb_snapshot.c]
425     printf
426         scanf
427             _flush_input [./tms_library/kb_func.c]
428                 _filbuf
429                     strlen
430                         toupper
431                             check_fmdb_filename_format [.../tms_library/fmdb_lib.c]
432                                 printf
433                                     up_arrow [.../tms_library/fmdb_lib.c]
434                                         _filbuf
435                                             read_fmdb_snapshot [./tms_library/fmdb_lib.c]
436                                                 strcpy
437                                                     strcat
438                                                         strlen
439                                                             OPEN
440                                                               PRINTF
441                     PUTMSG
442                         malloc
443                             QIOW
444                               DASSAN
445                                   strchr
446                                       memcmp
447                                           dump_mem [see line 191]
448                                               _filbuf
449                                                   check_filename [./tms_library/misc_func.c]
450                                                       strlen
451                                             find_string [./tms_library/misc_func.c]
452                                                 strlen
453                                                   memcmp
454                                                     find_string [see line 452]
455                                                       memcpy
456                                             get_number_from_header [./tms_library/misc_func.c]
457                                                 strlen
458                                                   find_string [see line 452]
459                                                       clear_R0 [./tms_library/misc_func.c]
460                                                       strlen
461                                                       exit
462                                                       STOP
463                                                       find_string [see line 452]
464                                                       _filbuf
465                                             read_fmdb_namefile [./tms_library/fmdb_lib.c]
466                                                 strcpy
467                                                     strcat
468                                                       strlen
469                                                         OPEN
470                                                               printf
471                     malloc
472                         QIOW
473                             DASSAN
474                                 strchr
475                                     memcmp
476                                         dump_mem [see line 191]
477                                             _filbuf
478                                               check_filename [see line 450]
479                                                   get_number_from_header [see line 458]
480                                                       vms_c_getch [see line 378]
481                                             fprintf
482                                               dump_fmdb_data_col [./tms_library/fmdb_lib.c]
483                                                 printf
484
485                                         dump_fmdb_data_col [./tms_library/fmdb_lib.c]
486                                             strlen
487                                              
```

del_actvdb.c

16/05/04
17:15:00

```
436 main [del_actvdb.c]
437     delete_global_section [./tms_library/global_sub.c]
438     strlen
439     DBLSC
440     printf
```

060514
15:30:14

```
del_cctvdb.c

436 main [del_cctvdb.c]
437     delete_global_section [...] /tms_library/global_sub.c]
438
439     strlen
440     DCBLSC
441     printf
```

del_fmdb.call

```
1 main [del_fmdb.c]
2 delete_global_section
3 printf
```

17/3645

del_gbl_db.call

1 main [del_gbl_db.c]
2 delete_global_section
3 printf

17:38:24

1

del_gcdb.call

1 main [del_gcdb.c]
2 delete_global_section
3 printf

06/05/14
117:40:49

del_optvdb.call

```
1 main [del_optvdb.c]
2 delete_global_section
3 printf
```

del_imdb.c

436 main [del_imdb.c]
437 delete_global_section [../../tms_library/global_sub.c]
438 strlent
439 DGBLSC
440 printf

del_qdb call

```
436 main [del_qdb.c]
437     delete_global_section [.. /tms_library/global_sub.c]
438     strien
439     DEBSC
440     printf
```

06/05/05
13:08:18

```
608 main [del_scheddb.c]
609     delete_global_section [../../tms_library/global_sub.c]
610     strlen
611     DEBUGC
612     printf
```

del_vaxportdb call

```
575 main [dal_vaxportdb.c]
576     delete_global_section [../../tms_library/global_sub.c]
577     strlen
578     DEBUGSC
579     printf
```

1
del_vmsdb.c

```
733 main [del_vmsdb.c]
734     delete_global_section [../../../tms_library/global_sub.c]
735     strlen
736     DBUSC
737     printf
```

dumydata call

```
 26      fclose
 27
 28      log_tms_event [./tms_library/event_log_sub.c]
 29          log_tms_common [see line 95]
 30
 31      strlen
 32          write_to_crash_log [see line 21]
 33              GETTIM
 34                  memcpy
 35                      memset
 36                          write_to_mailbox_nowait [./tms_library/mailbox.c]
 37
 38      ASSIGN
 39
 40      strcpy
 41          PRINTF
 42              write_to_crash_log [see line 21]
 43
 44      STOP
 45
 46      general_process_startup [./tms_library/proc_cntr1.c]
 47          connect_to_mailbox [./tms_library/mailbox.c]
 48              strlen
 49
 50      SETPRN
 51
 52      set_process_name [./tms_library/proc_cntr1.c]
 53          strlen
 54
 55      QROW
 56
 57      ASCEFC
 58          clear_all_event_flags [./tms_library/proc_cntr1.c]
 59              CLRREF
 60                  printf
 61                      stop
 62
 63      sprintf
 64          write_string_to_mailbox_nowait [./tms_library/mailbox.c]
 65
 66      QROW
 67
 68      log_tms_event [see line 94]
 69          map_to_RMDB [./tms_library/rmdb.lib.c]
 70              map_to_global_section [./tms_library/global_sub.c]
 71
 72      strlen
 73          MGBLSC
 74
 75      init_rtdb_t1 [./tms_library/rtdb.lib.c]
 76
 77      map_to_RMDB [./tms_library/rmdb.lib.c]
 78          map_to_global_section [see line 444]
 79              init_rtdb_t1 [./tms_library/rtdb.lib.c]
 80
 81      fopen
 82          exit
 83              fclose
 84
 85      WALTER
 86          CLRREF
 87
 88      SETPRN
 89
 90      read_data_file [dumydata.c]
 91          fgets
 92              strlen
 93                  sprintf
 94                      log_tms_event [see line 94]
 95                          memcmp
 96
 97      rewind
 98
 99      mememp
 100         sprintf
 101             find_fiddb_c1_name [./tms_library/fiddb.lib.c]
 102                 memcmp
 103
 104         memcpy
 105             atoi
 106                 sscnf
 107                     search_rtdb_name_table [./tms_library/rtdb.lib.c]
 108                         memcmp
 109                         pack_rtdb_spd_trap [./tms_library/pack_lib.c]
 110                             atof
 111                                 pack_rtdb_loop [./tms_library/pack_lib.c]
 112                                     READDF
 113
 114         write_to_crash_log [./tms_library/proc_cntr1.c]
 115             GETTIM
 116                 NDNTIM
 117                     fopen
 118                         fprintf
 119
 120
 121
 122
 123
 124
 125
```

event_logger.c

```
434 main [event_logger.c]
435     create_event_log_file [event_logger.c]
436         GETTIME
437             write_to_crash_log [see line 21]
438                 printf
439                     STOP
440                         NUMTIME
441                             sprintf
442                                 strlen
443                                     CREATE
444                                         CONNECT
445                                             PUT
446                                                 strcpy
447                                                     FLASH
448             connect_to_mailbox [see line 353]
449                 printf
450                     event_log_int_write [event_logger.c]
451                         GETTIME
452                             write_to_crash_log [see line 21]
453                                 NUMTIME
454                                     sprintf
455                                         strlen
456                                             PUT
457             write_to_crash_log [see line 21]
458                 STOP
459             write_string_to_mailbox_noawait [see line 366]
460                 map_to_VAXPORTDB [./tms_library/fddh_lib.c]
461                     map_to_global_section [./tms_library/global_sub.c]
462                         strlen
463                             MGBLSC
464             init_vxportdb_t1 [./tms_library/fddh_lib.c]
465                 find_first_last_port_unit [./tms_library/find_first_last.c]
466                     sprintf
467                         exit
468                         ASCEFC
469             queued_read_from_mailbox [./tms_library/mailbox.c]
470                 QIO
471                     set_process_name [see line 299]
472                         set_2min_timer [event_logger.c]
473                             GETTIME
474                                 write_to_crash_log [see line 21]
475                         STOP
476                             NUMTIME
477                                 sprintf
478                                     event_log_int_write [see line 450]
479             close_event_log_file [event_logger.c]
480                 GETTIME
481                     write_to_crash_log [see line 21]
482                         printf
483                             STOP
484                                 NUMTIME
485                                     strcpy
486                                         strlen
487                                             PUT
488                                                 sprintf
489                                                     FLUSH
490             CLOSE
491             create_event_log_file [see line 435]
492                 CVT_VECTIM
493                     ADD_TIMES
494                         SETIM
495                             fclose
496                                 WFOR
497                                     REDEF
```

fmdb_aggr call

```
422 main [fmdb_aggr.c]
423     general_process_startup [./tms_library/proc_cntrl.c]
424     connect_to_mailbox [./tms_library/mailbox.c]
425         strlen
426             ASSIGN
427                 strcpy
428                     printf
429                         write_to_crash_log [see line 21]
430                         STOP
431                         ASCEFC
432                             clear_all_event_flags [./tms_library/proc_cntrl.c]
433                                 CLREF
434                                     printf
435                                         STOP
436                                             sprintf
437                                                 write_string_to_mailbox_nowait [./tms_library/mailbox.c]
438                                                     QIDW
439             strlen
440                 log_tms_event [see line 94]
441                 map_to_RMDB [./tms_library/rtdb.lib.c]
442                     map_to_global_section [./tms_library/global_sub.c]
2:TAHITI: /homes/taylorc/TSMC/code\007wq!: Command not found.
443                         init_rtdb_t1 [./tms_library/rtdb.lib.c]
444                         PRINTF
445                         STOP
446                         map_to_FMDB [./tms_library/fmdb.lib.c]
447                             map_to_global_section [see line 442]
448                                 init_fmdb_t1 [./tms_library/fmdb.lib.c]
449                                     memset
450                                         fclose
451                                         WAITPR
452                                         log_tms_event_cc [./tms_library/event_log_sub.c]
453                                         log_tms_common [see line 95]
454                                         CLREF
455                                         memCPY
456                                         unpack_rtdb_loop_stn [./tms_library/pack.lib.c]
457                                         unpack_rtdb_station [./tms_library/pack.lib.c]
458                                         unpack_rtdb_spd_trap [./tms_library/pack.lib.c]
459                                         NUNTRIM
460                                         pack_fmdb_loop [./tms_library/pack.lib.c]
461                                         pack_fmdb_loop [./tms_library/pack.lib.c]
462                                         pack_fmdb_stn [./tms_library/pack.lib.c]
463                                         pack_fmdb_station [./tms_library/pack.lib.c]
464                                         pack_fmdb_spd_trap [./tms_library/pack.lib.c]
465                                         scroll_fmdb_col_offsets [./tms_library/fmdb.lib.c]
466                                         memset
467                                         SETEF
468                                         READEF
469                                         write_string_to_mailbox [./tms_library/mailbox.c]
470                                         strlen
471                                         QIDW
472                                         exit
```

fmdb_archiver.c

```
416 main [fmdb_archiver.c]
417     general_process_startup [./tms_library/proc_cntr1.c]
418     connect_to_mailbox [./cms_library/mailbox.c]
419         strlán
420             ASSIGN
421             strcpy
422             printf
423             write_to_crash_log [see line 21]
424             STOP
425             ASCIIFC
426             clear_all_event_flags [./tms_library/proc_cntr1.c]
427                 CLRFP
428                 printf
429                 STOP
430                 sprintf
431                 write_string_to_mailbox_nowait [./tms_library/mailbox.c]
432                     strlán
433                         QIOW
434                         log_tms_event [see line 88]
435                         CREF
436                         Printf
437                         log_tms_event_cc [./tms_library/event_log_sub.c]
438                             log_tms_common [see line 89]
439                             STOP
440                             map_to_FMDB [./tms_library/Fmdb.lib.c]
441                             map_to_global_section [./tms_library/global_sub.c]
442                             strlán
443                             MGBLSC
444                             init_fmdb_t1 [./tms_library/Fmdb.lib.c]
445                             GETTIM
446                             calc_next_20sec_time [fmdb_archiver.c]
447                             NUMTIM
448                             printf
449                             log_tms_event_cc [see line 437]
450                             CWT_VECTIM
451                             ADD_TIMES
452                             build_fmdb_snapshot_filename [./tms_library/Fmdb.lib.c]
453                             NUMTIM
454                             PRINTF
455                             build_fmdb_daily_filename [./tms_library/Fmdb.lib.c]
456                             NUMTIM
457                             sprintf
458                             malloc
459                             exit
460                             memset
461                             build_id_snapshot_header [fmdb_archiver.c]
462                             strcpy
463                             strlen
464                             strcat
465                             sprintf
466                             NUMTIM
467                             printf
468                             log_tms_event_cc [see line 437]
469                             log_tms_event [see line 88]
470                             create_fmdb_snapshot_file [fmdb_archiver.c]
471                             strcpy
472                             strcat
473                             strlen
474                             CREATE
475                             create_fmdb_daily_file [fmdb_archiver.c]
476                             strcpy
```

inc_detect.c

```
458     search_rtdb_name_table [../../tms_library/rtdb_lib.c]
459         memcmp
460         realloc
461         code_byte_to_table [../../tms_library/table_sub.c]
462         code_ushort_to_table [../../tms_library/table_sub.c]
463         code_ulong_to_table [../../tms_library/table_sub.c]
464         byte_to_table [../../tms_library/table_sub.c]
465         fclose
466         calc_table_checksum [../../tms_library/table_sub.c]
467         fclose
468         sprintf
469         WAITER
470         CLREF
471         calc_inc_det [inc_detect.c]
472         strcpy
473         unpack_rtdb_loop_stn [../../tms_library/pack_lib.c]
474         incident_detect [inc_detect.c]
475         pack_rtdb_inc_det [../../tms_library/pack_lib.c]
476         search_rtdb_offset_list [../../tms_library/rtdb_lib.c]
477         sprintf
478         log_tms_event [see line 413]
479         log_tms_common [see line 414]
480         SETEF
481         READDEF
482         write_string_to_mailbox [../../tms_library/mailbox.c]
483         strlen
484         QIOW
485         exit
486
487     stop
488     ASCEFC
489     clear_all_event_flags [../../tms_library/proc_ctrl.c]
490     PRINTF
491     strlent
492     log_tms_event [../../tms_library/event_log_sub.c]
493     log_tms_common [../../tms_library/event_log_sub.c]
494     log_tms_event [../../tms_library/event_log_sub.c]
495     log_tms_common [../../tms_library/event_log_sub.c]
496     write_to_crash_log [see line 21].
497     GETRTIM
498     memcpy
499     memset
500     write_to_mailbox_nowait [see line 393]
501     map_to_RMDB [../../tms_library/rtdb_lib.c]
502     map_to_global_section [../../tms_library/global_sub.c]
503     strlent
504     MGBLSC
505     init_rtdb_t1 [../../tms_library/rtdb_lib.c]
506     log_tms_event_cc [../../tms_library/event_log_sub.c]
507     log_tms_common [see line 414]
508     printf
509     STOP
510     map_to_RMDB [../../tms_library/rtdb_lib.c]
511     map_to_global_section [see line 422]
512     init_rmdb_t1 [../../tms_library/fddb_lib.c]
513     build_inc_det_table [inc_detect.c]
514     malloc
515     build_tap_error [../../tms_library/tap_sub.c]
516     sprintf
517     memcpy
518     printf
519     log_tms_event [see line 413]
520     exit
521     code_to_table [../../tms_library/table_sub.c]
522     ushort_to_table [../../tms_library/table_sub.c]
523     ulong_to_table [../../tms_library/table_sub.c]
524     date_time_to_table [../../tms_library/table_sub.c]
525     GETRTIM
526     classify_roadway [../../tms_library/rddb_lib.c]
527     PRINTF
528     memcpy
529     fopen
530     fgets
531     pad_end [../../tms_library/tap_sub.c]
532     strlen
533     find_fddb_cl_name [../../tms_library/fddb_lib.c]
534     memcmp
535     classify_roadway [../../tms_library/rddb_lib.c]
536     PRINTF
537     memcpy
538     fopen
539     fgets
540     pad_end [../../tms_library/tap_sub.c]
541     strlen
542     find_fddb_cl_name [../../tms_library/fddb_lib.c]
543     memcmp
544     classify_roadway [../../tms_library/rddb_lib.c]
545     PRINTF
546     memcpy
547     fopen
548     fgets
549     pad_end [../../tms_library/tap_sub.c]
```

not_event_logged

```
408 main [mon_event_log.c]
409     get_other_process_id [./tms_library/proc_cntrl.c]
410         strlen
411             GETJPIW
412                 printf
413                     exit
414                         STOP
415                             establish_process_name [./tms_library/proc_cntrl.c]
416                                 memset
417                                     strlen
418                                         strcpy
419                                             get_process_name [./tms_library/proc_cntrl.c]
420                                                 memset
421                                                     GENJPI
422                                                         strcmp
423                                                             get_terminal_name [./tms_library/proc_cntrl.c]
424                                                                 memset
425                                                                     GETJPIW
426             set_process_name [./tms_library/proc_cntrl.c]
427                 strlen
428                     SETPRN
429                         connect_to_mailbox [see line 335]
430                             flush_mailbox [./tms_library/mailbox.c]
431                                 QIO
432                                     log_tms_common [see line 95]
433                                         get_iocchan [./tms_library/kb_func.c]
434                                             strlen
435                                                 ASSIGN
436             get_term_char [./tms_library/kb_func.c]
437                 QIO
438             set_port_partial [./tms_library/kb_func.c]
439                 change_term_char [./tms_library/kb_func.c]
440                     QIO
441             CREATE_PASTEBORD
442                 CHANGE_PBD_CHARACTERISTICS
443                     CREATE_VIRTUAL_DISPLAY
444                         queued_get_lchar [./tms_library/kb_func.c]
445                             QIO
446             queued_read_from_mailbox [./tms_library/mailbox.c]
447                                 QIO
448                     SETIOR
449                         WFLOR
450                             READDEF
451                                 prompt_for_yes_no [./tms_library/kb_func.c]
452                                     strlen
453                                         printf
454                                         vms_c_getch [see line 364]
455                                         toupper
456                                         -flbuf
457                                         CANTIM
458                                         format_event_log_msg [./tms_library/format_el_msg.c]
459                                         NUMRIM
460                                         write_to_crash_log [see line 21]
461                                         sprintf
462                                         memcpy
463                                         strcpy
464                                         GETRIM
465                                         NUMTIM
466                                         memset
467                                         sprintf
468                                         CANCEL
469                                         DELETE_PASTEBORD
470                                         set_term_char [./tms_library/kb_func.c]
```

noaa_monitor.call

```
417 main [noaa_monitor.c]
418     general_process_startup [./tms_library/proc_ctrl.c]
419     connect_to_mailbox [./tms_library/mailbox.c]
420         strlen
421             ASSIGN
422                 STOP
423                     strcpy
424                         printf
425                             write_to_crash_log [see line 21]
426                                 STOP
427                                     ASBFC
428                                         clear_all_event_flags [./tms_library/proc_ctrl.c]
429                                             printf
430                                                 STOP
431                                                     sprintf
432                                                         write_string_to_mailbox_nowait [./tms_library/mailbox.c]
433                                                             strlen
434                                                               QIOW
435                     log_tms_event [see line 94]
436             set_process_name [see line 296]
437                 strcpy
438                     printf
439                         log_tms_event_cc [./tms_library/event_log_sub.c]
440                             log_tms_common [see line 95]
441                               write_to_crash_log [see line 21]
442                                   STOP
443                                       read_report_list [noaa_monitor.c]
444                                           fopen
445                                               strcpy
446                                                 perror
447                                                   log_tms_event [see line 94]
448                                                       write_to_crash_log [see line 21]
449                                                       exit
450               malloc
451                 printf
452                   fgets
453                     pad_end [./tms_library/tap_sub.c]
454                         strlen
455                           memcmp
456                             memcpy
457                               realloc
458                                 fclose
459                                   map_to_GBLDB [./tms_library/fddb_lib.c]
460                                       map_to_global_section [./tms_library/global_sub.c]
461                                         strlen
462                                           MBLSIC
463                                             init_gbl_db_t1 [./tms_library/fddb_lib.c]
464                                               map_to_VAXPORTDB [./tms_library/fddb_lib.c]
465                                                 map_to_global_section [see line 460]
466                                                   init_vaxportdb_t1 [./tms_library/fddb_lib.c]
467             create_logical_name [./tms_library/logical_name.c]
468                 strlen
469                   CRELEMN
470                     malloc
471                         exit
472                           start_noaa_port [noaa_monitor.c]
473                             find_fddb_c1_name [./tms_library/fddb_lib.c]
474                               memcmp
475                                 strcpy
476                                   printf
477                                     log_tms_event [see line 94]
478                                       write_to_crash_log [see line 21]
479                                         exit
480                                           strcmp
```

2

noaa_monitor.cai

bit_clr_i

545 bit_clr_i
546 CANTIM

set_process_name [.. /tms_library/proc_cntrl.c]
296 str_en
297 SETPRN
298

patch_cctvdb.call

```
516 main [patch_cctvdb.c]
517     map_to_cctvdb [../../tms_library/fddb_lib.c]
518     map_to_global_section [../../tms_library/global_sub.c]
519     strlen
520     MGBLSC
521     init_cctvdb_t1 [../../tms_library/fddb_lib.c]
522     printf
523     STOP
524     map_to_OOPTVDB [../../tms_library/fddb_lib.c]
525     map_to_global_section [see line 518]
526     init_oprtvdb_t1 [../../tms_library/fddb_lib.c]
527     map_to_VAXPORTDB [../../tms_library/fddb_lib.c]
528     map_to_global_section [see line 518]
529     init_vaxportdb_t1 [../../tms_library/fddb_lib.c]
530     fopen
531     exit
532     get_iocchan [../../tms_library/kb_func.c]
533     strlen
534     ASSIGN
535     get_term_char [../../tms_library/kb_func.c]
536     QIO
537     change_term_char [../../tms_library/kb_func.c]
538     ASSIGN
539     setjmp
540     vms_c_getch [../../tms_library/kb_func.c]
541     input [../../tms_library/kb_func.c]
542     QIO
543     toupper
544     dump_cctvdb_to_table_files [cctvdb_sub.c]
545     printf
546     translate_logical_name [../../../../tms_library/logical_name.c]
547     strlen
548     TRNUMW
549     memmove
550     build_fullname [../../fddb_sub.c]
551     strlen
552     memcpy
553     fopen
554     fddb_error [see line 130]
555     fclose
556     delete
557     sprintf
558     format_db_element_for_output [../../../../tms_library/format_d
559     time
560     ctime
561     fprintf
562     dump_fddb_elements [../../fddb_sub.c]
563     format_db_element_for_output [../../../../tms_library/format_db_
564     b.lib.c]
565     b.lib.c]
566     strcpy
567     strip_trailing_blanks [see line 142]
568     process_output_special_case [cctvdb_sub.c]
569     Convert_perm_mask [../../../../tms_library/format_db.lib.c]
570     ib.c]
571     ] 572     at_ab.lib.c]
573     574
575     street
576     strmp
577     NUMRIM
578     memset
579     print_data_col_list [patch_cctvdb.c]
580     printf
581     select_entry [../../../../tms_library/kb_func.c]
582     log10
583     printf
584     exit
585     memset
586     vms_c_getch [see line 540]
587     longjmp
588     atoi
589     longjmp
590     menu2 [patch_cctvdb.c]
591     printf
592     vms_c_getch [see line 540]
593     toupper
594     dump_cctvdb_to_table_files [see line 545]
595     print_group_name_list [patch_cctvdb.c]
596     memset
597     strlen
598     memcpy
599     printf
600     select_entry [see line 581]
601     longjmp
602     menu3 [patch_cctvdb.c]
603     printf
604     vms_c_getch [see line 540]
605     toupper
606     print_element_name_list [patch_cctvdb.c]
607     select_entry [see line 581]
608     longjmp
609     menu4 [patch_cctvdb.c]
610     format_db_element_for_out
611     put [see line 564]
612     613     ]
614     615     616     s_library/kb_f
617     unc.c]
618     619     620     line 540]
621     e
622     623     624     625     longjmp
626     output_cctvdb_out_fi [cctvdb_sub.c]
627     fopen
628     fddb_error [see line 130]
629     read_fddb_file [../../../../fddb_sub.c]
630     get_next_line [../../../../fddb_sub.c]
631     ftell
632     633     634
```

```

patch_cctvdbcall

632     fgets
633     strlen
634     fseek
635     convert_non_print_to_space
636     strip_trailing_blanks [see line 142]
637     strip_leading_blanks [../../tms_library/format_d
b_lib.c]

638     setjmp
639     fdab_error [see line 130]
640     sprintf
641     exit
642     time
643     ctime
644     fprintf
645     printf
646     rewind
647     fseek
648     init_group_table [see line 150]
649     print_db_table [./fdab_sub.c]
650     print_file_comments [./fdab_sub.c]
651     fseek
652     get_next_line [see line 630]
653     fprintf
654     copy_params_lines [./fdab_sub.c]
655     fseek
656     get_next_line [see line 630]
657     fprintf
658     format_db_element_for_output [see line 564]
659     print_db_table_special_case [cctvdb_sub.c]
660     printf
661     fprintf
662     strlen
663     fclose
664     dump_cctvdb_params [../../tms_library/fddb_lib.c]
665     dump_fddb_params [see line 280]
666     print
667     filbuf
668     print_camera_report [patch_cctvdb.c]
669     print
670     vms_c_getch [see line 540]
671     toupper
672     fopen
673     perror
674     GETTIME
675     STOP
676     NUMTIME
677     fprintf
678     sprintf
679     fclose
680     dump_cctvdb_offsets [../../tms_library/fddb_lib.c]
681     printf
682     dump_cctvdb_t1 [../../tms_library/fddb_lib.c]
683     print
684     dump_video_switch_mon_list [patch_cctvdb.c]
685     print
686     set_term_char [../../tms_library/kb_func.c]
687     QIO

```

patch_gbldb call

```
519 main [patch_gbldb.c]
520     map_to_global_section [../../../tms_library/global_sub.c]
521         strlen
522             MGBLSC
523                 printf
524                     STOP
525                         init_gbldb_t1 [../../../tms_library/fddb_lib.c]
526                             fopen
527                                 exit
528                                     get_iocchan [../../../tms_library/kb_func.c]
529                                         strlen
530                                             ASSIGN
531                                                 get_term_char [../../../tms_library/kb_func.c]
532                                                     QIO
533                                                         change_term_char [../../../tms_library/kb_func.c]
534                                                             QIO
535                                                               vms_c_getch [../../../tms_library/kb_func.c]
536                                                                 input [../../../tms_library/kb_func.c]
537                                                                     ASSIGN
538
539             QIOW
540               toupper
541                 dump_gbldb_to_table_files [gbldb_sub.c]
542                   printf
543                     translate_logical_name [../../../tms_library/logical_name.c]
544                       strlen
545                         TRNINN
546                           memmove
547                             build_full_name [../../../fddb_sub.c]
548                               strlen
549                                 memcpy
550                                   fopen
551                                     fddb_error [see line 130]
552                                       fclose
553                                         delete
554                                           sprintf
555                                             time
556                                               ctime
557                                                 fprintf
558                                                   dump_fddb_elements [../../../fddb_sub.c]
559                                         fprintf
560                                           format_db_element_for_output [../../../tms_library/format_d
561                                             b_lib.c]
562                                               sprintf
563                                                 strcpy
564                                                   strip_trailing_blanks [see line 142]
565                                                     process_output_special_case [gbldb_sub.c]
566                                                       convert_parm_mask [../../../tms_library/format_db_1
567                                                         ib.c]
568                                                       leading_zero_pad [../../../tms_library/format_d
569                                                         lib.c]
570                                             strlen
571                                               strcat
572                                                 strcmp
573                                                   NUMTRIM
574                                                     memset
575                                                       print_data_col_list [patch_gbldb.c]
576                                                         printf
577                                                           select_entry [../../../tms_library/kb_func.c]
578                                                             log10
579
580             printf
581               exit
582                 memset
583                   vms_c_getch [see line 536]
584                     longjmp
585                       atoi
586                         menu2 [patch_gbldb.c]
587                           printf
588                             vms_c_getch [see line 536]
589                               toupper
590                                 dump_gbldb_to_table_files [see line 541]
591                                   print_group_name_list [patch_gbldb.c]
592                                     memset
593                                       strlen
594                                         memcpy
595                                           printf
596                                             select_entry [see line 577]
597                                               longjmp
598                                                 menu3 [patch_gbldb.c]
599           vms_c_getch [see line 536]
600             toupper
601               print_element_name_list [patch_gbldb.c]
602                 printf
603                   select_entry [see line 577]
604                     longjmp
605                       menu4 [patch_gbldb.c]
606                         format_db_element_for_out
607                           put [see line 560]
608                             menu3
609                               vms_c_getch [see line 536]
610                                 toupper
611                                   strchr
612                                     get_param_value [../../../../tim
613                                       printf
614                                         memset
615                                           vms_c_getch [see
616                                             longjmp
617                                               process_input_special_cas
618                                                 longjmp
619                                                   output_gbldb_out_fil [gbldb_sub.c]
620                                                     fopen
621                                                       fddb_error [see line 130]
622                                                         read_fddb_file [../../../fddb_sub.c]
623                                                           get_next_line [../../../fddb_sub.c]
624                                                             ftell
625                                                               fgets
626                                                                 strlen
627                                                                   seek
628             convert_nonprint_to_space
629               strip_trailing_blanks [see line 142]
630                 strip_leading_blanks [../../../tms_library/format_db_1
631                   lib.c]
632                     memset
633                       print
634                         fddb_error [see line 130]
635                           sprint
636                             select_entry [../../../tms_library/kb_func.c]
637                               log10
638
639             strlen
640               strcat
641                 strcmp
642                   NUMTRIM
643                     memset
644                       print_data_col_list [patch_gbldb.c]
645                         printf
646                           select_entry [../../../tms_library/kb_func.c]
647                             log10
```

patch_gdb.c

```

637     exit
638     time
639     ctime
640     fprintf
641     rewind
642     fseek
643     init_group_table [see line 150]
644     print_db_table [./fddb_sub.c]
645     print_file_comments [./fddb_sub.c]
646     fseek
647     get_next_line [see line 626]
648     fprintf
649     copy_params_lines [./fddb_sub.c]
650     fseek
651     get_next_line [see line 626]
652     fprintf
653     format_db_element_for_output [see line 560]
654     print_db_table_special_case [gdbdb_sub.c]
655     print_f
656     fprintf
657     fprintf
658     strlen
659     fclose
660     dump_gdbdb_params [./tms_library/fddb_lib.c]
661     dump_fddb_params [see line 231]
662     -fillbuf
663     set_term_char [./tms_library/kb_func.c]
664     QIO

130     fddb_error [./fddb_sub.c]
131     sprintf
132     strcpy
133     printf
134     fprintf
135     build_up_arrow [./fddb_sub.c]
136     find_err_text [./fddb_sub.c]
137     sprintf
138     strlen
139     memcpy
140     _fillbuf
141     strip_trailing_blanks [./tms_library/format_db_lib.c]
142     strlen
143     init_group_table [./fddb_sub.c]
144     memset
145     memcpy

231     dump_fddb_params [./tms_library/fddb_lib.c]
232     printf
233     NUMNIM
234     STOP

440     load_param [./fddb_sub.c]
441     get_token
442     fddb_error [see line 130]
443     pad_end [./tms_library/tap_sub.c]
444     strlen
445     find_fddb_nt_name [./tms_library/fddb_lib.c]
446     memcp
447     printf
448     atoll
449     range_check [./fddb_sub.c]

```

patch_gcdb_call

```
508 main [patch_gcdb.c]
509     map_to_GCDB [...] /tms_library/fddb_lib.c]
510     map_to_global_section [...] /tms_library/global_sub.c]
511     strlen
512     NBLSC
513     init_gcdb_t1 [...] /tms_library/fddb_lib.c]
514     printf
515     SSTOP
516     map_to_VAXPORTDB [...] /tms_library/fddb_lib.c]
517     map_to_global_section [see line 510]
518     init_vaxportdb_t1 [...] /tms_library/fddb_lib.c]
519     fopen
520     exit
521     get_iocchan [...] /tms_library/kb_func.c]
522     strlen
523     ASSIGN
524     get_term_char [...] /tms_library/kb_func.c]
525     QIO
526     change_term_char [...] /tms_library/kb_func.c]
527     QIO
528     setjmp
529     vms_c_getch [...] /tms_library/kb_func.c]
530     input [...] /tms_library/kb_func.c]
531     ASSIGN
532     QIOW
533     toupper
534     dump_gcdb_to_table_files [gcdb_sub.c]
535     Printf
536     translate_logical_name [...] /tms_library/logical_name.c]
537     strlen
538     TRNINN
539     memmove
540     build_full_name [...] /fddb_sub.c]
541     strlen
542     memcpy
543     fopen
544     fddb_error [see line 130]
545     fclose
546     delete
547     sprintf
548     time
549     ctime
550     sprintf
551     dump_fddb_elements [...] /fddb_sub.c]
552     fprintf
553     format_db_element_for_output [...] /tms_library/format_db_
554     b.lib.c]
555     sprintf
556     strcpy
557     process_output_special_case [gcdb_sub.c]
558     convert_parm_mask [...] /tms_library/format_db_
ib.c]
559     strcpy
560     byte_to_floating_point [...] /tms_library/format_db_
561     lib.c]
562     strlen
563     strlen
564     strcat
565     strncat
566     NUMTRIM
```

patch_oprtvdb.c

```

516 main [patch_oprtvdb.c]
517     map_to_oprtvdb [../../tms_library/fddb_lib.c]           576
518     map_to_global_section [../../tms_library/global_sub.c]   577
519         strlen                                              578
520             MGBUSC                                           579
521                 init_oprtvdb_t1 [../../tms_library/fddb_lib.c] 580
522                     printf                                         580
523                         STOP                                     581
524                         map_to_global_section [see line 518] 582
525                         init_oprtvdb_t1                           583
526                         fopen                                         584
527                         exit                                         585
528                         get_jochan [../../tms_library/kb_func.c] 586
529                             strlen                                         587
530                             ASSIGN                                         588
531                             get_term_char [../../tms_library/kb_func.c] 589
532                                 QIO                                         590
533                                 change_term_char [../../tms_library/kb_func.c] 591
534                                     QIO                                         592
535                                         strlen                                         593
536                                         strcpy                                         594
537                                         vms_c_getchar [../../tms_library/kb_func.c] 595
538                                             input [../../tms_library/kb_func.c]    596
539                                                 ASSIGN                                         597
540                                                 QROW                                         598
541                                                 toupper                                         599
542                                                 dump_oprtvdb_to_table_files [oprtvdb_sub.c] 600
543                                                 printf                                         601
544                                                 translate_logical_name [../../tms_library/logical_name.c] 602
545                                         strlen                                         603
546                                         TANIM                                           604
547                                         memmove                                         605
548                                         build_full_name [../../fddb_sub.c]    606
549                                         strlen                                         607
550                                         strcpy                                         608
551                                         fopen                                         609
552                                         fddb_error [see line 130]          610
553                                         fclose                                         611
554                                         delete                                         612
555                                         sprintf                                         613
556                                         ctime                                         614
557                                         fprintf                                         615
558                                         dump_fddb_elements [../../fddb_sub.c] 616
559                                         fprintf                                         617
560                                         format_db_element_for_output [../../tms_library/format_d
561                                         b.lib.c]                                         618
562                                         sprintf                                         619
563                                         strcpy                                         620
564                                         strip_trailing_blanks [see line 142] 621
565                                         process_output_special_case [oprtvdb_sub.c] 622
566                                         convert_perm_mask [../../tms_library/format_db_1
567                                         lib.c]                                         623
568                                         leading_zero_pad [../../tms_library/form
at_db.lib.c]                                         624
569                                         strlen                                         625
570                                         strcat                                         626
571                                         strlen                                         627
572                                         strcpy                                         628
573                                         NUMTIN                                         629
574                                         memset                                         630
575                                         print_data_col_list [patch_oprtvdb.c] 631
576                                         longjmp                                         632
577                                         at_db.lib.c]                                         633
578                                         fgetss                                         634
579                                         fseek                                         635
580                                         convert_nonPrint_to_space 636
581                                         strip_trailing_blanks [see line 142] 637
582                                         strip_leading_blanks [../../tms_library/format_db
583                                         lib.c]                                         638

```

patch_optvdb_call

```
        setjmp
        fddb_error [see line 130]
        sprintf
        exit
        time
        ctime
        sprintf
        printf
        rewind
        fseek
        init_group_table [see line 150]
        print_db_table [./fddb_sub.c]
        print_file_comments [./fddb_sub.c]
        fseek
        get_next_line [see line 626]
        fprintf
        copy_param_lines [./fddb_sub.c]
        fseek
        get_next_line [see line 626]
        fprintf
        format_db_element_for_output [see line 560]
        print_db_table_special_case [optrvdb_sub.c]
        printf
        fprintf
        strlen
        fclose
        dump_optvdb_params [././tms_library/fddb_lib.c]
        dump_fddb_params [see line 228]
        _filbuf
        print_opr_report [patch_optvdb.c]
        print
        vms_c_getch [see line 536]
        toupper
        fopen
        perror
        GETTIME
        STOP
        NUMTIME
        fprintf
        sprintf
        convert_status [././tms_library/fddb_lib.c]
        strcpy
        memcpy
        fclose
        set_term_char [././tms_library/kb_func.c]
        QIO
        fddb_error [./fddb_sub.c]
        sprintf
        strcpy
        printf
        build_up_arrow [./fddb_sub.c]
        find_err_text [./fddb_sub.c]
        sprintf
        strlen
        memcpy
        _filbuf
        strip_trailing_blanks [././tms_library/format_db_lib.c]
        load_param [./fddb_sub.c]
        get_token
        437
        143
        438
```

patch_rmdb call

```

.395 main [patch_rmdb.c]
.396     map_to_RMDB [../../tms_library/fddb.lib.c]
.397         map_to_global_section [../../tms_library/global_sub.c]
.398             strlen
.399                 MBMSC
.400                     init_rmdb_t1 [../../tms_library/fddb.lib.c]
.401                         printf
.402                             STOP
.403                                 map_to_VAXPORTDB [../../tms_library/fddb.lib.c]
.404                                     map_to_global_section [see line 397]
.405                                         init_vaxportdb_t1 [../../tms_library/fddb.lib.c]
.406                                         compare_VAX_time
.407                                         exit
.408                                         connect_to_mailbox [see line 327]
.409                                             fopen
.410                                                 get_iocchan [../../tms_library/kb_func.c]
.411                                                     strlen
.412                                                       ASSIGN
.413                                                         get_term_char [../../tms_library/kb_func.c]
.414                                                             QIO
.415                                                 change_term_char [../../tms_library/kb_func.c]
.416                                                     QIO
.417                                                     setjmp
.418                                                         check_rmdb_changed [patch_rmdb.c]
.419                                                             prompt_for_yes_no [../../tms_library/kb_func.c]
.420                                                               strlen
.421                                                               printf
.422                                                               vms_c_getch [../../tms_library/kb_func.c]
.423                                                               input [../../tms_library/format_db.lib.c]
.424                                                               ASSIGN
.425                                                               QIOW
.426                                                               toupper
.427                                                               mail_patch_cmd_to_rmdc_comm [patch_rmdb.c]
.428                                                               printf
.429                                                               write_to_mailbox_nowait [see line 100]
.430                                                               vms_c_getch [see line 422]
.431                                                               toupper
.432                                                               dump_rmdb_to_table_files
.433                                                               print_data_col_list [patch_rmdb.c]
.434                                                               printf
.435                                                               select_entry [../../tms_library/kb_func.c]
.436                                                               log10
.437                                                               printf
.438                                                               exit
.439                                                               memset
.440                                                               vms_c_getch [see line 422]
.441                                                               longjmp
.442                                                               atoi
.443                                                               longjmp
.444                                                               menu2 [patch_rmdb.c]
.445                                                               printf
.446                                                               vms_c_getch [see line 422]
.447                                                               toupper
.448                                                               dump_rmdb_to_table_files
.449                                                               print_group_name_list [patch_rmdb.c]
.450                                                               memset
.451                                                               strlen
.452                                                               memcpy
.453                                                               printf
.454                                                               select_entry [see line 435]
.455                                                               longjmp
.456                                                               menu3 [patch_rmdb.c]
.457                                                               printf
.458                                                               vms_c_getch [see line 422]

toupper
disnable_enable_loop [patch_rmdb.c]
printf
select_entry [see line 435]
longjmp
memcpy
prompt_for_yes_no [see line 419]
filbuf
printf
select_entry [see line 435]
longjmp
menu4 [patch_rmdb.c]
format_db_element_for_out
sprintf
strcpy
strip_trailing_b1
strlen
process_output_sp
convert_perm_mask
ans [../../tms_library/format_db.lib.c]
put [../../tms_library/format_db.lib.c]
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510

toupper
map_to_RMDB [patch_rmdb.c]
map_to_global_section [see line 397]
init_vaxportdb_t1 [../../tms_library/fddb.lib.c]
compare_VAX_time
exit
connect_to_mailbox [see line 327]
fopen
get_iocchan [../../tms_library/kb_func.c]
strlen
convert_perm_mask
ans [../../tms_library/format_db.lib.c]
put [../../tms_library/format_db.lib.c]
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510

process_output_sp
convert_perm_mask
ans [../../tms_library/format_db.lib.c]
put [../../tms_library/format_db.lib.c]
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510

process_input_special_cas
load_param
check_rmdb_changed [see line 1]
longjmp
dump_loop_table_to_file
dump_speed_traps_to_file
longjmp
mail_patch_cmd_to_rmdc_comm [see line 427]
filbuf
output_rmdb_out_file
dump_rmdb_params [../../tms_library/fddb.lib.c]
dump_fddb_params [see line 171]
printf

```

patch_rmdb call

```

511         filbuf
512         _print_rmdc_report [patch_rmdc.c]
513         printf
514             vms_c_getch [see line 422]
515             toupper
516             fopen
517             perror
518             GETTIME
519             STOP
520             NUMTIME
521             fprintf
522             sprintf
523             convert_status [../../tms_library/fddb_lib.c]
524             strcpy
525             memcpy
526             fclose
527             dump_rmdb_offsets [../../tms_library/fddb_lib.c]
528             printf
529             dump_rmdb_tl [../../tms_library/fddb_lib.c]
530             printf
531             dump_fddb_name_table [../../tms_library/fddb_lib.c]
532             printf
533             dump_fddb_col_list [../../tms_library/fddb_lib.c]
534             printf
535             set_term_char [../../tms_library/kb_func.c]
536             QIO

                                write_to_mailbox_nowait [../../tms_library/mailbox.c]
100
101             QIOW

171             dump_fddb_params [../../tms_library/fddb_lib.c]
172             printf
173             NUMTIME
174             STOP

237             connect_to_mailbox [../../tms_library/mailbox.c]
328             strlen
329             ASSIGN

```

06/05/14
12:21:15

patch_vaxportdb.c

```
548 main [patch_vaxportdb.c]
549     map_to_RMDB [...] /tms_library/fddb.lib.c]
550     map_to_global_section [...] /tms_library/global_sub.c]
551     strlen
552     MGBLSC
553     init_rmdb_t1 [...] /tms_library/fddb.lib.c]
554     printf
555     STOP
556     map_to_GBLDS [...] /tms_library/fddb.lib.c]
557     map_to_global_section [see line 550]
558     init_gbldb_t1 [...] /tms_library/fddb.lib.c]
559     map_to_VMSDB [...] /tms_library/fddb.lib.c]
560     map_to_global_section [see line 550]
561     init_vmsdb_t1 [...] /tms_library/fddb.lib.c]
562     map_to_OCTVNPB [...] /tms_library/fddb.lib.c]
563     map_to_global_section [see line 550]
564     init_octvnpb_t1 [...] /tms_library/fddb.lib.c]
565     map_to_OPRVNPB [...] /tms_library/fddb.lib.c]
566     map_to_global_section [see line 550]
567     init_oprvnpb_t1 [...] /tms_library/fddb.lib.c]
568     map_to_VAXPORTDB [...] /tms_library/fddb.lib.c]
569     map_to_global_section [see line 550]
570     init_vaxportdb_t1 [...] /tms_library/fddb.lib.c]
571     map_to_GCDB [...] /tms_library/fddb.lib.c]
572     map_to_global_section [see line 550]
573     init_gcdb_t1 [...] /tms_library/fddb.lib.c]
574     fopen
575     perror
576     exit
577     get_iocchan [...] /tms_library/kb_func.c]
578     strlen
579     ASSIGN
580     get_term_char [...] /tms_library/kb_func.c]
581     QIO
582     change_term_char [...] /tms_library/kb_func.c]
583     QIO
584     setjmp
585     vms_c_getch [see line 481]
586     toupper
587     print_VAXPort_list [vaxportdb_sub.c]
588     print_VAXPort_flag [patch_vaxportdb_sub.c]
589     select_entry [...] /tms_library/kb_func.c]
590     log10
591     printf
592     exit
593     memset
594     vms_c_getch [see line 481]
595     longjmp
596     atoi
597     longjmp
598     change_VAXPort_flag [patch_vaxportdb.c]
599     prompt_for_yes_no [...] /tms_library/kb_func.c]
600     convert_VAXPort_flag [vaxportdb_sub.c]
601     strcpy
602     vms_c_getch [see line 481]
603     toupper
604     prompt_for_yes_no [...] /tms_library/kb_func.c]
605     strlent
606     printf
607     vms_c_getch [see line 481]
608     bit_clr_i
609     longjmp
610     port_control [patch_vaxportdb.c]
611     convert_DB_type [see line 642]
612     convert_status [see line 613]
613     strcpy
614     memory
615     vms_c_getch [see line 481]
616     toupper
617     prompt_for_yes_no [see line 604]
618     memset
619     longjmp
620     print_UnitList [patch_vaxportdb_sub.c]
621     printf
622     sprint
623     select_entry [see line 589]
624     longjmp
625     unit_control [patch_vaxportdb.c]
626     printf
627     convert_status [see line 613]
628     convert_start_flag [vaxportdb_sub.c]
629     vms_c_getch [see line 481]
630     toupper
631     prompt_for_yes_no [see line 604]
632     memset
633     longjmp
634     dump_vaxportdb_params [...] /tms_library/fddb.lib.c]
635     printf
636     STOP
637     NUMTIN
638     _filbuf
639     dump_VAXPortPdStats [vaxportdb_sub.c]
640     printf
641     convert_DB_type [vaxportdb_sub.c]
642     strcpy
643     select_DB_for_UnitList [patch_vaxportdb.c]
644     printf
645     vms_c_getch [see line 481]
646     toupper
647     longjmp
648     dump_UnitPdStats [vaxportdb_sub.c]
649     printf
650     sprint
651     dump_UnitTable [vaxportdb_sub.c]
652     printf
653     convert_status [see line 613]
654     convert_start_flag
655     strcmp
656     sprintf
657     dump_VAXPortTable [vaxportdb_sub.c]
658     printf
659     convert_DB_type [see line 642]
660     convert_VAXPort_flag [see line 600]
661     convert_VAXPort_Status [see line 613]
662     print_VAXPortStatus [vaxportdb_sub.c]
663     printf
664     vms_c_getch [see line 481]
665     toupper
666     fopen
667     perror
668     GETTIME
669     STOP
670     NUMTIN
671     fprintf
672     sprint
673     convert_DB_type [see line 642]
674     convert_status [see line 613]
```

patch vaxportdb:call

```
676      fclose [../../tms_library/kb_func.c]
677      set_termchar [../../tms_library/kb_func.c]
678
478      get_param_value [../../tms_library/kb_func.c]
479      printf
480      memset
481      vms_c_getch [../../tms_library/kb_func.c]
482      input [../../tms_library/kb_func.c]
483      ASSIGN
484      QIOW
485      longjmp
```

patch_vmsdb.c

```
64         fddb_error [./fddb_sub.c]
65             sprintf
66                 strcpy
67                     printf
68                         fprintf
69                             build_up_arrow [./fddb_sub.c]
70                                 find_err_text [./fddb_sub.c]
71                                     sprintf
72                                         strlen
73                                             memcpy
74                                                 _fillbuf
75
76             strip_trailing_blanks [../../tms_library/format_db_lib.c]
77                 strlen
78             init_group_table [./fddb_sub.c]
79                 memset
80                     memcpy
81
82             crack_vms_message [../../tms_library/vms_lib.c]
83                 init_vms_msg_struct [../../tms_library/vms_lib.c]
84
85             memset
86
87             copy_flash [../../tms_library/vms_lib.c]
88
89             build_FP_static_msg [../../tms_library/vms_lib.c]
90                 build_FP_msg_header [../../tms_library/vms_
91                     ms_lib.c]
92
93             build_FP_multiphase_msg [../../tms_library/vms_
94                     lib.c]
95                 build_FP_msg_header
96                     strlen
97                         toupper
98
99             build_FP_right_arrow_to_buffer [../../tms_library/vms_
100                     ms_lib.c]
101
102             build_FP_arrow_msg [../../tms_library/vms_lib.c]
103
104             build_FP_msg_header
105                     strlen
106                         toupper
107                             FP_right_arrow_to_buffer
108
109             find_vmsdb_entry_by_name [../../tms_library/vms_
110                     ms_lib.c]
111
112             memcmp
113
114             dump_fddb_params [../../tms_library/fddb_lib.c]
115                 printf
116                     _try/vms_ms.lib.c]
117
118             dump_mem [../../tms_library/dump_mem.c]
119                 print_hex_ascii_line [../../tms_library/dump_mem.c]
120
121             main [patch_vmsdb.c]
122                 map_to_vmsdb [../../tms_library/fddb_lib.c]
123                     map_to_global_section [../../tms_library/global_sub.c]
124                         strlen
125                             MGBMSC
126
127             init_vmsdb_t1 [../../tms_library/fddb_lib.c]
128
129             printf
130                 STOP
131                     get_iocchan [../../tms_library/kb_func.c]
132                         strlen
133                             ASSIGN
134
135             set_port_partial [../../tms_library/kb_func.c]
136
137             exit
138
139             get_term_char [../../tms_library/kb_func.c]
140                 QIO
141
142             set_port_change_term [../../tms_library/kb_func.c]
143                 QIO
144
145             setjmp
146                 vms_c_getch [../../tms_library/kb_func.c]
147
148             input [../../tms_library/kb_func.c]
149
150             toupper
151
152             cluster_menu [patch_vmsdb.c]
153
154             select_entry [../../tms_library/kb_func.c]
155
156             log10
157
158             print_vmsdb_entry_by_index [patch_vmsdb.c]
159
160             print
161                 dump_mem [see line 285]
162                     crack_vms_message [see line 131]
163                     dump_vms_msg_struct [vmsdb_sub.c]
164
165             print_vms_struct_field [vmsdb_sub.c]
166                 print_vms_ms.lib.c]
167                     print_vms_ms_type [vmsdb_sub.c]
168
169             print_vms_auxout [../../tms_library/vms_
170                     1]
171
172             print_fp_static_msg [see line 135]
173                 print_crack_fp_ms [../../tms_library/crack_fp_ms.c]
174
175             lib.c]
176
177             stop
178
179             print_hex_ascii_line [../../tms_library/crack_fp_ms.c]
180
181
182             build_fp_static_msg [see line 135]
183                 crack_fp_ms [../../tms_library/crack_fp_ms.c]
```

patch_vmsdb.call

```

06/05/15

720   sprintf
721     dump_fddb_elements [./fddb_sub.c]
722   fprintf
723     format_db_element_for_output [./tms_library/format_db_db
724       _lib.c]
725     vms_c_getch [see line 681]
726     toupper
727     build_fp_multiphase_msg [see line 145]
728     build_fp_arrow_msg [see line 149]
729     print_vmsdb_message_entry [./tms_library/vms
730       _lib.c]
731     fprintf
732       dump_mem [see line 285]
733     print_vms_auxout [see line 714]
734     print_vms_line [./tms_library/vms_li
735       t_db_lib.c]
736     b.c]
737     fprintf
738     print_vms_time [./tms_library/vms_li
739       _lib.c]
740     fprintf
741     print_vms_repeats [./tms_library/vms
742       _lib.c]
743     fprintf
744     dump_mem [see line 285]
745     strlen
746     print_vmsdb_library_entry [./tms_library/vms
747       _lib.c]
748     print_vmsdb_queue_entry [./tms_library/vms
749       _lib.c]
750     dump_mem [see line 285]
751     print_vms_time [see line 736]
752     strlen
753     print_vmsdb_cluster_entry [./tms_library/vms
754       _lib.c]
755     print_vmsdb_full_name [./fddb_sub.c]
756     build_full_name [./fddb_sub.c]
757     strlen
758     memmove
759     fopen
760     dump_vmsdb_to_table_files [vmsdb_sub.c]
761     printf
762     translate_logical_name [./tms_library/logical_name.c]
763     strlen
764     TRLNAME
765     memmove
766     build_full_name [./fddb_sub.c]
767     strlen
768     memcopy
769     fopen
770     fddb_error [see line 64]
771     fclose
772     delete
773     printf
774     get_param_value [./tm
775     s_library/kb_func.c]
776     ctime
777     puts
778     vms_c_getch [see line 681]
779     toupper
780     strchr
781     memset
782     printf
783     strcpy
784     strip_trailing_blanks [see line 76]
785     process_output_special_case [vmsdb_sub.c]
786     convert_perm_mask [./tms_library/format_db_db
787       _lib.c]
788     strcpy
789     byte_to_float [./tms_library/format_db_db
790       _lib.c]
791     leading_zero_pad [./tms_library/format
792       _db_db.lib.c]
793     memset
794     follow_vmsdb_list [see line 690]
795     print_data_col_list [patch_vmsdb.c]
796     filbuf
797     printf
798     select_entry [see line 695]
799     menu2 [patch_vmsdb.c]
800     printf
801     vms_c_getch [see line 681]
802     toupper
803     dump_vmsdb_to_table_files [see line 761]
804     print_group_name_list [patch_vmsdb.c]
805     memset
806     strlen
807     memcpy
808     printf
809     select_entry [see line 695]
810     menu3 [patch_vmsdb.c]
811     printf
812     vms_c_getch [see line 681]
813     toupper
814     print_vmsdb_list_names [see line 693]
815     memset
816     gets
817     strlen
818     memcpy
819     dump_mem [see line 285]
820     printf
821     follow_vmsdb_list [see line 690]
822     print_element_name_list [patch_vmsdb.c]
823     printf
824     select_entry [see line 695]
825     menu4 [patch_vmsdb.c]
826     printf
827     format_db_element_for_out
828     vms_c_getch [see line 681]
829     ]
830     toupper
831     strchr
832     get_param_value [./tm
833     s_library/kb_func.c]
834     ctime

```

Patch_vmsdb.call

```

835     vms_c_getch [see
Line 681]           longjmp
836                                     perror
837                                     GETRM
838                                     STOP
se [see line 467]           process_input_special_c
                                     NUMTRN
                                     fprintf
                                     sprintf
                                     convert_status [.../tms_library/fddb_lib.c]
                                     strncpy
                                     memcp
                                     longjmp
                                     fclose
                                     dump_vmsdb_offsets [.../tms_library/fddb_lib.c]
                                     printf
                                     dump_vmsdb_bt1 [.../tms_library/fddb.lib.c]
                                     printf
                                     set_term_char [.../tms_library/kb_func.c]
                                     QIO
                                     get_next_line [...]/fddb_sub.c]
                                     ftell:
                                     fgets
                                     strlen
                                     fstrlen
                                     fseek
                                     convert_nonprint_to_space
                                     strip_trailing_blanks [see line 76]
                                     strip_leading_blanks [.../tms_library/Format_d
b.lib.c]
                                     setjmp
                                     fddb_error [see line 64]
                                     sprintf
                                     exit
                                     time
                                     crtime
                                     fprintf
                                     printf
                                     rewind
                                     fseek
                                     init_group_table [see line 84]
                                     print_file_comments [...]/fddb_sub.c]
                                     fseek
                                     get_next_line [see line 847]
                                     fprintf
                                     print_vmsdb_cluster_entry [see line 755]
                                     print_db_table [...]/fddb_sub.c]
                                     print_file_comments [see line 866]
                                     copy_param_lines [...]/fddb_sub.c]
                                     fseek
                                     get_next_line [see line 847]
                                     fprintf
                                     format_db_element_for_output [see line 780]
                                     print_db_table_special_case [vmsdb_sub.c]
                                     fprintf
                                     print_vmsdb_message_entry [see line 729]
                                     print_vmsdb_library_entry [see line 743]
                                     print_vmsdb_queue_entry [see line 748]
                                     printf
                                     fprintf
                                     strlen
                                     fclose
                                     dump_vmsdb_params [...]/tms_library/fddb.lib.c]
                                     dump_fddb_params [see line 264]
                                     print
                                     print_vmsdb_report [patch_vmsdb.c]
                                     print
                                     vms_c_getch [see line 681]
                                     toupper
                                     fopen
                                     
```

read_rtdb.c

152
112 47-36

```
439 main [read_rtdb.c]
440     printf
441     ASCFC
442     STOP
443     TWO_bit_mask [./tms_library/skel_sub.c]
444     CLREF
445     WFLOR
446     READEP
447     exit
448     GETTIM
449     NONTIM
```

reset_modem.c

```
466 main [reset_modem.c]
467     printf
468     get_port_name [reset_modem.c]
469         printf
470             scanf
471                 strl_en
472                     toupper
473                         flush_input [./tms_library/kb_func.c]
474                             _filbuf
475                                 strlen
476                                     ASSIGN
477                                         STOP
478                                             GETDVI
479                                                 prompt_for_yes_no [./tms_library/kb_func.c]
480                                                     strlen
481                                                         printf
482                                                             vms_c_getch [ see line 415 ]
483                                                               toupper
484         exit
485             vms_c_getch [ see line 415 ]
486             toupper
487             talk_through [reset_modem.c]
488             printf
489             gets
490             strlen
491             toupper
492             HM_command [reset_modem.c]
493                 send_HM_cmd [ see line 2 ]
494                     printf
495                         convert_HM_result [ see line 13 ]
496                             wait_HM_result [ see line 6 ]
497             HM_view [reset_modem.c]
498             memset
499             printf
500             send_HM_cmd [ see line 2 ]
501             convert_HM_result [ see line 13 ]
502             exit
503             get_HM_input [ see line 9 ]
504             memcpy
505             _filbuf

2 2 send_HM_Cmd [reset_modem.c]
3 3         strlen
4 4             QIOW
5 5             printf
6 6         wait_HM_result [reset_modem.c]
7 7             GETTIM
8 8             printf
9 9             get_HM_input [reset_modem.c]
10 10             sprintf
11 11                 printf
12 12                     QIOW
13 13             convert_HM_result [reset_modem.c]
14 14             strcpy
15 15             memset
16 16             memcpy
17 17             msleep [./tms_library/misc_func.c]
18 18             SCHDWK
19 19             log_tms_event_cc [./tms_library/event_log_sub.c]
20 20             log_tms_common [./tms_library/event_log_sub.c]
21 21             write_to_crash_log [./tms_library/proc_
22 22             write_to_mailbox_nowait [./tms_library/m
```

```

151 main [rmdc_comm.c]
152   fopen
153   Printf
154   exit
155   fclose
156   check_process_interactive
157   write_to_crash_log
158   STOP
159   create_comm_log_file [see line 31]
160   log_comm_event_cc [see line 65]
161   log_comm_event [see line 5]
162   log_comm_process_name [./tms_comm_sub.c]
163   general_process_startup
164   establish_process_name
165   init_rmdc_event_flags
166   restore_comm_process_name [see line 5]
167   strlen
168   set_process_name
169   log_comm_msg [see line 75]
170   log_comm_event_cc [see line 65]
171   get_terminal_name
172   log_comm_event [see line 5]
173   get_Did
174   sprintf
175   init_indriver [./tms_comm_sub.c]
176   strlen
177   CMKRN
178   PRINTF
179   log_comm_event_cc [see line 65]
180   write_to_crash_log
181   restore_comm_process_name [see line 166]
182   STOP
183   map_to_RMDB
184   map_to_RMDB
185   map_to_VAXPORTDB
186   map_to_VAXPORTDB
187   compare_VAX_time
188   find_first_last_port_unit
189   init_rmdc_list_heads
190   get_proc_info
191   CLREF
192   get_jochan
193   get_berm_char
194   change_term_char
195   connect_to_mailbox
196   flush_mailbox
197   alloc_and_queue_empty_jhub [see line 59]
198   start_rmdc_port [rmdc_comm_sub.c]
199   sprintf
200   log_comm_event [see line 5]
201   bit_clr_i
202   setup_port_for_protocol [./tms_comm_sub.c]
203   memcomp
204   strlen
205   ASSIGN
206   log_comm_event_cc [see line 65]
207   CMKRN
208   log_comm_event [see line 5]
209   sprintf
210   QIOW
211   log_comm_event_cc [see line 65]
212   DASGN
213   set_JH_speed_parity [./tms_comm_sub.c]
214   strcpy

215   QIOW
216   turn_on_JH_RTS [./tms_comm_sub.c]
217   QIOW
218   turn_on_JH_DTR [./tms_comm_sub.c]
219   QIOW
220   log_comm_event [see line 5]
221   init_rmdc_gc_unit [rmdc_comm_sub.c]
222   init_gc_unit [rmdc_comm_sub.c]
223   log_comm_event [see line 5]
224   init_rmdc_unit [rmdc_comm_sub.c]
225   log_comm_event [see line 5]
226   build_and_queue_170_msg [rmdc_comm_sub.c]
227   bit_set_i
228   set_unit_status [rmdc_comm_sub.c]
229   send_rmdc_sc_start_msgs [rmdc_comm_sub.c]
230   send_gc_start_msgs [rmdc_comm_sub.c]
231   log_comm_event [see line 5]
232   build_and_queue_170_msg [rmdc_comm_sub.c]
233   sprintf
234   log_comm_event [see line 5]
235   get_rmdc_buffer [see line 55]
236   init_tx_rx_jhub [./tms_comm_sub.c]
237   sprintf
238   log_comm_event [see line 5]
239   memset
240   add_to_list_tail_i
241   SETEF
242   log_comm_event_cc [see line 65]
243   build_and_queue_170_date_time [rmdc_comm_sub.c]
244   sprintf
245   log_comm_event [see line 5]
246   get_rmdc_buffer [see line 55]
247   init_tx_rx_jhub [see line 236]
248   add_to_list_tail_i
249   SETEF
250   log_comm_event_cc [see line 65]
251   log_comm_msg [see line 75]
252   log_driver_stats [./tms_comm_sub.c]
253   log_comm_msg [see line 75]
254   sprintf
255   simple_write_to_comm_log [see line 130]
256   FLUSH
257   printf
258   write_to_crash_log
259   log_24Hr_port_stats [./tms_comm_sub.c]
260   log_comm_msg [see line 75]
261   log_24Hr_unit_stats [./tms_comm_sub.c]
262   log_driver_stats
263   log_comm_msg [see line 75]
264   sprintf
265   send_rmdc_start_msgs [rmdc_comm_sub.c]
266   log_comm_event [see line 5]
267   log_comm_msg [see line 5]
268   build_and_queue_170_msg [see line 232]
269   build_and_queue_170_date_time [see line 243]
270   build_and_queue_load_params [rmdc_comm_sub.c]
271   log_comm_event [see line 5]
272   sprintf
273   get_rmdc_buffer [see line 55]
274   init_tx_rx_jhub [see line 236]
275   add_to_list_tail_i
276   SETEF
277   log_comm_event_cc [see line 65]
278   log_comm_msg [see line 75]

```

```

279 log_driver_stats [see line 252]
280 log_24hr_port_stats [see line 259]
281 log_24hr_unit_stats [see line 262]
282 sprintf
283 queued_get_1_char
284 -fillbuf
285 queued_read_from_mailbox
286 WFLOR
287 READDEF
288 write_string_to_mailbox
289 exit_multi_rmdc_comm [rmdc_comm_sub.c]
290 stop_rmdc_port [rmdc_comm_sub.c]
291 log_comm_event [see line 5]
292 CANCEL
293 log_comm_event_cc [see line 65]
294 DASSGN
295 remove_from_list_head_i
296 return_to_free_list
297 stop_rmdc_gc_unit [rmdc_comm_sub.c]
298 stop_gc_unit [rmdc_comm_sub.c]
299 log_comm_event [see line 5]
300 stop_rmdc_unit [rmdc_comm_sub.c]
301 log_comm_event [see line 5]
302 bit_clr_i
303 log_comm_event [see line 5]
304 log_comm_event_cc [see line 65]
305 set_term_char
306 printf
307 restore_comm_process_name [see line 166]
308 close_comm_log_file [see line 22]
309 exit
310 exit_rmdc_comm [rmdc_comm_sub.c]
311 stop_rmdc_port [see line 290]
312 printf
313 prompt_for_yes_no
314 prompt_for_unit_enable [rmdc_comm_sub.c]
315 format_unit_name [.. /tms_comm_sub.c]
316 sprintf
317 printf
318 prompt_for_yes_no
319 printf
320 set_term_char
321 restore_comm_process_name [see line 166]
322 close_comm_log_file [see line 22]
323 exit
324 multi_menu [rmdc_comm_sub.c]
325 CANCEL
326 printf
327 CLRBF
328 timed_wait_for_1_char
329 toupper
330 FLUSH
331 prompt_for_yes_no
332 exit_multi_rmdc_comm [see line 289]
333 process_mailbox_command [rmdc_comm_sub.c]
334 sprintf
335 log_comm_event [see line 5]
336 build_and_queue_170_msg [see line 232]
337 build_and_queue_load_params [see line 270]
338 remove_from_list_head_i
339 SERVER
340 log_comm_bfr [.. /tms_comm_sub.c]
341 get_mpu_name [see line 6]
342 sprintf
343 write_to_crash_log
344 log_comm_msg [see line 75]
345 GETINT
346 PRINTF
347 NUMTIN
348 sprintf
349 fmt_write_comm_msg [see line 21]
350 simple_write_to_comm_log [see line 130]
351 format_ascii_hex [.. /tms_comm_sub.c]
352 memset
353 memory
354 memset
355 crack_fp_msg
356 FLUSH
357 return_to_free_list
358 process_rmrdc_tx_wait_rx_errors [rmrdc_comm_sub.c]
359 check_rmrdc_port_status [rmrdc_comm_sub.c]
360 log_comm_event [see line 5]
361 return_to_free_list
362 log_comm_msg [see line 75]
363 log_comm_event [see line 5]
364 log_comm_event_cc [see line 65]
365 log_iosb_fields [.. /tms_comm_sub.c]
366 sprintf
367 simple_write_to_comm_log [see line 130]
368 FLUSH
369 printf
370 write_to_crash_log
371 memset
372 add_to_list_tail_i
373 SETTER
374 memcmpp
375 process_rmrdc_good_comm [rmrdc_comm_sub.c]
376 log_comm_event [see line 5]
377 check_rmrdc_port_status [see line 359]
378 sprintf
379 process_data_response [rmrdc_comm_sub.c]
380 log_comm_event [see line 5]
381 sprintf
382 unpack_170_volt_occ [rmrdc_comm_sub.c]
383 get_nibble [rmrdc_comm_sub.c]
384 pack_rtdb_loop
385 memcpy
386 log_170_data_roll [rmrdc_comm_sub.c]
387 log_comm_event [see line 5]
388 log_170_commcmsg [see line 75]
389 sprintf
390 simple_write_to_comm_log [see line 130]
391 unpack_170_speed_trap [rmrdc_comm_sub.c]
392 memcpy
393 strcpy
394 FLUSH
395 printf
396 write_to_crash_log
397 process_gc_status [rmrdc_comm_sub.c]
398 log_comm_event [see line 5]
399 CANTIN
400 process_rmrdc_err_resp [rmrdc_comm_sub.c]
401 log_comm_event [see line 5]
402 GETIN
403 log_comm_event_cc [see line 65]
404 log_170_error_response [rmrdc_comm_sub.c]
405 log_comm_msg [see line 75]
406 sprintf

```

Rmde_comm_call

```

407 simple_write_to_comm_log [see line 130]
408 FLUSH
409 printf
410 write_to_crash_log
411 process_gc_err_resp [rmdc_comm_sub.c]
412 log_comm_event [see line 5]
GETTIM
413 log_comm_event_cc [see line 65]
414 build_and_queue_170_msg [see line 232]
415 add_to_list_head_i
416 fill_in_date_time [rmdc_comm_sub.c]
GETRIM
417 log_comm_event_cc [see line 65]
418 ADD_TIMES
419 calc_check [...]/tms_comm_sub.c]
420 NUMTIN
421 DAY_OF_WEEK
422 circ_rmdc_tx_msg [rmdc_comm_sub.c]
423 calc_check [...]/tms_comm_sub.c]
424 OTO
425 GETTIM
426 NUMTIN
427 log_comm_msg [see line 75]
428 check_rmdc_port_status [see line 359]
429 scroll_port_stats [...]/tms_comm_sub.c]
430 scroll_unit_stats [...]/tms_comm_sub.c]
431
432
433 log_comm_msg [see line 75]
434 check_ERRREQ_bit [rmdc_comm_sub.c]
435 log_comm_event [see line 5]
436 FLUSH
437 stop_rmdc_port [see line 290]
438 init_rmdc_gc_unit [see line 222]
439 stop_rmdc_gc_unit [see line 298]
440 ADD_TIMES
441 SETTMR
442 build_and_queue_170_date_time [see line 243]
443 read_JH_modem_status [...]/tms_comm_sub.c]
444 QION
445 test_menu [rmdc_comm_sub.c]
446 CANCEL
447 printf
448 auto_menu [rmdc_comm_sub.c]
449 format_unit_name [see line 316]
450 CANCEL
451 printf
452 CLRREF
453 stop_rmdc_port [see line 290]
454 format_unit_name [see line 222]
455 vms_c_getch
456 flush_rmdc_queues [rmdc_comm_sub.c]
457 toupper
458 remove_from_list_head_i
459 print
460 return_to_free_list
461 CLRREF
462 init_rmdc_event_flags
463 init_rmdc_gc_unit [see line 298]
464 get_unit_status
465 send_rmdc_gc_start_msgs [see line 229]
466 stop_rmdc_gc_unit [see line 298]
467 test_unit_select [rmdc_comm_sub.c]
468 flush_rmdc_queues [see line 457]
469 stop_rmdc_gc_unit [see line 298]
470 print_unit_list [...]/tms_comm_sub.c]
471 format_unit_name [see line 316]
472 _fillbuf
473 select_entry
474 prompt_for_unit_enable [see line 315]
475 init_rmdc_event_flags
476 init_rmdc_gc_unit [see line 222]
477 manual_menu [rmdc_comm_sub.c]
478 CANCEL
479 print
480 CLRREF
481 stop_rmdc_port [see line 290]
482 format_unit_name [see line 316]
483 vms_c_getch
484 toupper
485 flush_rmdc_queues [see line 457]
486 test_unit_select [see line 467]
487 stop_rmdc_gc_unit [see line 467]
488 build_and_queue_170_msgs [see line 232]
489 build_and_queue_170_date_time [see line 243]
490 send_rmdc_gc_start_msgs [see line 229]
491 build_and_queue_load_params [see line 270]
492 format_unit_name [see line 316]
493 get_unit_status
494 vms_c_getch
495 toupper
496 print_port_list [...]/tms_comm_sub.c]
497 print
498 start_rmdc_port [see line 198]
499 _fillbuf
500 select_entry
501 sleep
502 restore_comm_process_name [see line 166]
503 exit
504 start_rmdc_port [see line 198]
505 _fillbuf
506 FLUSH
507 exit_test_rmdc_comm [see line 311]
508 enable_disable_unit [rmdc_comm_sub.c]
509 prompt_for_yes_no
510 flush_rmdc_queues [see line 457]
511 stop_rmdc_port [see line 290]
512 prompt_for_unit_enable [see line 315]
513 test_unit_select [see line 467]
514 set_term_char
515 close_comm_log_file [see line 22]
516 turn_off_JH_DTR [...]/tms_comm_sub.c]
517 QIOW
518 turn_off_JH_RTS [...]/tms_comm_sub.c]
519 QIOW
520 tx_done_rx_ast_func [rmdc_comm_sub.c]
521 add_to_list_tail_i
522 SETTER
523 log_comm_event_cc [see line 65]
524
1 _align [rmdc_comm_sub.c]
2 init_empty_jhub [...]/tms_comm_sub.c]
3 memset

```

Module Comm call

```

4   add_to_list_tail_i
5   log_comm_event [./tms_comm_sub.c]
6   get_mpu_name [./tms_comm_sub.c]
7   memset
8   strlen
9   memcpy
10  get_port_name [./tms_comm_sub.c]
11  memset
12  strlen
13  strcpy
14  get_unit_name [./tms_comm_sub.c]
15  memset
16  strcpy
17  strlen
18  memcpy
19  write_to_crash_log
20  fmt_write_comm_msg [./tms_comm_sub.c]
21  close_comm_log_file [./tms_comm_sub.c]
22  GETTIM
23  NUMTIM
24  strcpy
25  strlen
26  PUT
27  sprintf
28  FLUSH
29  CLOSE
30  create_comm_log_file [./tms_comm_sub.c]
31  GETTIM
32  NUMTIM
33  strcpy
34  strcat
35  strlen
36  CREATE
37  CONNECT
38  sprintf
39  PUT
40  FLUSH
41  GETTIM
42  NUMTIM
43  strcpy
44  sprintf
45  strlen
46  memcpy
47  PUT
48  printf
49  FLUSH
50  write_to_crash_log
51  sprintf
52  free
53  free

54  attn_ast_func [rmdc_comm_sub.c]
55  get_rmdc_buffer [rmdc_comm_sub.c]
56  sprintf
57  log_comm_event [see line 5]
58  remove_from_list_head_i
59  alloc_and_queue_empty_jhub [./tms_comm_sub.c]
60  log_comm_event [see line 5]
61  init_empty_jhub [see line 2]
62  return_to_free_list
63  log_comm_event [see line 5]
64  log_comm_event_cc [./tms_comm_sub.c]
65  get_mpu_name [see line 6]
66  memset
67  write_to_crash_log
68  fmt_write_comm_msg [see line 21]
69  init_attn_ast_jhub [./tms_comm_sub.c]
70  memset
71  add_to_list_tail_i
72  SENDF
73  check_JHUB [./tms_comm_sub.c]
74  sprintf
75  log_comm_msg [./tms_comm_sub.c]
76  get_mpu_name [see line 6]
77  write_to_crash_log
78  fmt_write_comm_msg [see line 21]
79  free

80  check_vpt_name [./tms_comm_sub.c]
81  memcap
82  printf
83  dump_mem
84  _filbuf
85  check_list_unit_no [./tms_comm_sub.c]
86  printf
87  dump_jhub_hex [./tms_comm_sub.c]
88  printf
89  _filbuf
90  check_port_inactive [./tms_comm_sub.c]
91  log_comm_event [see line 5]
92  dump_jhub [./tms_comm_sub.c]
93  dump_jhub_hex [see line 87]
94  dump_mem
95  dump_list [./tms_comm_sub.c]
96  printf
97  dump_list_rel [./tms_comm_sub.c]
98  printf
99  init_cctv_rx_jhub [./tms_comm_sub.c]
100  sprintf
101  log_comm_event [see line 5]
102  memset
103  init_cctv_tx_jhub [./tms_comm_sub.c]
104  sprintf
105  log_comm_event [see line 5]
106  memset
107  init_rx_jhub [./tms_comm_sub.c]
108  sprintf
109  log_comm_event [see line 5]
110  memset
111  init_tx_jhub [./tms_comm_sub.c]
112  sprintf
113  log_comm_event [see line 5]
114  memset
115  init_video_tx_jhub [./tms_comm_sub.c]
116  sprintf
117  log_comm_event [see line 5]
118  memset

```

func_comm.c

```
119 init_vms_tx_rx_jhub [./tms_comm_sub.c]
120     sprintf
121         log_comm_event [see line 5]
122         memset
123         log_comm_msg_cc [./tms_comm_sub.c]
124             get_mpu_name [see line 6]
125             write_to_crash_log
126             fmt_write_comm_msg [see line 21]
127         log_jhub_hdr [./tms_comm_sub.c]
128             sprintf
129             log_comm_msg [see line 75]
130             simple_write_to_comm_log [./tms_comm_sub.c]
131                 PGM
132                 printf
133                 write_to_crash_log
134             FLUSH
135             printf
136             write_to_crash_log
137         log_list [./tms_comm_sub.c]
138             log_comm_msg [see line 75]
139             simple_write_to_comm_log [see line 130]
140             FLUSH
141             printf
142             write_to_crash_log
143         log_list_rel [./tms_comm_sub.c]
144             log_comm_msg [see line 75]
145             sprintf
146                 simple_write_to_comm_log [see line 130]
147                 FLUSH
148                 printf
149                 write_to_crash_log
150             
```

96 05/14
12 18 09

rt_skeleton.call

```
421 main [rt_skeleton.c]
422     map_to_RtDB [./tms_library/rtedb_lib.c]
423     map_to_global_section [./tms_library/global_sub.c]
424         strlen
425             MGBUSC
426             init_rtdb_t1 [./tms_library/rtedb_lib.c]
427             log_tms_event_cc [./tms_library/event_log_sub.c]
428             log_tms_common [see line 94]
429             printf
430                 STOP
431                 general_process_startup [./tms_library/proc_cntrl.c]
432                     connect_to_mailbox [./tms_library/mailbox.c]
433                         strlen
434                             ASIENN
435                                 strcpy
436                                     printf
437                                         write_to_crash_log [see line 20]
438                                         STOP
439                                         ASCEFC
440                                         clear_all_event_flags [./tms_library/proc_cntrl.c]
441                                         CLREF
442                                         printf
443                                         STOP
444                                         sprintf
445                                         write_string_to_mailbox_nowait [./tms_library/mailbox.c]
446                                         strlen
447                                         QIOW
448                                         log_tms_event [see line 93]
449                                         calc_next_22sec [rt_skeleton.c]
450                                             GETTIM
451                                                 NUMTIM
452                                                 CVT_VECTIM
453                                                 ADD_TIMES
454                                                 SCHDWR
455                                                 ASCEFC
456                                                 clear_all_event_flags [see line 440]
457                                                 fclose
458                                                 HIBER
459                                                 GETTIM
460                                                 time_stamp_rtdb [rt_skeleton.c]
461                                                 NUMTIM
462                                                 CVT_VECTIM
463                                                 SUB_TIMES
464                                                 ADD_TIMES
465                                                 READBF
466                                                 SETBF
467                                                 run_polling_processes [rt_skeleton.c]
468                                         CLREF
469                                         log_tms_event [see line 93]
470                                         run_process_one_bit [rt_skeleton.c]
471                                             ASDEFC
472                                         SETIMER
473                                         WAITER
474                                         READDEF
475                                         clear_process_event_flag [rt_skeleton.c]
476                                         ASCEFC
477                                         sprintf
478                                         log_tms_event [see line 93]
479                                         CLREF
480                                         WFLOR
481                                         CANTIM
482                                         ASCEFC
483                                         CLREF
484                                         CANTIM
```

shutdown_opc_comm.c

```
436 main [shutdown_opc_comm.c]
437     ASCEFC
438     printf
439     STOP
440     get_other_process_id [ ../../tms_library/proc_cntrl.c ]
441     strlen
442     GETUPW
443     exit
444     _fillbuf
445     SETEF
```

1

shutdown_rmdc_comm.call

06/15/1
16:44:11

```
436 main [shutdown_rmdc_comm.c]
437     ASCEFC
438     printf
439     STOP
440     get_other_process_id [../../../../tms_library/proc_ctrl.c]
441     strlen
442     GETPRW
443     exit
444     fillbuf
445     SETEF
```

06/15/15
shutdown_vms_comm.c

```
436 main [shutdown_vms_comm.c]
437     ASCBFC
438     printf
439     STOP
440     get_other_process_id [../../../../tms_library/proc_ctrl.c]
441     strlen
442     GETUPIW
443     exit
444     _filbuf
445     SETTER
```

1
05/07/95
6:53:31
snap_loop_err.call

```
439 main [snap_loop_err.c]
440     printf
441     fopen
442     perror
443     exit
444     map_to_RMDB [.. /tms_library/rtdb.lib.c]
445     map_to_global_section [.. /tms_library/global_sub.c]
446     strlen
447     MGRUSC
448     init_rtdb_t1 [.. /tms_library/rtdb.lib.c]
449     STOP
450     prompt_for_yes_no [.. /tms_library/kb_func.c]
451     strlen
452     printf
453     VMS_C_getch [see line 393]
454     copper
455     ASCBFC
456     two_bit_mask [.. /tms_library/skel_sub.c]
457     CLREF
458     WFLOR
459     NUMRTM
460     sprintf
461     unpack_rtdb_loop_strn [.. /tms_library/pack.lib.c]
462     fclose
```

stn_aggr.call

```
396 main [stn_aggr.c]
397     general_process_startup [./tms_library/proc_cntrl.c]
398     connect_to_mailbox [./tms_library/mailbox.c]
399         strlen
400             ASSIGN
401         strcpy
402         printf
403         write_to_crash_log [see line 21]
404             STOP
405             ASCEFC
406             clear_all_event_flags [./tms_library/proc_cntrl.c]
407                 CUREF
408                     printf
409                         stop
410                         sprintf
411                         write_string_to_mailbox_nowait [./tms_library/mailbox.c]
412                             strlen
413                                 QTOW
414                                     log_tms_event [./tms_library/event_log_sub.c]
415                                         log_tms_common [./tms_library/event_log_sub.c]
416                                         strlen
417                                         write_to_crash_log [see line 21]
418                                         GETTIM
419                                         memcpy
420                                         memset
421                                         write_to_mailbox_nowait [see line 394]
422                                         map_to_RMDB [./tms_library/rtdb.lib.c]
423                                         map_to_global_section [./tms_library/global_sub.c]
424                                         strlen
425                                         MEMSC
426                                         init_rtdb_bt [./tms_library/rtdb.lib.c]
427                                         log_tms_event_cc [./tms_library/event_log_sub.c]
428                                         log_tms_common [see line 415]
429                                         printf
430                                         STOP
431                                         map_to_RMDB [./tms_library/rtdb.lib.c]
432                                         map_to_global_section [see line 423]
433                                         init_rtdb_bt [./tms_library/rtdb.lib.c]
434                                         build_stn_aggr_table [stn_aggr.c]
435                                         malloc
436                                         build_tap_error [./tms_library/tap_sub.c]
437                                         sprintf
438                                         memory
439                                         printf
440                                         log_tms_event [see line 414]
441                                         exit
442                                         code_to_table [./tms_library/table_sub.c]
443                                         ushort_to_table [./tms_library/table_sub.c]
444                                         ulong_to_table [./tms_library/table_sub.c]
445                                         date_time_to_table [./tms_library/table_sub.c]
446                                         GETTIM
447                                         printf
448                                         memory
449                                         fopen
450                                         fgets
451                                         read_end [./tms_library/tap_sub.c]
452                                         strlen
453                                         memcp
454                                         search_rtdb_name_table [./tms_library/rtdb.lib.c]
455                                         memcmp
456                                         find_fdb_bt_name [./tms_library/fdb.lib.c]
457                                         memcmp
458                                         classify_roadway [./tms_library/fdb.lib.c]
459                                         memcmp
```

90/05/15
16:30:01

switch_tty.call

```
454 main [switch_tty.c]
455     printf
456     strlen
457     CMKRN1
458     exit
459     map_to_RMDB [...] /tms_library/fddb_lib.c]
460     map_to_global_section [...] /tms_library/global_sub.c]
461
462     init_rmdb_t1 [...] /tms_library/fddb_lib.c]
        strcpy
        dump_VAXportable [...] /fddb/vaxport/vaxportdb_sub.c]
        printf
        convert_DB_type [see line 176]
        convert_VAXPort_flag [...] /fddb/vaxport/vaxportdb_sub.c]
        strcpy
        convert_status [see line 504]
        print_VAXPort_list [...] /fddb/vaxport/vaxportdb_sub.c]
        printf
        select_entry [see line 513]
        longjmp
        strcpy
        convert_DB_type [...] /fddb/vaxport/vaxportdb_sub.c]
        strcpy
        vms_c_getch [...] /tms_library/kb_func.c]
        input [...] /tms_library/kb_func.c]
        ASSIGN
        QION
        402
463     STOP
464     map_to_GHLDDB [...] /tms_library/fddb_lib.c]
465     map_to_global_section [see line 460]
        init_gblddb_t1 [...] /tms_library/fddb_lib.c]
466     map_to_CCRVDB [...] /tms_library/fddb_lib.c]
467     map_to_global_section [see line 460]
        init_ccrvdb_t1 [...] /tms_library/fddb_lib.c]
468     map_to_OPRVDB [...] /tms_library/fddb_lib.c]
469     map_to_global_section [see line 460]
        init_vaxportdb_t1 [...] /tms_library/fddb_lib.c]
470
471     map_to_global_section [see line 460]
        init_ooprdb_t1 [...] /tms_library/fddb_lib.c]
472     map_to_global_section [see line 460]
        init_vaxportdb_t1 [...] /tms_library/fddb_lib.c]
473
474     setjmp
475     toupper
476     sprintf
477     switch_port [switch_tty.c]
478     toupper
479     sprintf
480     switch_port [switch_tty.c]
481     toupper
482     memmove
483     strlen
484     ASSIGN
485     printf
486     exit
487     CMKRN1
488     DASSEN
489     strcmp
490     prompt_for_yes_no [...] /tms_library/kb_func.c]
491     toupper
492     printf
493     vms_c_getch [see line 399]
494     toupper
495     bit_clr_i
496     CANCEL
497     get_port_name [switch_tty.c]
498     printf
499     scanf
500     strlen
501     toupper
502     dump_UnitTable [...] /fddb/vaxport/vaxportdb_sub.c]
503     convert_status [...] /tms_library/fddb_lib.c]
504     strcpy
505     print_UnitList [...] /fddb/vaxport/vaxportdb_sub.c]
506     strcpy
507     convert_start_flag [...] /fddb/vaxport/vaxportdb_sub.c]
508     strcpy
509     sprintf
510     print
511     printf
512     sprintf
513     select_entry [...] /tms_library/kb_func.c]
514     Log10
515     printf
516     exit
517     memset
```

t_token.c

```
472 main [t_token.c]
473     convert_non_print_to_space [token.c]
474     dump_chk_table [t_boken.c]
475         Printf
476             constant_name [token.c]
477             fopen
478                 perror
479                     exit
480                         memset
481                             fgets
482                                 printf
483                                     check_name [t_token.c]
484                                         strlen
485                                             dump_mem [see line 222]
486                                                 tolower
487                                                     get_token [token.c]
488                                                         nul_to_dest [token.c]
489                                                             char_in_set [token.c]
490                                                               tolower
491           constant_name
492               memcmp
493                   leading_zero_pad [see line 370]

222     dump_mem [../../tms_library/dump_mem.c]
223         Print_hex_ascii_line [../../tms_library/dump_mem.c]
224             sprintf
225                 printf
226                     print_n_skipped [../../tms_library/dump_mem.c]
227                         printf
370                             leading_zero_pad [../../tms_library/format_db_lib.c]
371                                 strlen
```

```

434 main [tms_shutdown.c]
435     printf
436     prompt_for_yes_no [../tms_library/kb_func.c]
437         strlen
438         printf
439             vms_c_getch [see line 388]
440             toupper
441             exit
442             connect_to_mailbox [see line 356]
443             STOP
444             flush_mailbox [./tms_library/mailbox.c]
445                 QIOW
446             log_tms_event [see line 94]
447                 ASCEFC
448                 SETEF
449                 CUREF
450             get_other_process_id [./tms_library/proc_ctrlr.c]
451                 strlen
452                 GENJPIN
453             read_from_mailbox [./tms_library/mailbox.c]
454                 QIOW
455                 strcmp

94     log_tms_event [./tms_library/event_log_sub.c]
95     log_tms_common [./tms_library/event_log_sub.c]
96         strlen
97         write_to_crash_log [see line 21]
98         GETTIME
99         memory
100        memset
101        write_to_mailbox_nowait [./tms_library/mailbox.c]
102                QIOW

356 connect_to_mailbox [./tms_library/mailbox.c]
357         strlen
358         ASSIGN
359             input [./tms_library/kb_func.c]
360             ASSIGN
361             QIOW

```

this_startupcall

```
426 main [tms_startup.c]          99      memcopy
427     printf                         100      memset
428     get_device_name [./tms_library/proc_cntrl.c] 101      write_to_mailbox_noawait [./tms_library/mailbox.c]
429     strlen                         102      QIOW
430     GETDVIW
431     STOP
432     get_proc_info [./tms_library/proc_cntrl.c] 358      clear_all_event_flags [./tms_library/proc_cntrl.c]
433     GETUPI
434     fopen
435     exit
436     fclose
437     create_logical_name [./tms_library/logical_name.c]
438     strlen
439     CRELIW
440     create_mailbox [./tms_library/mailbox.c]
441     strlen
442     CREMBX
443     flush_mailbox [./tms_library/mailbox.c]
444     QIOW
445     start_processx [tms_startup.c]
446     strlen
447     CREPRC
448     read_from_mailbox [./tms_library/mailbox.c]
449     QIOW
450     filbuf
451     ASCDFC
452     clear_all_event_flags [see line 358]
453     start_tms_process [tms_startup.c]
454     start_processesx [see line 445]
455     Printf
456     STOP
457     read_from_mailbox [see line 448]
458
459     map_to_RMDB [./tms_library/rmdb_lib.c]
460     map_to_global_section [./tms_library/global_sub.c]
461
462     init_rtab_b1 [./tms_library/rmdb_lib.c]
463     MGBLSC
464     log_tms_event_cc [./tms_library/event_log_sub.c]
465     log_tms_log_tms_common [see line 95]
466     map_to_FMDB [./tms_library/fmdb_lib.c]
467     map_to_global_section [see line 460]
468     init_fmdb_t1 [./tms_library/fmdb.lib.c]
469     map_to_RMDB [./tms_library/rmdb.lib.c]
470     map_to_global_section [see line 460]
471     init_rmdb_t1 [./tms_library/fmdb.lib.c]
472     link_rmdb_to_rmdb [tms_startup.c]
473     memset
474     memcpy
475     search_rtdb_name_table [./tms_library/rtdb.lib.c]
476     memcomp
477     printf
478     memcmp
479     search_fmdb_name_table [./tms_library/fmdb.lib.c]
480     memcmp
481     start_com_process [tms_startup.c]
482     strlen
483     CREPRC
484
485     log_tms_common [./tms_library/event_log_sub.c]
486     strlen
487     write_to_crash_log [see line 21]
488     GERTIM
```

upi_xinit.call

```
1 HM_hangup [upi_xmit.c]
2 send_HM_cmd [upi_xmit.c]
3 strlen
4 QIOW
5 log_tms_event_cc [./tms_library/event_log_sub.c]
6 log_tms_common [./tms_library/event_log_sub.c]
7 strlen
8 write_to_crash_log [./tms_library/proc_ctrl.c]
9 GETTIM
10 NUTFTIM
11 fopen
12 fprintf
13 fclose
14 GETTIM
15 memcpY
16 memset
17 write_to_mailbox_nowait [./tms_library/mailbox.
18 QIOW
19 wait_HM_result [upi_xmit.c]
20 GETTIM
21 log_tms_event_cc [see line 5]
22 get_HM_input [upi_xmit.c]
23 sprintf
24 log_tms_event [./tms_library/event_log_sub.c]
25 log_tms_common [see line 6]
26 QIOW
27 log_tms_event_cc [see line 5]
28 memcpY
29 msleep [./tms_library/misc_func.c]
30 SCHDNK
31 log_tms_event_cc [see line 5]
32 HIBER
33 atoi
34 fl_delta_time [./tms_library/misc_func.c]
35 PRINTF
36 SUB_TIMES
37 CTYPE_FROM_INTERNAL_TIME
38
39 set_process_name [./tms_library/proc_ctrl.c]
40 strlen
41 SERPRN
42
43
44 log_upi_event_cc [upi_xmit.c]
45 printf
46 main [upi_xmit.c]
47 general_process_startup [./tms_library/proc_ctrl.c]
48 connect_to_mailbox [./tms_library/mailbox.c]
49 strlen
50 memset
51 general_process_startup_flags [./tms_library/proc_ctrl.c]
52 clear_all_event_flags [./tms_library/proc_ctrl.c]
53
54 ASSIGN
55 strcpy
56 printf
57 write_to_crash_log [see line 8]
58 STOP
59 ASOFC
60 send_HM_cmd [see line 2]
61 wait_HM_result [see line 19]
62 msleep [see line 29]
63 return_HM_to_cmd_mode [upi_xmit.c]
64 STOP
65 msleep [see line 29]
```

```

528
529     memset
530     ORION
531     log_tms_event_cc [see line 5]
532     wait_HM_result [see line 19]
533     convert_HM_result [see line 516]
534     exit
535     HM_initialize [upi_xmit.c]
536     send_HM_cmd [see line 2]
537     wait_HM_result [see line 19]
538     getall_HM_Registers [upi_xmit.c]
539     read_HM_Register [upi_xmit.c]
540     sprintf
541     log_tms_event [see line 24]
542     send_HM_Cmd [see line 2]
543     wait_HM_number [upi_xmit.c]
544     GETTIN
545     log_tms_event_cc [see line 5]
546     get_HM_input [see line 22]
547     atoi
548     f1_delta_time [see line 34]
549     queue_read_upi_mbx [upi_xmit.c]
550     READER
551     READER_
552     log_tms_event_cc [see line 5]
553     queued_read_from_mailbox [./tms_library/mailbox.c]
554     QIO
555     sprintf
556     fclose
557     WFLOR
558     READER
559     CLREF
560     CANTIM
561     dial_upi_send_msg [upi_xmit.c]
562     dial_upi_number [upi_xmit.c]
563     GETTIN
564     log_tms_event_cc [see line 5]
565     turn_on_HM_DTR [./tms_library/tt_func.c]
566     memset
567     QIO
568     dial_HM_number [upi_xmit.c]
569     strcpy
570     sprintf
571     log_tms_event [see line 24]
572     strlen
573     send_HM_Cmd [see line 2]
574     wait_HM_result [see line 19]
575     sprintf
576     log_tms_event [see line 24]
577     HM_turn_off_DTR [see line 511]
578     convert_HM_result [see line 516]
579     msleep [see line 29]
580     read_HR_modem_status [./tms_library/tt_func.c]
581     memset
582     QIO
583     queue_read_upi_mbx [see line 550]
584     f1_delta_time [see line 34]
585     start_redial_timer [upi_xmit.c]
586     CVTF_TO_INTERNAL_TIME
587     SETIMER
588     log_tms_event_cc [see line 5]
589     send_report [upi_xmit.c]
590     GETTIN
591     log_tms_event_cc [see line 5]

```

vms comm call

```
1  _align [vms_comm.c]
2  main
3    init_bfr_tracking_params
4      fopen
5      printf
6      exit
7      fclose
8      check_process_interactive [...] ./tms_library/proc_cntrl.c]
9      GETJPI
10     write_to_crash_log [...] ./tms_library/proc_cntrl.c]
11     GERTTM
12     NUMTTM
13     fopen
14     fprintf
15     false
16     STOP
17     sprintf
18     create_comm_log_file [...] ./tms_comm_sub.c]
19     GERTTM
20     NUMTTM
21     strcpy
22     strcat
23     strlen
24     CREATE
25     CONNECT
26     PUT
27     GERTTM
28     FLUSH
29     log_comm_event_cc [...] ./tms_comm_sub.c]
30     get_mpul_name [...] ./tms_comm_sub.c]
31     memset
32     strlen
33     memcpy
34     get_port_name [...] ./tms_comm_sub.c]
35     memset
36     strlen
37     memcpy
38     get_unit_name [...] ./tms_comm_sub.c]
39     memset
40     strlen
41     strcpy
42     strlen
43     memory
44     strcmp
45     write_to_crash_log [see line 10]
46     fmt_write_comm_msg [...] ./tms_comm_sub.c]
47     GERTTM
48     NUMTTM
49     strcpy
50     strlen
51     PUT
52     sprintf
53     FLUSH
54     CLOSE
55     create_comm_log_file [see line 18]
56     GERTTM
57     NUMTTM
58     sprintf
59     strlen
60     memcpy
61     PUT
62     printf
63     FLUSH
64     map_to_VMSDB [...] ./tms_library/fddb_lib.c]
65     log_comm_event [...] ./tms_comm_sub.c]
66     get_mpul_name [see line 30]
67     write_to_crash_log [see line 10]
68     fmt_write_comm_msg [see line 45]
69     general_process_startup [...] ./tms_library/proc_cntrl.c]
70     connect_to_mailbox [...] ./tms_library/mailbox.c]
71     strlen
72     ASSIGN
73     strcpy
74     printf
75     write_to_crash_log [see line 10]
76     STOP
77     ASCREC
78     clear_all_event_flags [...] ./tms_library/proc_cntrl.c]
79     CIRREF
80     printf
81     STOP
82     sprintf
83     write_string_to_mailbox_nowait [...] ./tms_library/mailbox.c]
84     QIOW
85     strlen
86     establish_process_name [...] ./tms_library/proc_cntrl.c]
87     memset
88     memset
89     strlen
90     strcpy
91     get_process_name [...] ./tms_library/proc_cntrl.c]
92     memset
93     GETJPI
94     strcmp
95     get_terminal_name [...] ./tms_library/proc_cntrl.c]
96     memset
97     GETJPI
98     set_process_name [...] ./tms_library/proc_cntrl.c]
99     strlen
100    SERTPRN
101    restore_process_name [...] ./tms_library/proc_cntrl.c]
102    strlen
103    set_process_name [see line 98]
104    log_comm_event_flags [...] ./tms_comm_sub.c]
105    get_mpul_name [see line 30]
106    write_to_crash_log [see line 10]
107    fmt_write_comm_msg [see line 45]
108    log_comm_event_cc [see line 29]
109    get_terminal_name [see line 95]
110    log_comm_event [see line 66]
111    get_pid [...] ./tms_library/proc_cntrl.c]
112    GETJPI
113    sprintf
114    init_jndriver [...] ./tms_comm_sub.c]
115    strlen
116    CMKRN
117    printf
118    log_comm_event_cc [see line 29]
119    write_to_crash_log [see line 10]
120    restore_comm_process_name [see line 102]
121    STOP
122    map_to_VMSDB [...] ./tms_library/fddb_lib.c]
123    map_to_global_section [...] ./tms_library/global_sub.c]
124    strlen
125    MGBLSC
126    init_VMSDB_T1 [...] ./tms_library/fddb_lib.c]
127    map_to_OPRTVDB [...] ./tms_library/fddb_lib.c]
128    map_to_OPRTVDB [...] ./tms_library/fddb_lib.c]
```

```

129    map_to_global_section [see line 124]
130    init_optrvdb_tl [../../tms_library/fddb_lib.c]
131    map_to_VAXPRTDB [../../tms_library/fddb_lib.c]
132    map_to_global_section [see line 124]
133    init_vaxportdb_tl [../../tms_library/fddb_lib.c]
134    compare_VAX_time [../../tms_library/misc_func.c]
135    find_first_last_port_unit [../../tms_library/find_first_last.c]
136    sprintf
137    init_VMS_LIST_heads
138    get_proc_info [../../tms_library/proc_ctrl.c]
139    GETVPI
140    get_iocban [../../tms_library/kb_func.c]
141    strlen
142    ASSIGN
143    get_term_char [../../tms_library/kb_func.c]
144    QIO
145    change_term_char [../../tms_library/kb_func.c]
146    QIO
147    connect_to_mailbox [see line 71]
148    flush_mailbox [../../tms_library/mailbox.c]
149    QIOW
150    alloc_and_queue_empty_vms
151    start_VMS_PORT
152    send_VMS_START_MSGS
153    queued_get_1_char [../../tms_library/kb_func.c]
154    QIO
155    filbuf
156    queued_read_from_mailbox [../../tms_library/mailbox.c]
157    QIO
158    CIRREF
159    WFLOR
160    READDEF
161    write_string_to_mailbox [../../tms_library/mailbox.c]
162    QIOW
163    exit_MULTI_VMS_COMM
164    exit_TEST_VMS_COMM
165    multi_menu
166    strcpy
167    memcp
168    build_and_queue_VMS_BLANK_SIGN
169    find_VMSDB_ENTRY_BY_NAME [../../tms_library/vms_lib.c]
170    memcmp
171    find_msg_in_library [../../tms_library/vms_lib.c]
172    memcp
173    build_and_queue_VMS_RECALL
174    build_fp_message [../../tms_library/vms_lib.c]
175    crack_vms_message [../../tms_library/vms_lib.c]
176    init_vms_msg_struct [../../tms_library/vms_lib.c]
177    memset
178    copy_flash [../../tms_library/vms_lib.c]
179    build_fp_static_msg [../../tms_library/vms_lib.c]
180    build_fp_msg_header [../../tms_library/vms_lib.c]
181    strlen
182    toupper
183    build_fp_flashing_msg [../../tms_library/vms_lib.c]
184    build_fp_header
185    strlen
186    toupper
187    PP_POSITION_CURSOR [../../tms_library/vms_lib.c]
188    sprintf
189    build_fp_multiphase_msg [../../tms_library/vms_lib.c]
190    build_fp_msg_header
191    strlen
192    toupper
193    build_fp_arrow_msg [../../tms_library/vms_lib.c]
194    build_fp_msg_header
195    strlen
196    toupper
197    PP_RIGHT_ARROW_TO_BUFFER [../../tms_library/vms_lib.c]
198    PP_POSITION_CURSOR [see line 188]
199    PP_LEFT_ARROW_TO_BUFFER [../../tms_library/vms_lib.c]
200    strlen
201    build_and_queue_VMS_DISPLAY
202    PP_POSITION_CURSOR [see line 188]
203    build_and_queue_VMS_LIB_LOAD
204    find_qe_in_library [../../tms_library/vms_lib.c]
205    memcmp
206    remove_from_list_head_i [../../tms_library/intlk_queue.c]
207    memcmp
208    _REMOHI
209    log_comm_event [see line 66]
210    ADADI
211    ADAWI
212    log_comm_msg [see line 105]
213    log_comm_bfr [../../tms_comm_sub.c]
214    get_mpj_name [see line 30]
215    write_to_crash_log [see line 101]
216    log_comm_msg [see line 105]
217    GETRIN
218    printf
219    NWTRIN
220    sprintf
221    fmt_write_comm_msg [see line 45]
222    simple_write_to_comm_log [../../tms_comm_sub.c]
223    PUT
224    printf
225    write_to_crash_log [see line 10]
226    format_ascii_hex [../../tms_comm_sub.c]
227    memset
228    memcp
229    memset
230    crack_fp_msg [../../tms_library/crack_fp_msg.c]
231    memcp
232    memset
233    FLUSH
234    log_VMS_MSG_IN_HEX
235    return_to_free_list
236    QIOW
237    process_VMS_TX_WAIT_RX_ERRORS
238    process_VMS_GOOD_COMUM
239    process_EENO_STATUS
240    process_EENO_CONFIG
241    QIO
242    GETTIM
243    check_vms_port_status
244    scroll_port_stats [../../tms_comm_sub.c]
245    scroll_unit_stats [../../tms_comm_sub.c]
246    memcp
247    log_comm_msg [see line 105]
248    FLUSH
249    stop_vms_port
250    init_vms_unit
251    stop_vms_unit
252    build_and_queue_VMS_EXIT
253    build_and_queue_VMS_ENQUIRE

```

watch_actv_attrib

```

427 main [watch.actv.anal.c]
428     printf
429     map_to_RMDB [./tms_library/fddb.lib.c]
430     map_to_global_section [./tms_library/global_sub.c]
431     strlen
432     M68ISC
433     init_rmdb_t1 [./tms_library/fddb.lib.c]
434     STOP
435     map_to_RMDB [./tms_library/fddb.lib.c]
436     map_to_global_section [see line 430]
437     init_gdb_t1 [./tms_library/fddb.lib.c]
438     map_to_ACTVDB
439     map_to_RMDB [./tms_library/rtdb.lib.c]
440     map_to_global_section [see line 430]
441     init_rtdb_t1 [./tms_library/rtdb.lib.c]
442     get_iochan [./tms_library/kb_func.c]
443     strlen
444     ASIGN
445     get_term_char [./tms_library/kb_func.c]
446     QIO
447     set_port_partial [./tms_library/kb_func.c]
448     change_term_char [./tms_library/kb_func.c]
449     QIO
450     exit
451     vms_c_getch [see line 390]
452     toupper
453     print_egn_list [watch_actv_anal.c]
454     printf
455     select_entry [./tms_library/kb_func.c]
456     log10
457     printf
458     exit
459     memset
460     vms_c_getch [see line 390]
461     longjmp
462     atoi
463     dump_actvdb_egn
464     filbuf
465     dump_actvdb_name_table
466     dump_actvdb_params
467     dump_actvdb_offsets
468     set_term_char [./tms_library/kb_func.c]
469     QIO
470     get_tty_bit [./tms_library/fmdb.lib.c]
471     ASPECIC
472     printf
473     STOP
474     READER
475     -FFC
476     SETTER
477     one_bit_mask [./tms_library/skel_sub.c]
478     ASCEPC
479     two_bit_mask [./tms_library/skel_sub.c]
480     column_layout [watch_actv_anal.c]
481     CREATE_PASTEBORD
482     CHANGE_PBO_CHARACTERISTICS
483     CREATE_VIRTUAL_DISPLAY
484     PASTE_VIRTUAL_DISPLAY
485     DRAW_LINE
486     RWWRSTR [watch_actv_anal.c]
487     strlen
488     PUT_CHARS
489     STOP
490     display_static_labels [watch_actv_anal.c]

```

watch_bottleneck.c

```
422 main [watch_bottleneck.c]
423     printf
424         map_to_RMDB [...]tms_library/rtdb.lib.c]
425             map_to_global_section [...]tms_library/global_sub.c]
426                 strlen
427                     MBILSC
428                         init_rtdb_tl [...]tms_library/rtdb.lib.c]
429                             STOP
430                                 map_to_RMDB [...]tms_library/fddb.lib.c]
431                                     map_to_global_section [see line 425]
432                                         init_rtdb_tl [...]tms_library/rtdb.lib.c]
433                                             get_iocchan [...]tms_library/kb_func.c]
434                                                 strlen
435                                                     ASSIGN
436                                                         get_term_char [...]tms_library/kb_func.c]
437                                                             QIO
438                     set_port_partial [...]tms_library/kb_func.c]
439                         change_term_char [...]tms_library/kb_func.c]
440                             QIO
441                                 exit
442                                     malloc
443                                         process_equation_file [watch_bottleneck.c]
444                                             fopen
445                                                 perror
446                                                     exit
447             fgets
448                 pad_end [...]tms_library/tap_sub.c]
449                     strlen
450                         memcpy
451                             trim_trailing_blanks [...]tms_library/utility_func.c]
452                                 strlen
453                                     memcmp
454                                         print
455                                             _fillbuf
456                                                 find_fddb_cl_name [...]tms_library/fddb.lib.c]
457                                                     memcmp
458                                         classify_roadway [...]tms_library/fddb.lib.c]
459                                             memcmp
460                                         check_reversible [...]tms_library/tap_sub.c]
461                                             search_rtdb_name_table [...]tms_library/rtdb.lib.c]
462                                                 memcmp
463                                     atoi
464                                         strcpy
465                                             sort_list [...]tms_library/sort_lib.c]
466                                                 memset
467                                                     select_entry [...]tms_library/kb_func.c]
468                                         vms_c_getch [see line 385]
469                                             toupper
470                                                 memset
471                                         print_cabinet_list [watch_bottleneck.c]
472                                             printf
473                                                 select_entry [...]tms_library/kb_func.c]
474                                                     log10
475                                                         printf
476                                                         exit
477                                                 memset
478                                                     vms_c_getch [see line 385]
479                                         longjmp
480                                             atoi
481                                         strcpy
482                                             find_fddb_cl_name [see line 456]
483                                                 _fillbuf
484                                                     memcp
485                                         in_order_layout [watch_bottleneck.c]
```

486 printf [watch_bottleneck.c]
487 two_group_layout [watch_bottleneck.c]
488 sort_list [see line 466]
489 printf [watch_bottleneck.c]
490 squeeze_layout [watch_bottleneck.c]
491 fit_largest_unassigned [watch_bottleneck.c]
492 fit_eqn_in_win_col [watch_bottleneck.c]
493 search_rtdb_name_table [see line 461]
494 get_tty_bit [...]tms_library/rtdb.lib.c]
495 ASCIIFC
496 printf
497 STOP
498 READBF
499 FFC
500 SETTER
501 one_bit_mask [...]tms_library/skel_sub.c]
502 ASCEFC
503 two_bit_mask [...]tms_library/skel_sub.c]
504 CREATE_PASTEBOARD
505 CHANGE_PBD_CHARACTERISTICS
506 CREATE_VIRTUAL_DISPLAY
507 PASTE_VIRTUAL_DISPLAY
508 DRAW_LINE
509 sprint
510 mvwtstr [watch_bottleneck.c]
511 strlen
512 PUT_CHARS
513 STOP
514 display_sc_labels [watch_bottleneck.c]
515 mvwtstr [see line 510]
516 mvwtstr_attrib [watch_bottleneck.c]
517 PUT_CHARS
518 STOP
519 DRAW_LINE
520 mvwtstr_attrib [watch_bottleneck.c]
521 STOP
522 queued_get_l_char [...]tms_library/kb_func.c]
523 DELETE_PASTEBOARD
524 QIO
525 WFLOR
526 READBF
527 CLREF
528 CANCEL
529 DELETE_PASTEBOARD
530 NUMTMR
531 display_vol_data [watch_bottleneck.c]
532 unpack_rtbd_loop_stn [...]tms_library/pack_lib.c]
533 mvwtstr_attrib [see line 517]
534 mvwtstr [see line 510]
535 interpolate_metering_curve [watch_bottleneck.c]
536 highlight_max_rate [watch_bottleneck.c]
537 mvwtstr_attrib [see line 517]
538 mvwtstr [see line 510]
539 interpolate_metering_curve [watch_bottleneck.c]
540 highlight_max_rate [watch_bottleneck.c]
541 strcpy
542 mvwtstr_attrib [see line 517]
543 memory
544 display_history_data [watch_bottleneck.c]
545 mvwtstr [see line 510]
546 sprintf
547 mvwtstr_attrib [see line 517]
548 memory
549 memcp

watch_bottleneck call

9601514
11/17/92

```
550     memset
551     SET_CURSOR_ABS
552     set_term_char [.. /tms_library/kb_func.c]
553
385     vms_c_getch [.. /tms_library/kb_func.c]
386     input [.. /tms_library/kb_func.c]
387     ASSIGN
388     QIOW
```

Watch_fmdb.call

```
417 main [watch_fmdb.c]
418     map_to_RMDB [...] /tms_library/rmdb.lib.c]
419     map_to_global_section [...] /tms_library/global_sub.c]
420     strlen
421     MGRALSC
422     init_rmdb_t1 [...] /tms_library/rmdb.lib.c]
423     printf
424     STOP
425     map_to_RMDB [...] /tms_library/fmdb.lib.c]
426     map_to_global_section [see line 419]
427     init_rmdb_t1 [...] /tms_library/fmdb.lib.c]
428     map_to_FMDB [...] /tms_library/fmdb.lib.c]
429     map_to_global_section [see line 419]
430     init_fmdb_t1 [...] /tms_library/fmdb.lib.c]
431     vms_c_getch [see line 378]
432     toupper
433     dump_fmdb_params [...] /tms_library/fmdb.lib.c]
434     printf
435     NUMRTM
436     filbuf
437     exit
438     dump_Emdb_col_offsets [...] /tms_library/fmdb.lib.c]
439     printf
440     dump_fmdb_name_table [...] /tms_library/fmdb.lib.c]
441     dump_fmdb_offsets [...] /tms_library/fmdb.lib.c]
442     printf
443     dump_fmdb_t1 [...] /tms_library/fmdb.lib.c]
444     printf
445     get_iochan [...] /tms_library/kb_func.c]
446     printf
447     strlent
448     ASSIGN
449     get_tty_bit [...] /tms_library/fmdb.lib.c]
450     ASCEFC
451     PRINTF
452     STOP
453     READDEF
454     FFC
455     SETEF
456     one_bit_mask [...] /tms_library/skel_sub.c]
457     ASCEFC
458     two_bit_mask [...] /tms_library/skel_sub.c]
459     setjmp
460     print_data_col_list [watch_fmdb.c]
461     printf
462     select_entry [...] /tms_library/kb_func.c]
463     log10
464     printf
465     exit
466     memset
467     vms_c_getch [see line 378]
468     longjmp
469     atoi
470     longjmp
471     memcpy
472     CREATE_PASTEBOARD
473     CHANGE_PBD_CHARACTERISTICS
474     CREATE_VIRTUAL_DISPLAY
475     PASTE_VIRTUAL_DISPLAY
476     DRAW_LINE
477     name_time_titles [watch_fmdb.c]
478     mwrstr [watch_fmdb.c]
479     strlen
480     PUT_CHARS
481     STOP
482     loop_titles [watch_fmdb.c]
483     mwrstr_underline [watch_fmdb.c]
484     strlen
485     PUT_CHARS
486     STOP
487     display_loop_names [watch_fmdb.c]
488     memory
489     mwrstr [see line 478]
490     search_fmdb_name_table [...] /tms_library/fmdb.lib.c]
491     memcmp
492     trap_titles [watch_fmdb.c]
493     mwrstr_underline [see line 483]
494     display_trap_names [watch_fmdb.c]
495     memory
496     mwrstr [see line 478]
497     search_fmdb_name_table [see line 490]
498     strn_titles [watch_fmdb.c]
499     mwrstr_underline [see line 483]
500     display_stn_names [watch_fmdb.c]
501     memory
502     mwrstr [see line 478]
503     vms_c_getch [see line 378]
504     mwrstr [see line 478]
505     queued_set_1_char [...] /tms_library/kb_func.c]
506     QIO
507     WFLOR
508     READDEF
509     CLEARF
510     CANCEL
511     DASSGN
512     DELETE_PASTEBOARD
513     NUMTIN
514     sprintf
515     NUMTIN
516     display_loop_work_data [watch_fmdb.c]
517     sprintf
518     mwrstr [see line 478]
519     display_station_work_data [watch_fmdb.c]
520     sprintf
521     mwrstr [see line 478]
522     display_trap_work_data [watch_fmdb.c]
523     sprintf
524     mwrstr [see line 478]
525     display_loop_curr_data [watch_fmdb.c]
526     unpack_fmdb_loop
527     sprintf
528     mwrstr [see line 478]
529     display_station_curr_data [watch_fmdb.c]
530     unpack_fmdb_station
531     sprintf
532     mwrstr [see line 478]
533     display_trap_curr_data [watch_fmdb.c]
534     unpack_fmdb_spd_trap
535     sprintf
536     mwrstr [see line 478]
537     SET_CURSOR_ABS
538     vms_c_getch [...] /tms_library/kb_func.c]
539     Input [...] /tms_library/kb_func.c]
540     ASSIGN
541     QIOW
```

96/05/14
13:28:52

watch_rmdc_call

```

415 main [watch_rmdc.c]
416     map_to_rtdb [...]tms_library/rtdb_lib.c]
417     map_to_global_section [...]tms_library/global_sub.c]
418     strlen
419     MBFLSC
420     init_rtdb_t1 [...]tms_library/rtdb_lib.c]
421     printf
422     STOP
423     map_to_rtdb [...]tms_library/fddb_lib.c]
424     map_to_global_section [see line 417]
425     init_rtdb_t1 [...]tms_library/fddb_lib.c]
426     vms_c_getch [see line 376]
427     toupper
428     dump_rtdb_params [...]tms_library/rtdb_lib.c]
429     printf
430     NUMTIM
431     filbuf
432     exit
433     dump_rtdb_col_offsets [...]tms_library/rtdb_lib.c]
434     printf
435     dump_rtdb_name_table [...]tms_library/rtdb_lib.c]
436     printf
437     dump_rtdb_offsetsets [...]tms_library/rtdb_lib.c]
438     printf
439     dump_rtdb_t1 [...]tms_library/rtdb_lib.c]
440     printf
441     get_iocman [...]tms_library/kb_func.c]
442     strlen
443     ASSIGN
444     get_tty_bit [...]tms_library/fmrdc.lib.c]
445     ASCFRC
446     printf
447     STOP
448     READF
449     _FFC
450     _SETEF
451     one_bit_mask [...]tms_library/skel_sub.c]
452     ASCFRC
453     two_bit_mask [...]tms_library/skel_sub.c]
454     setjmp
455     print_data_col_list [watch_rmdc.c]
456     printf
457     select_entry [...]tms_library/kb_func.c]
458     log10
459     printf
460     exit
461     memset
462     vms_c_getch [see line 376]
463     longjmp
464     atoi
465     longjmp
466     memcpy
467     CREATE_PASTEBOARD
468     CHANGE_PBD_CHARACTERISTICS
469     CREATE_VIRTUAL_DISPLAY
470     PASTE_VIRTUAL_DISPLAY
471     mrvrtstr [watch_rmdc.c]
472     strlen
473     PUT_CHARS
474     STOP
475     loop_titles [watch_rmdc.c]
476     mrvrtstr_underline [watch_rmdc.c]
477     convert_status [...]tms_library/fddb_lib.c]
478     strcpy

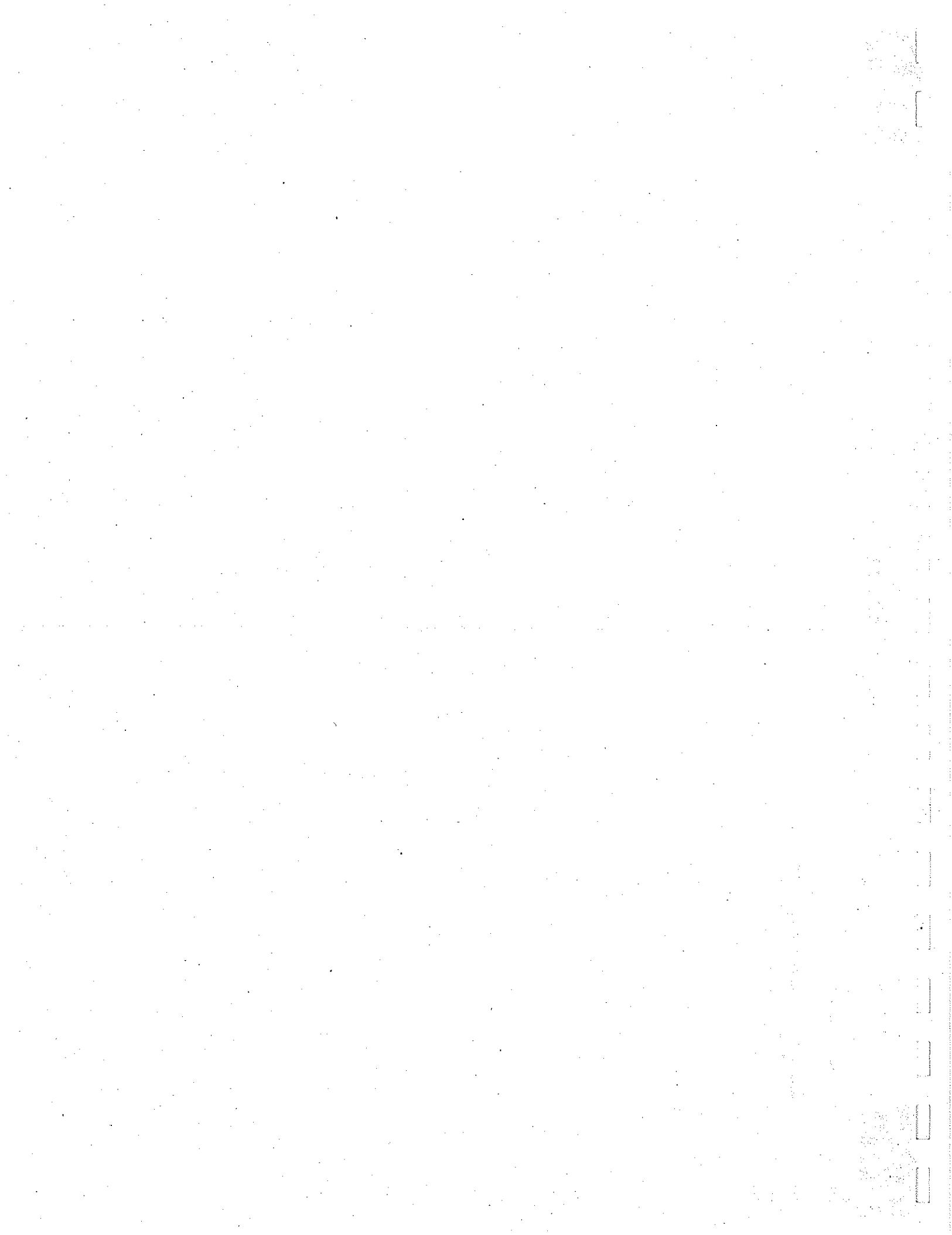
```

1

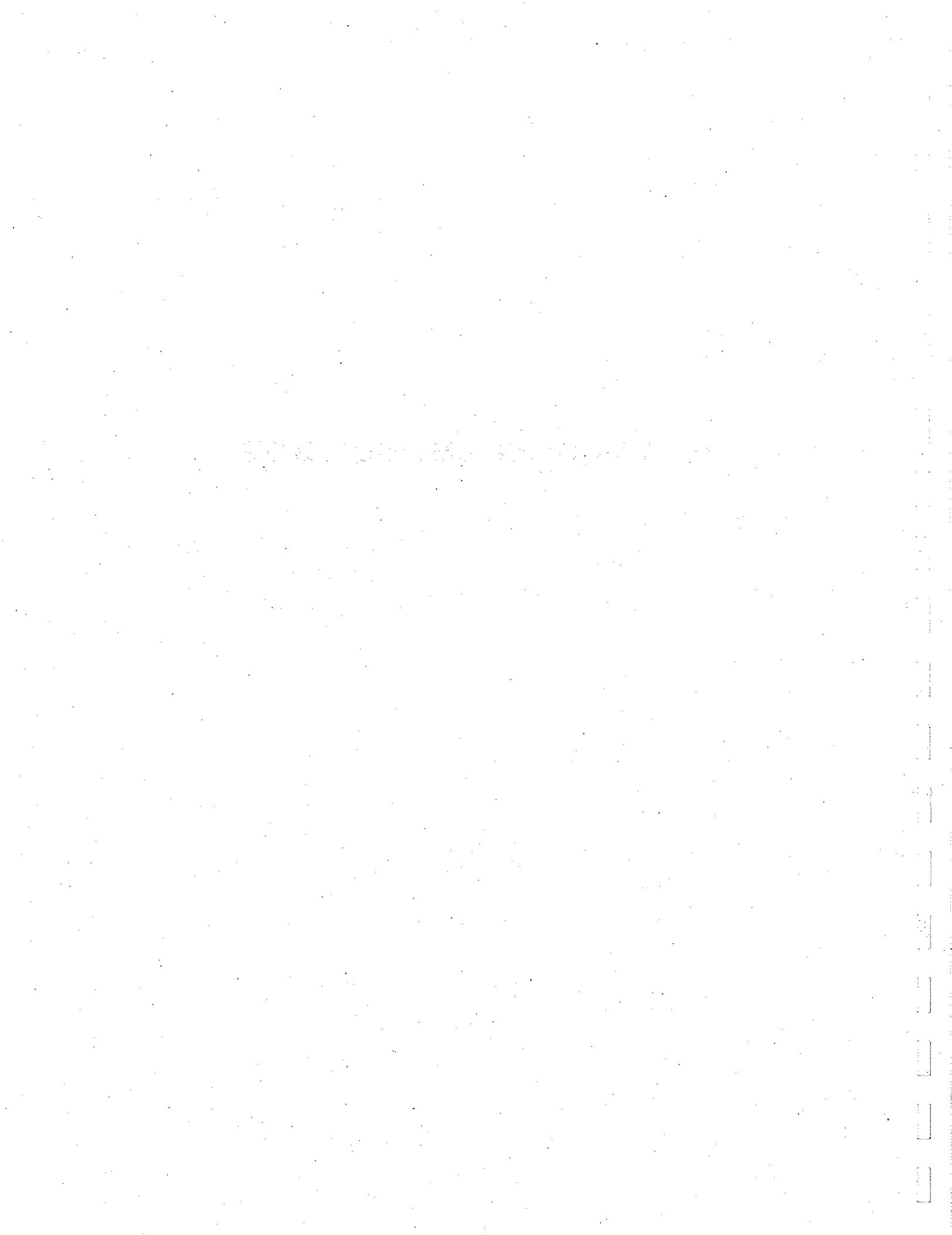
Watch command call

543 mvwtstr [see line 471]
544 SET_PHYSICAL_CURSOR

376 vms_c_getch [...]ems_library/kb_func.c]
377 input [...]tms_library/kb_func.c]
378 ASSIGN
379 QIOW



II. VAX-170 COMMUNICATIONS



VAX <--> 170 Communications:

These notes supplement HNTB's documentation, clarifying when each command is issued and what parameters of interest it contains. See HNTB documentation for bit specifications and parameters sent.

Error Request

Sent in startup sequence following load parameters.
Also sent following a NAK response.

Error Detail

Sent in response to error request.
Contains last restart time, time duration of last power outage, general error bits, error bits for ramp lanes 1, 2, and 3 (bulbs out, loop detector failure).

170 Response

Sent in response to date/time, meter control, idle, reset, lp1, lp2, and lp3.
There is a 1 bit difference in Command byte between ACK and NAK. This error bit is set when any failure bit is set or cleared in ERROR DETAIL.

Date/Time

Sent every 6 hours to resynchronize 170 with VAX.

Data Poll

Sent every 20 seconds. Separate poll and data response for each 170.
Parameter specifies metering rate adjustment.

Data Response

Sent in response to data poll.
Contains ramp rate, lane status (local, bottleneck, override condition), volume, occupancy, error flags (short pulse, chatter, etc) speed traps data.

Meter Control

Sent as needed.
Parameter specifies start or stop of central or TOD metering.

Idle

Sent when 170 sends NAK instead of ACK. Idles are then sent every 20 seconds until an ACK response. 170 Startup sequence occurs when an IDLE/ACK exchange occurs.

Reset

Reset (set all bits) for startup.

Also sent following error request??

Parameter specifies what to reset: loops, counters, 170 s/w, or comm protocol.

Load Parameters 1

Sent in startup sequence following date/time.

Also sent when lp1 change occurs from OPC or PATCH.

If change is only in lp1, then lp2 and lp3 are not sent.

Contains Local metering table, minimum and maximum metering rates, queue adjustment parameters, etc.

Load Parameters 2

Sent in startup sequence following load parameters 1.

Also sent when load parameter change occurs from OPC or PATCH.

Contains TOD metering table.

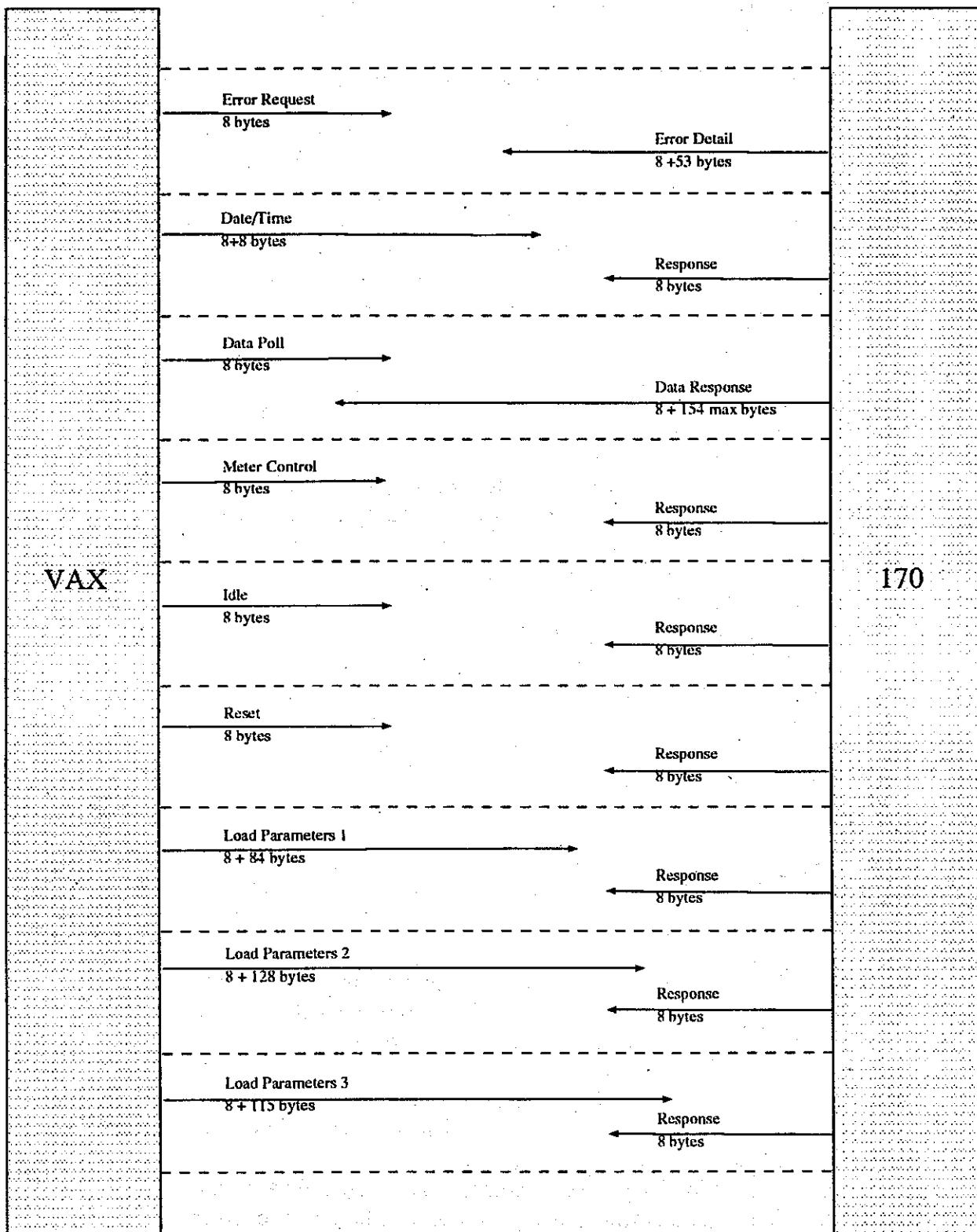
Load Parameters 3

Sent in startup sequence following load parameters 2.

Also sent when load parameter chance occurs from OPC or PATCH.

Contains loop function codes, speed trap parameters, data validity parameters, data switch (data collector or ramp meter), # of actives loops, number of metered lanes, control switch (Central or TOD), etc.

VAX <--> 170 Communication



RMDC_COMM

Ramp Meter/Data Collectors Communications Handler

In main RMDC_COMM processing loop, check for tms shutdown, process keyboard interrupts (such as go to test menu), process mailbox interrupts (such as update 170 parameters), process data received from 170s, process buffers ready to send to 170s, and send data polls to 170s.

- The cmd_list contains protocol commands, which are used by OPC_COMM
- **find_first_last_port_unit** -- Finds first and last VAX port, and first and last unit by searching the port_device_table, which was built prior offline [fddb/vaxport/build_vaxport.c]. The first port is the first one that has the same db_type. The last port is the last one that has the same db_type. The ports were previously sorted by db_type in sort_list_by_db_type [fddb/vaxport/build_vaxport.c]
- **start_rmdc_port** invoked in multi_port startup enables all ports
- In startup, **send_rmdc_start_msgs** calls:
 - **build_and_queue_170_msgs**
 - **build_and_queue_170_date_time**
 - **build_and_queue_load_parameters** [in rmdb_comm_sub.c] initializes 170 parameters
 - Check that port corresponds to RM/DC rather than gate controller
 - Get port number (calculated from unit_no passed in) and check that it is good
 - Make sure port is in correct mode: test versus multi
 - Calculate RMDB column pointer from unit_no and obtain drop address from data column
 - **get_rmdc_buffer** -- get 266 byte Tx/8 transmission buffer
 - **init_tx_rx_jhub** -- initialize the buffer
 - Fill in header fields of buffer, such as drop address, port #
 - Build one of three transmission messages from parameters in data column
 - LOAD_PARAM1
 - LOAD_PARAM2
 - LOAD_PARAM3
 - **add_to_list_tail_i** -- put buffer at end of transmission queue
 - Set flag RMDC_V_TX_WAIT_RX to indicate ready to transmit to 170
- In main processing loop:
 - Changes in RMDC are made through mailbox interrupts, which are sent from OPC_COMM. The mailbox message contains the command and the unit number (from which the rmdb column pointer can be calculated). It only processes 1 command at a time.
process_mailbox_command [in rmdc_comm_sub] switches based on the command, which include start metering, stop metering, load parameters, reset loops, reset communications, and reset counters. For load parameters, it checks to see if any of the update flags are set for the given data column. There are different update bits for load parameters 1, load parameters 2, load parameters 3 (the group to which each element

belongs is assigned in the rmdb name table [fddb/rmdb/rmdb_tbl.c]). This means that load parameters 1 can be sent without sending load parameters 2 and 3. If update flag is set (0x00 means don't update), the new parameters or commands are sent to the 170. It then clears the update flag.

Updates to RMDC can occur in two ways:

- 1) Update flag can be set in **process_read_write_update** [comm_prot/opc_comm/opc_comm_sub.c], which is called from **opc_comm** in response to mailbox command. It in turn calls **mail_rmdc_updates**, which calls **mail_cmd_to_rmdc_comm** (using the command **OPC_RMDC_LOAD_PARAM** meaning a message is sent from **OPC_COMM** to **RMDC_COMM**), which calls **write_to_mailbox_nowait**.
- 2) Update flag can be set in **patch_rmdb** when user selects modification. **patch_rmdb** calls **check_rmdb_changed**, which calls **mail_patch_cmd_to_rmdc_comm**, which calls **write_to_mailbox_nowait**. The mailbox command used in this case is **OPC_RMDC_LOAD_PARAM** (see /tms_include/tms_system.h for definitions of mailbox commands, not to be confused with the VAX to 170 commands defined in **rmdc_comm.h**). In this case, **RMDC_COMM** handles the command directly rather than going through the **OPC_COMM**

NOTE on protection: **process_read_write_update** checks the user's permission to see if they can make changes to the RMDB. For all load parameters, the permission established in **rmdb_tbl.c** (which can be overridden in **rmdb_input.fil**) is that anyone can view the load parameters, and the managers, supervisors, and operators can change the RMDB (trainees cannot). **patch_rmdb** does not require that the user has permission.

NOTE on journaling: **process_read_write_update** writes to **rmbd_journal.fil** whenever a change is made to the RMDB. It writes the column name, operator initials, time of change, followed by element name = new value. **patch_rmdb** does not write to the journal file when a change is made! The journal file was opened by **create_journal_file** [comm_prot/opc_comm/opc_comm_sub.c], which is called from **opc_comm** upon startup. The journal file is closed by **close_file_for_exit** [comm_prot/opc_comm/opc_comm_sub.c] by **exit_test_opc_comm** upon shutdown either from **opc_comm** menu or main shutdown flag.

- Process data received from 170s
 - **process_rmdc_good_comm** [in **rmdc_comm_sub.c**] -- process 170 buffer that has no error. Check retry count, status, etc.
 - response command indicates whether M170_ACK, DATA_RESP, etc.
 - **process_data_response** [in **rmdc_comm_sub.c**] called when tx_header says DATA POLL
 - the received 170 buffer (-> tx_ubfr) is passed in
 - check for RMDC (make sure it's not a gate controller)

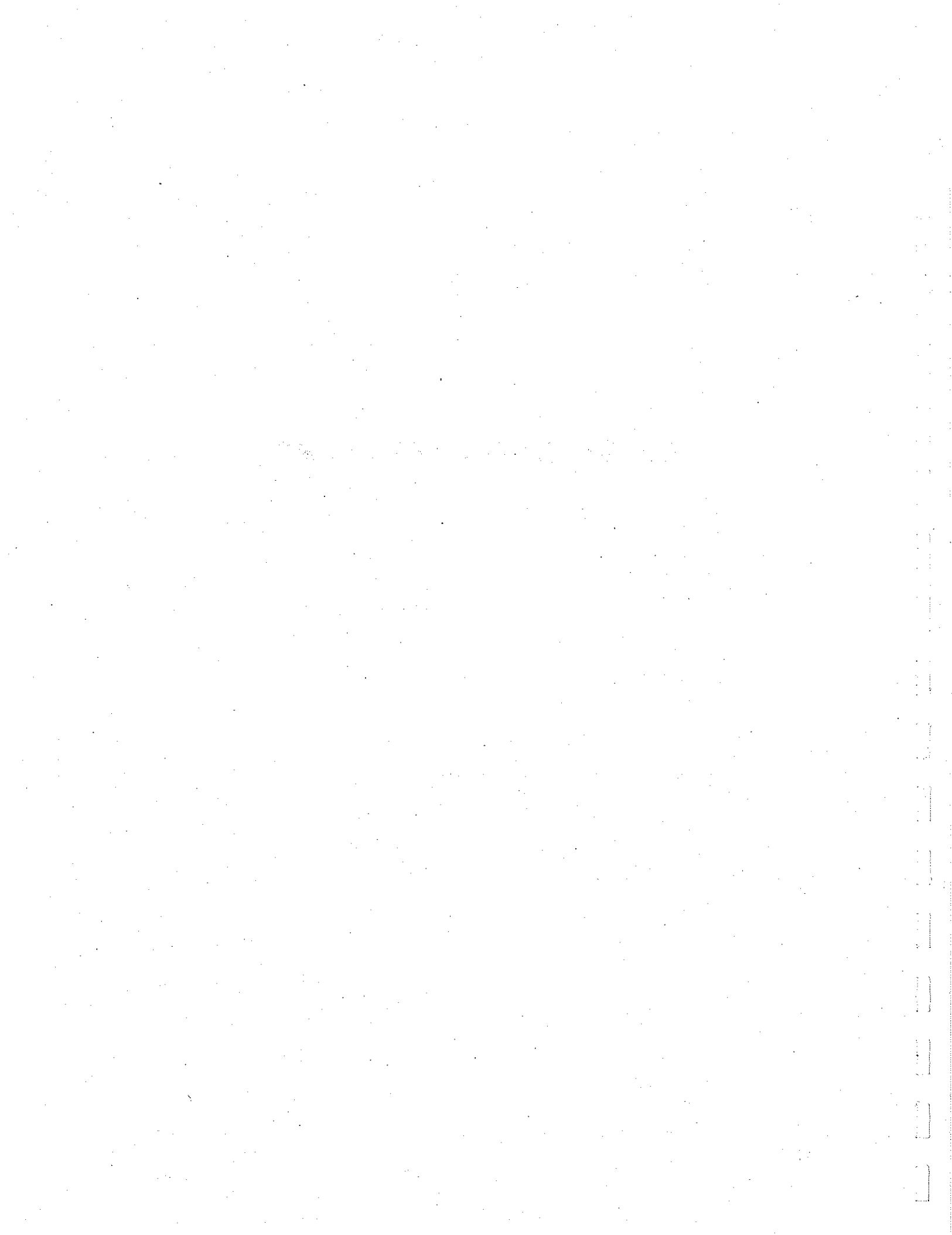
- calculate pointer to rmdb column from unit_no given in received buffer
 - calculate pointer to buffer that has been received from 170 and call it rx_ubfr
 - Calculate how long the buffer should be and make sure it's that length
 - Copy 170s calculated local metering rates 1, 2, and 3 from received buffer to RMDB
 - Get lane status 1, 2, and 3 from 170 buffer and copy into RMDB
 - **unpack_170_vol_occ** for each loop in 170-- extracts volume and scan count from received 170 buffer
 - For each loop in 170, get data validity flag
 - If the operator disabled bit is set, set it to OPER_DISABLED
 - **pack_rtdb_loop** -- enters lane volume, scan count, and data validity flag into RTDB
 - **log_170_data_poll** to file rmdc_comm_x.log
- Process buffers that are ready to be sent to 170s
 These buffers were queued in **build_and_queue_170_msg** (see below), and may be of type IDLE_COMM, REST, METER_CTL, GATE_STATUS, DATA_POLL, ERROR_REQ, or GATE_ERR_REQ
- If RMDC_M_TX_WAIT_RX flag is set
 - For each port and while buffers are still on the queue
 Transmit buffer with system command SYS\$QIO
 - For each port and for each unit, data polling takes place

Note: The non-active ports are skipped over if the VAXPort for that data column is "NO_PORT" or "NO_CTRL" when the port_device_table is built [in fddb/vaxport/build_vaxport.c]. The user defined default data columns have NO_PORT, which is why they are not treated as real devices.

build_and_queue_170_msg [in rmdc_comm_sub.c]

- Get port # (calculated fromm unit_no, which was passed in)
- Make sure port is in correct mode -- multi_mode
- Calculate the drop address from the unit_no
- **get_rmdc_buffer** of appropriate size
 - 8 byte Tx/8 used by IDLE_COMM, REST, METER_CTL, GATE_STATUS
 - 8 byte Tx/266 used by DATA_POLL, ERROR_REQ, GATE_ERR_REQ
- **init_tx_rx_jhub** -- initialize the buffer
- Fill in header fields, such as drop address, port #, 170 command, and 170 parameter (such as metering rate adjustment)
- **add_to_list_tail_i** -- puts buffer on end of queue (called tx_wait_rx_list) uses a build-in VMS function _INSQTI. Queue is later transmitted by \$QIO.
- Set event flag RMDC_V_TX_WAIT_RX to indicate a buffer is waiting to be transmitted

III. REAL-TIME SOFTWARE



TMS_STARTUP

Supplemental notes to HNTB's documentation

To Start System:

1) Rebuild databases off line if any changes have been made:

See HNTB documentation for list of **build_xxdb.exe** and input files required. Note that **build_rtdb** and **build_fmdb** are not off-line builds, but run from **tms_startup**.

2) Run **TMS_STARTUP.EXE** [./rt_skeleton/tms_startup.c] to start mains. This is not called from a .com file.

- Creates all permanent mailboxes so that process can assign channels to them as they startup:
TMS_CONTROL_MBX, EVENT_LOG_MBX, RMDC_COMM_M_MBX,
RMDC_COMM_T_MBX, OPC_COMM_M_MBX, OPC_COMM_T_MBX,
VMS_COMM_M_MBX, VMS_COMM_T_MBX, UPI_XMIT_MBX,
MON_EVT_LOG_MBX
- Start event logger and wait to make sure successfully started.
- Create permanent Event Flag Clusters
- Starts upi_xmit, noaa_monitor, build_rtdb, and build_fmdb
- map_to_RTDB, map_to_FMDB, and map_to_RMDB. link_rmdb_to_rtdb.
- Start dumydata, stn_aggr, inc_detect, bottleneck, fmdb_aggr, fmdb_archiver, rt_skeleton
- Starts communications handlers MULTI_RMDC, MULRI_OPX_XOMM,
MULTI_VMS_COMM with call to
*cond_code = start_comm_process("MULTI_RMDC_COMM", "MULTI_RMDC_COMM",
0,
ast_limit, bio_limit, byte_limit, dio_limit, status_flag,
&pid, input_dev_name, output_dev_name, error_dev_name);*

Calls function for each main using **start_tms_process**

```
start_tms_process("BOTTLENECK", tms_control_mbx, 1,  
input_dev_name, output_dev_name, error_dev_name);
```

Note: The parameters given are the same for each main (were assigned in prior call to *get_device_name*) and will not need to be changed for FUZZYMETER. Just need to add an extra call to **start_tms_process** here. Sub-functions don't need modification.

Note: First parameter called filename is image name.

Note: Although this starts main for TAPS, it does not run calculations until **rt_skeleton** calls it every 20 seconds. It does build tables, but then waits for flag with system utility **WAITFR**.

Note: Although the **watch_bottleneck** event flag clusters are associated here, the main is not started here.

start_tms_process [./rt_skeleton/tms_startup.c] sets privilege descriptors, status flag, and then calls

```
cond_code = start_processsx(filename, filename, 0,  
0, status_flag, &pid, input_dev_name, output_dev_name, error_dev_name);
```

start_processsx [./rt_skeleton/tms_startup.c]

- Builds descriptors for image name, process name, SYS\$INPUT, SYS\$OUTPUT, SYS\$ERROR
- Build process quota list
- Start process using VMS system service

```
cond_code = SYS$CREPRC(pid, &image, &sys_input, &sys_output, &sys_error,  
priv_ptr, quotas, &prcnam, 0, 0, 0, status_flag);
```

SYS\$CREPRC is the function that actually starts a main from a main!

SYS\$CREPRC

- process address - where CREPRC writes process ID
- image - char pointer to file name
- input - SYS\$INPUT
- output - SYS\$OUTPUT
- error - SYS\$ERROR
- prvadr - privileges of process
- quota - max subprocess, max ASTs, etc
- prcname - process name 1-15 chars
- baspri - base priority of process. 31 is highest, 0 is lowest. Real time priorities 16-31.
- uic - user ID code to assign to process
- mbxunt - mailbox # to send termination message to when process deleted
- stsflag - status flag where each bit has a symbolic name for options, like detached, hyber, etc.

TMS_SHUTDOWN

Tms_shutdown sets the shutdown flag for the communications handlers: `multi_opc_comm`, `test_opc_comm`, `multi_rmdc_comm`, `test_rmdc_comm`, `multi_vms_comm`, `test_vms_comm`, `upi_xmit`, `noaa_monitor`, and real time processes. Each of these mains checks to see if the shutdown bit is set before completing the next loop. "Run TMS_SHUTDOWN" initiates shutdown.

Note: `stop_tms.com` is not normally used. `@STOP_TMS` aborts the processes and does not give control to `exit_handling` routines. It is used after a failed startup to get rid of all TMS processes. It shuts down `stop_tms.com`, `event_logger`, `rt_skeleton`, `dumydata`, `stn_aggr`, `inc_detect`, `bottleneck`, `read_rtddb`, `fmdb_aggr`, `fmdb_archiver`, `noaa_monitor`, `upi_xmit`, `multi_opc_comm`, `multi_rmdc_comm`, `multi_vms_comm`.

- Display shutdown banner on screen
- Prompt to screen to verify shutdown
- `connect_to_mailbox TMS_CONTROL_MBX` and flush it
- `connect_to_mailbox EVENT_LOG_MBX` and send the `TMS_SHUTDOWN` message
- Set the shutdown bit for `multi_opc_comm`, `test_opc_comm`, `multi_rmdc_comm`, `test_rmdc_comm`, `multi_vms_comm`, `test_vms_comm`, `upi_xmit`, `noaa_monitor`, and `rt_common` (the ones in *italics* are not shutdown in `stop_tms.com`). The `rt_common` flag (`RT_V_SHUTDOWN_START`) is used for all of the real time processes, which include `rt_skeleton`, `dumydata`, `stn_aggr`, `inc_detect`, `bottleneck`, `fmdb_aggr`, `fmdb_archiver`, and `event_logger`. In `rt_skeleton`, if `RT_V_SHUTDOWN_START` is set, then set the `RT_V_SHUTDOWN_TMS` flag. The real-time processes check for the shutdown tms flag, which is synchronized by the 20 sec tick.
 - Associate to the event flag cluster with system call `SYS$ASCEFC`
 - Set the shutdown flag (for example, `OPC_V_SHUTDOWN_TMS`) with system call `SYSS$SETEF`
 - If the flag was clear, no problem
 - Else if the flag was not clear, clear it and reset it.
 - Print to screen that flag was successfully set.
- Loop until `read_from_mailbox` returns message "FMDB_ARCHIVER Terminating"

Note: `Read_rtddb` checks a different shutdown flag, masked by `RR_M_SHUTDOWN_TMS`. In `call SYS$READEF`, results are written to `efc_bits`, and bitwise check is done later to check shutdown. Where does `RR_M_SHUTDOWN_TMS` get set? `RR_V_20_sec_tick` is set by `run_process_alt_bit`.

Note: `tms_shutdown` waits until `fmdb_archiver` is shut down, then says shut down is complete. How is it that `fmdb_archiver` is the last process to shut down?

RT_SKELETON

Description of algorithm and system utilities used. RT_SKELETON associates and initializes event flags, then enters loop. Every 20 seconds, it starts up polling processes (multi_rmdc_comm, test_rmdc_comm, and dummy_data). It starts traffic analysis programs (stn_aggr, inc_detect, and bottleneck), then scrolls RTDB. At this point, the databases are stable. The communications handlers are run (multi_opc_comm, test_opc_comm, multi_vms_comm, and test_vms_comm). The monitoring programs are run (watch_rmdc, watch_fmdb, watch_btlneck, and read_rtdb). Stay in loop until shut down tms event flag turns on.

rt_skeleton [./rt_skeleton/rt_skeleton.c]

- **map_to_RTDB**
 - map_to_global_section
 - init_rtdb_tl
- **general_process_startup**
 - connect_to_mailbox
 - write_to_crash_log
 - Associate to event flag cluster
 - Clear all event flags
- **log_tms_event**
 - log_tms_common
 - write_to_crash_log
 - get time
 - compose message
 - write_to_mailbox_nowait
- **calc_next_22sec** -- begin data polling 2 seconds after 20 sec clock tick to allow time for 170 data responses to return
- Associate to WATCH_RMDC Event Flag Cluster
Clear All Event Flag Bits
Initialize WATCH_RMDC 20 Second Tick EF Bit Number
Associate to W_RMDC_TTYS Event Flag Cluster
Clear All Event Flag Bits
- Repeat above for WATCH_FMDB
- Repeat for WATCH_BTLNECK
- Repeat for READ_RTDB
- Close stdin, stdout, stderr
- **While SHUTDOWN_TMS event flag is clear**
 - Get current time with system utility SYS\$GETTIM
 - time_stamp_rtdb with current time. The start time for the previous 20 second period is located at the rtdb new column base, and the end time is located right after the start time.
 - Test SHUTDOWN_TMS event flag (EF) with call to
cond_code = SYS\$READEF(RT_V_SHUTDOWN_TMS, &efc_bits);
If set, then set SHUTDOWN_TMS bit so each process can terminate

```
cond_code = SYS$SETEF(RT_V_SHUTDOWN_TMS);
```

Continue with loop, with each process checking to see if it should shutdown.

- Call **run_polling_processes** (see below for details) to start MULTI_RMDC_COMM, TEST_RMDC_COMM, and DUMMY_DATA using **run_process_one_bit**.
- Call **run_process_wait** (see below for details) to start TAPS. Wait for each to complete before starting next.

```
run_process_wait("STN_AGGR", RT_V_STN_AGGR_START,  
RT_V_STN_AGGR_DONE, 1000);
```

```
run_process_wait("INC_DETECT", RT_V_INC_DETECT_START,  
RT_V_INC_DETECT_DONE, 1000);
```

```
run_process_wait("BOTTLENECK", RT_V_BOTTLENECK_START,  
RT_V_BOTTLENECK_DONE, 1000);
```

Note: Additional call needed to **run_process_wait** added for fuzzymeter.

The event flags RT_V_FUZZYMETER_START, RT_V_FUZZYMETER_DONE, RT_M_FUZZYMETER_START, RT_M_FUZZYMETER_DONE must be defined in tms_realtime.h, along with their corresponding event flag masks.

Note: Event flags with _V are used in SYS\$CLREF, SYS\$READDEF, SYS\$SETEF, and SYSS\$SETMR. The _V is the process name described by a defined #, not a flag in that sense that it can't turn on and off. The flag is internal to the system, and the _V is just used to indicate which event. The _M is 128-bit mask used in bitwise AND with efc_bits, and bitwise OR with ef_mask. Each event flag mask has only one of its 128 bits on.

- **scroll_rtdb_col_offsets** [in tms_library/rtdb_lib.c] makes room for the latest data by moving data columns down one.
 - rtdb_cl is an array of offsets to data columns. Rather than moving the data itself, the data offset is incremented by one. The offset to the last data column is relocated to the first offset so that the same memory can be reused.
 - The first data column is initialized to zero (not the offset itself, but where it points to).
- Call **run_process_one_bit** to start operator/console communications

```
run_process_one_bit("MULTI_OP_CCOMM", &opc_comm_efc_m,  
OPC_V_20_SEC_TICK);
```

```
run_process_one_bit("TEST_OP_CCOMM", &opc_comm_efc_t,  
OPC_V_20_SEC_TICK);
```

```
run_process_one_bit("MULTI_VMS_COMM", &vms_comm_efc_m,  
VMS_V_20_SEC_TICK);
```

```
run_process_one_bit("TEST_VMS_COMM", &vms_comm_efc_t,  
VMS_V_20_SEC_TICK);
```

Associate to process event flag cluster with SYS\$ASCEFC, then start event flag to begin process with SYS\$SETEF.

- Call **run_process_alt_bit** to start WATCH_RMDC, WATCH_FMDB, WATCH_BTLNECK, and READ_RTDB with
- ```
run_process_alt_bit("WATCH_RMDC", &watch_rmdc_efc, &wr_v_20_sec_tick);
```
- ```
run_process_alt_bit("WATCH_FMDB", &watch_fmdb_efc, &wf_v_20_sec_tick);
```

```
run_process_alt_bit("WATCH_BTLNECK", &watch_btlneck_efc, &wb_v_20_sec_tick);
run_process_alt_bit("READ_RTDB", &read_rtdb_efc, &rr_v_20_sec_tick);
```

This function starts a process which uses an alternating bit start. The difference between this function and run_process_one_bit is the extra step of toggling the start_event_flag (flip all bits). It uses SYS\$SETEF(*start_event_flag) to start process.

Note: WATCH_FUZZYMR will need to be started here. watch_fuzzymr_efc will need to be added to tms_realtime.h, along with the corresponding 20_sec_tick definitions. wf has already been used for watch_fmdb and cannot be reused for watch_fuzzymr.

- Test SHUTDOWN_TMS event flag with call to
cond_code = SYS\$READEF(RT_V_SHUTDOWN_TMS, &efc_bits);

run_polling_processes -- starts MULTI_RMDC_COMM, TEST_RMDC_COMM DUMMY_DATA and waits for completion or timeout

- Clear active bits and done bits for MULTI_RMDC_COMM, TEST_RMDC_COMM, and DUMMY_DATA using SYS\$CLREF.
- Call **run_process_one_bit** to start MULTI_RMDC_COMM, TEST_RMDC_COMM, and DUMMY_DATA
- Set timer with SYS\$TIMR for 1 second, during which time polling message are built and queued. Wait for timer flag RT_V_SKEL_TIMER to finish with SYS\$WAITFR, then clear flag with SYS\$CLREF -- A 1 second delay.
- Read event flags to see if MULTI_RMDC_COMM, TEST_RMDC_COMM, or DUMMY_DATA is active with call
cond_code = SYS\$READEF(RT_V_MULTI_RMDC_COMM_ACTIVE, &efc_bits);

Note: All three processes use the same flag RT_V_MULTI_RMDC_COMM_ACTIVE.

- Do below for all three processes:
- If the result is still equal to the mask, then the process is active -- Bitwise AND efc_bits with mask (of form RT_M_*_ACTIVE)
 - Clear active event flag (of form RT_V_*_ACTIVE) using SYS\$CLREF so it can be rechecked later.
 - If process is done (Bitwise AND with mask of RT_M_*_DONE is still equal to mask)
 - Clear done flag (of form RT_V_*_DONE) using SYS\$CLREF so it can be rechecked later
 - Else the process is not done. Build a polling mask (Bitwise AND polling_done_mask with RT_M_*_DONE)
- Else the process is not active. The polling mask remains blank.
- If the polling mask is still blank, all processes are done. Return.
- Set timer with SYS\$TIMR for 10 seconds.
- While processes (MULTI_RMDC_COMM, TEST_RMDC_COMM, or DUMMY_DATA) are still running and timer still running

- Build wait mask -- Bitwise AND of processes that are not completed and timer mask.
- Sit and wait until a processes completes or timeout -- using SYS\$WFLOR
- Read the event flag using SYS\$READDEF to set efc_bits.
Note: efc_bits represents all 3 processes.
- Repeat below for all 3 processes:
- If the process is done (Bitwise AND of efc_bits with RT_M_*_DONE)
 - Clear the done flag (of form RT_V_*_DONE) with SYS\$CLREF
 - Zero that bit in polling mask for the event that completed -- using one's complement of event mask Bitwise ANDed with polling mask
 - If all processes are done (polling mask is zeroed)
Cancel the timer with SYS\$CANTIM and return

run_process_wait -- Used to start TAPS. Waits for completion before returning.

- SYS\$SETEF is system utility that sets event flag, such as RT_V_BOTTLENECK_START. Bottleneck.c checks this flag to see when to run calculation loop.

cond_code = SYS\$SETEF(start_event_flag);

The cond_code WASCLR means that the event flag was previously clear, as expected.

- Set a timer that sets event flag RT_V_SKEL_TIMER when reaching timeout (converted wait time given as argument).

cond_code = SYS\$SETIMR(RT_V_SKEL_TIMER, &timeout, 0, WAIT_TIMER_ID, 0);

- Build wait_mask

done_ef_mask = one_bit_mask(done_event_flag);

wait_mask = done_ef_mask | RT_M_SKEL_TIMER;

One_bit_mask creates a 128 bit mask with 1 bit set to indicate which event.

Here, the done_event_flag (such as RT_V_BOTTLENECK_START) is set by the function that was started by SYS\$SETEF. It performs a bitwise OR, so that wait_mask turns on when the function completes or times out. Note that _M is used here, not the _V used above in SYS\$SETIMR.

- SYS\$WFLOR sits and waits for either of two event flags, the function times out or is completed. Note that the _V is used here, same as in WAITIMR, so the _M is not redundant in wait_mask.

cond_code = SYS\$WFLOR(RT_V_SKEL_TIMER, wait_mask);

- Read the event flags

cond_code = SYS\$READEF(RT_V_SKEL_TIMER, &efc_bits

This command sets efc_bits.

- Check if process done bit is set. The done_ef_mask is a 1 bit mask for done_event_flag. If this bit is still on after bitwise and with efc_bits, then process was done. Cancel timer with SYS\$CANTIM and clear done bit with SYS\$CLREF.

if ((efc_bits & done_ef_mask) == done_ef_mask)

- Check if timeout bit is set. If RT_M_SKEL_TIMER bit is still set after bitwise and with efc_bits, then timeout occurred. If so, clear start bit with SYS\$CLREF, clear done bit with SYS\$CLREF, and issue timeout error message.

else if ((efc_bits & RT_M_SKEL_TIMER) == RT_M_SKEL_TIMER)

ERROR HANDLING

FDDDB_ERROR [fddb/fddb_sub.c]

Called from `get_inc_det_eqn`, `get_btl_neck`, `eqn`, and `get_stn_aggr_eqn`. Will be called from `get_fuzzy_eqn` too.

- Builds error message.
- Calculate number of columns by `src_ptr - field_start` where `field_start` is current column in source. `src_ptr` not used for anything else.
- `find_err_text` (in `fddb_sub.c`) finds text for that error code (error code may be returned from `get_token`), such as invalid char, invalid term, invalid parameter, etc. If `NO_ERR_CODE`, search is skipped.
- If `fddb_error_mode` is `FILE_TERMINAL` or `FILE_DETACHED`, write error message to `fddb_error_file`. Note that `fddb_error_mode` is set to `FILE_TERMINAL` in `build_rmdb`.
- Unless `FILE_DETACHED`, write error message to screen.

Error handling functions called from taps

Note on event codes. The event codes are in hex. The upper 4 hexadecimals indicate which function sent the message [defined in `tms_include/event_common.h`]. The lower two bits indicate the condition code severity [defined in `tms_include/tms_system.h`]. The lower 4 hexadecimals, excluding the last 3 bits (the word must end in either a 0 or 8), are used as the error code. The 1st nibble of the error code is always 8, although it is not clear what this means. The error codes are only unique within the main, not system wide. They become unique because they are bitwise or'd with the function event codes.

`build_tap_error` -- called from `get_fuzzy_eqn`

`log_tms_event` -- sends message to *TMS Event Logger*

Pass in `event_code` and `msg_txt`.

Note on event codes: The event codes are defined in the beginning of the main taps programs, such as `bottleneck.c`. Examples of event codes include `FUZ_BUILD_ERR`, `FUZ_COMPLETE`, `FUZ_MAP_GBL`, `FUZ_OVERFLOW`, `FUZ_START`, `FUZ_ST_COMPL`, `FUZ_TERMINATE`, etc.

`log_tms_common` -- The `module_name` is a global initialized at the beginning of main
`write_to_mailbox_nowait`

`log_tms_event_cc` -- sends message with condition code to *TMS event logger*

Pass in `event_code`, `msg_text`, and `cond_code`.

`log_tms_common`
`write_to_mailbox_nowait`

`log_tms_mpu_name`

write_to_mailbox_nowait

Error handling functions in comm_prot

log_comm_event - Log a message to the *TMS Event Logger*

Passed variables include error_code (which is 0x00000000 if not appropriate or no error?), unit_no, and error_msg

```
/* Flag Bit Usage in mpu_no:  
 *  COMM_LOG_M_MODULE_NAME Use module name for logging  
 *  COMM_LOG_M_PORT_NO Low 16 bits are a Port Number  
 *  COMM_LOG_M_FORCE_FILE Force write to log file, even if disabled  
 *  COMM_LOG_M_NO_FILE No logging to file, even if enabled  
 *  COMM_LOG_M_FORCE_SCR Force write to screen, even if disabled  
 *  COMM_LOG_M_NO_SCREEN No logging to screen, even if enabled  
 */  
/* Note: the FORCE bits have precedence over the NO bits */
```

log_comm_event_cc - Log a message with a condition code to the *TMS Event Logger*

```
get_mpu_name  
  get_port_name  
  get_unit_name  
write_to_crash_log  
fmt_write_comm_msg  
  close_comm_log_file  
  create_comm_log_file  
write_to_crash_log
```

log_comm_msg - Log a message to *Comm Handler Log File*

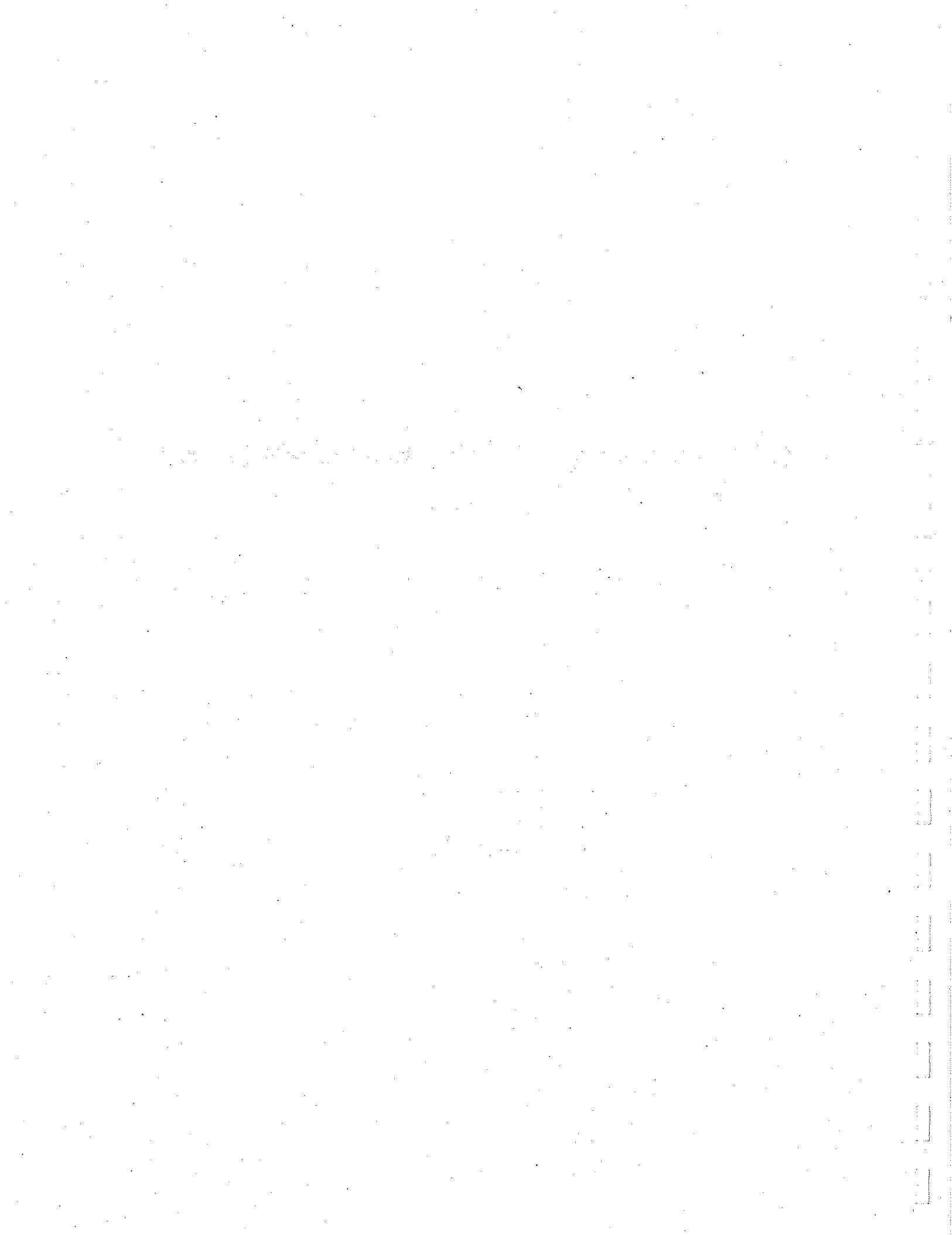
Pass event code, mpu_no (module/port/unit # w/Flag bits), msg to write

Always called from a comm handler.

```
get_mpu_name (see above)  
write_to_crash_log  
fmt_write_comm_msg (see above)
```

log_comm_msg_cc - Log a message with a condition code to *Comm Handler Log File*

VI. TRAFFIC ANALYSIS PROGRAMS (TAPS)



BOTTLENECK

bottleneck [./rt_skeleton/bottleneck.c] starts up main, builds bottleneck analysis table, and then waits for event flag from real time skeleton to do calculation

- general_process_startup
 - connect_to_mailbox
 - write_to_crash_log
 - Associate to event flag cluster
 - Clear all event flags
- log_tms_event
 - log_tms_common
 - write_to_crash_log
 - get time
 - compose message
 - write_to_mailbox_nowait
- map_to_RTDB
 - map_to_global_section
 - init_rtdb_tl
- map_to_RMDB
 - map_to_global_section
 - init_rmdb_tl
- build_bottleneck_table (see below)
- While SHUTDOWN_TMS event flag is clear
 - Wait for bottleneck start event flag with SYS\$WAITFR
This flag gets set in rt_skeleton by SYSS\$SETEF
 - Clear bottleneck start flag with SYS\$CLREF
 - calc_bottleneck (see below)
 - Set bottleneck done flag with SYSS\$SETEF
 - Test shutdown flag with SYS\$READEF
cond_code of SS\$_WASCLR means don't shutdown
cond_code of SS\$_WASSET means shutdown

build_bottleneck_table - Parses the bottleneck equation file line by line, searches for cabinet name in RMDB, roadway type, reversible type, station names in RTDB, and writes it all to bottleneck analysis table. (see following bottleneck table description)

- Initial memory allocation for bottleneck table
- write header to table: start table label, number of sets, table size, date/time
- open bottleneck equation file
- Initialize counters, pointers, and line type
- While not EOF, read line of equation file

Example of 1 set:

*ES-170D:_MN_Btl=ES-170D:_MN_Stn+ES-170D:_MNH_5-ES-172R:_MN_Stn-
ES-172R:_MNH_5-ES-172R:_MN_X_1
Wts:ES-159R= 51,ES-151R= 36,ES-146R= 13*

- Switch based on type of line:
 - 1) SKIP_EQN -- do nothing.
EQN_FIRST is next action.
 - 2) EQN_FIRST -- beginning of an storage rate equation.
Calculate pointer to beginning of set
Get index in RMDB for cabinet name (7 chars) with **find_fddb_cl_name**
classify_roadway returns roadway_type (7 possibilities of form *_ROADWAY) by comparing 3 chars after : (MMS in this case)
check_reversible returns rv_type (NB, SB, or NR of form *_SET_START) by comparing chars [9,10] of line buffer (MS in this case is nonreversible).
It's used to indicate the beginning of a set.

Write begin set label to table

Calculate pointer to station in RMDB, later used to get occupancy threshold.

add_sub_code = BA_ADD_VOL initially

Initialize # loops and # weights/ equation

EQN_CONTINUE is next action

- 3) EQN_CONTINUE -- parse a station in storate rate equation.

Get station name from line buffer

Search for station name (16 chars) in RTDB with **search_rtdb_name_table**

and find index to it. This index will be used to get volume from this station

Calculate pointer to station in RTDB

Write station pointer to table

Increment # of loops

If + sign is next

BA_ADD_VOL -- add in volume for this station to calculate storage rate

If more stations on line, repeat

else EQN_CONTINUE and get next line

else if - sign is next

BA_SUB_VOL -- subtract volume for this station to calculate storate rate

If more stations on line, repeat

else EQN_CONTINUE and get next line

else done reading line -- WTS_FIRST and get next line

4) WTS_FIRST -- beginning of weights line.
Check to make sure line begins with "Wts:"
WTS_CONTINUE

5) WTS_CONTINUE -- Parse a cabinet name in weights line.
Get index in rmdb for cabinet name with **find_fddb_cl_name**
Calculate pointer to station in RMDB
Get weight from line buffer and write to table
Increment # of weights
If more weights on same line -- repeat for next station
else more weights continued on next line --
 get next line and then WTS_CONTINUE
else no more stations --
 check for errors, then EQN_FIRST

- Close equation file
- Write number of sets and table size into table header
- Write table end label, check sum (calc. # bytes in table)
- Trim table size

BTN_TABLE -- Description of what it looks like in memory

LABELS	ITEM	# BYTES	DESCRIPTION
table_base->	code	1	BA_TABLE_START (0xF2)
	ushort	2	number of sets
	ulong	4	table size
	date_time	struct size	struct system_time (end of table header)
			(beginning of sets)
set_start -> (Also EQN_FIRST)	code	1	rv_type (NB, SB, or NR of form *_SET_START)
	byte	1	bytes in set
	code	1	roadway_type (NB, NC, NH, SB, SC, or RV of form *_ROADWAY)
	ulong	4	col_ptr for cabinet name in RMDB (end of set header)
			(begin SR stations)
EQN_CONT	code	1	add_sub_code (BA_ADD_VOL for the first one)
	ushort	2	rtdb_offset to station name in RTDB (end of station)
Repeat above (add_sub_code and rtdb_offset) for every station in storage rate calc for this set			
			(done with SR stations)
			(begin ramp weightings)
WTS_FIRST	code	1	wt_code -- numerical but char string This is both the action code (<100) and the weight itself.
	ulong	4	col_ptr to cabinet name with above weight (end of a ramp weighting)
Repeat above (wt_code and col_ptr) for every cabinet affected by this set's calc. rate adj			
			(done with ramp weightings)
			(label end of set)
	code	1	BA_SET_END -- code for end of set where 1 set for each metering rate calc (end of a set)
			(begin next set)
Repeat for each set (every ramp that has bottleneck metering), including set header			
			(end of all sets)
			(label end of table)
	code	1	TABLE_END
	short	2	check sum - # of bytes in table (end of table)

calc_bottleneck -- processes the bottleneck analysis table 1 line at a time to calculate bottleneck adjustment rate, which is sent through data poll. Gets cabinet index, checks bottleneck flags, checks occupancy threshold. Calculates storate rate. Sums weights, calculates metering rate adjustment for each cabinet the bottleneck affects. If the new adjustment is more restrictive, use it in place of the old adjustment. Store the name of the bottleneck cabinet that changed the metering rate for each cabinet.

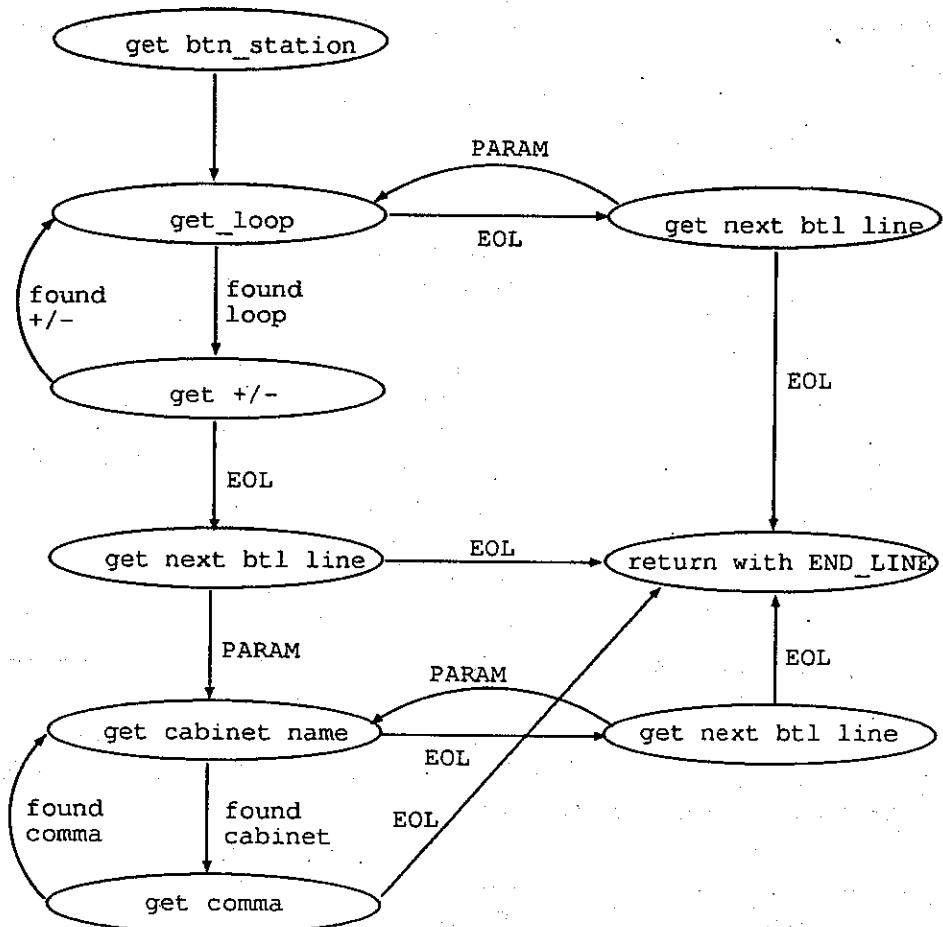
- For all data columns in RMDB
 - Skip if data column for min, max, default or prot mask
 - Initialize the metering rate to minimum adjustment
 - Initialize the rate adjustment index to bogus -- not set by any cabinet
- Initialize pointer to beginning of bottleneck table
- Skip past table header
- While not end of table
 - Get action code from table
 - If action code < 100, it is a weight. Get weight and reset action code to WEIGHT
 - Switch for action code:
 - 1) Reversible type (3 forms of form BA_*_SET_START)
This is the beginning of a set -- initialize counters, flags
 - 2) Roadway type (7 types of form *_ROADWAY) -- get RMDB parameters.
 - Get index for current station
 - Get flag for incident forcing bottleneck calculation
 - Get flag for operator permitting bottleneck calculation
 - Get occupancy threshold for bottleneck calculation
 - 3) BA_ADD_VOL or BA_SUB_VOL -- for each station in storage rate equation
 - Calculate pointer to station in RTDB for calculating storage rate
 - unpack_rtdb_loop_stn**
 - If first station
 - If bottleneck was not forced on -- check that bottleneck conditions are met
 - If operator disabled bottleneck, skip to next equation set
 - If no usable data, skip to next equation set
 - If not enough good data, skip to next equation set
 - If not enough loops, skip to next equation set
 - If occupancy is less than bottleneck threshold, skip to next set
 - Accumulate storage rate with station volume
 - 4) WEIGHT -- for each weight used to calculate metering rate
 - If first weight in equation, save pointer to start of weights
 - Add to accumulate sum of weights
 - Increment # of upstream ramps affected by this bottleneck
 - Note: Get next weight and skip cabinet name for now -- get later
 - 5) BA_SET_END -- when finished summing weights
 - Calculate storage/sum_weights
 - Initialize a pointer to start of weights
 - For each weight in bottleneck calculation
 - Get weight
 - Calculate metering rate = weight*storage/sum_weights

- Calculate pointer to adjusted cabinet in RMDB
- Get index to name of adjusted cabinet
- Check for overflow
- Get metering rate that may have been set by previous bottleneck calculation
- If new metering rate adjustment is greater (more restrictive) than previous
 - update ramp metering rate for that cabinet
 - update index to current cabinet name (which set metering rate)
- 6) TABLE_END -- return (this exits infinite loop)

GUIDE TO ACTION CODES FOR CALC_BOTTLENECK

ACTION	DEFINED (in taps.h)	DESCRIPTION
BA_TABLE_START	0xF2	Beginning of Bottleneck Analysis table ----- <i>One of these three reversible types is first byte in each set-----</i>
BA_NR_SET_START	0xC0	Non-reversible set start
BA_NB_SET_START	0xC1	Northbound set start
BA_SB_SET_START	0xC2	Southbound Set Start ----- <i>One of these 7 roadway types is 3rd byte in each set-----</i>
NB_ROADWAY	0xB0	Northbound Mainline Lanes of form *MN***
NC_ROADWAY	0xB1	Northbound Collector/Distributor of form *CN***
NH_ROADWAY	0xB2	Northbound HOV Lanes of form *MNH***
SB_ROADWAY	0xB3	Southbound Mainline lanes of form *MS***
SC_ROADWAY	0xB4	Southbound Collector/Distributor
SH_ROADWAY	0xB5	of form *MSH***
RV_ROADWAY	0xB6	Reversible lanes of form *R_*** ----- <i>For each station in storage rate equation, begin with one of these codes-----</i>
----- <i>followed by index to station in RTDB -----</i>		
BA_ADD_VOL	0xD4	Add this volume to calc. storage rate
BA_SUB_VOL	0xD5	Subtract this volume to calc. storage rate ----- <i>For each weighted ramp in metering rate equation-----</i>
WEIGHT	<100	Both weight and action code
----- <i>-----</i>		
BA_SET_END	0xC3	End of a set
----- <i>-----</i>		
TABLE_END	0xFF	End of calculation table
----- <i>-----</i>		

READ BOTTLENECK EQUATION



WATCH_BOTTLENECK

`watch_bottleneck` is started offline, although event flags are created in `tms_startup`.

- `process_equation_file` opens `btl_neck.eqn` (the same file created by `build_rmdb` and used by `bottleneck.c`)
 - Duplicates the code in `bottleneck.c` that reads each line, parses each equation, searches for cabinet name in data base, classifies roadway type, check reversibility, process weights line, wrx.
- Sorts the cabinet list in ascending order
- Initialize storage of calculations
- A menu appears on screen and user chooses from `print_cabinet_list`
- Write equations onto screen
- Sit and wait for tick from `rt_skeleton`.
 - `display_vol_data` -- do metering rate calculations for each station/loop. If occupancy is not usable, `interpolate_metering_curve`
 - `highlight_max_rate`
 - `display_history_data` -- displays last times, cabinet names, and rate adjustments
 - Prompt to exit

INC_DET_TABLE -- Description of what it looks like in memory

LABELS	ITEM	# BYTES	DESCRIPTION
<i>(begin table)</i>			
table_base->	code	1	ID_TABLE_START (0xF1)
	ushort	2	number of sets
	ulong	4	table size
	date_time	struct size	struct system_time
<i>(end of table header)</i>			
<i>(beginning of sets)</i>			
	code	1	rv_type -- start set of this reversible type (NB, SB, or NR of form *_SET_START)
	byte	1	bytes in set
	code	1	ID_THIS_STN (0xD2)
	ushort	2	offset to this station in RTDB (to get occ)
	code	1	ID_DWNSTR_STN (0xD3)
	ushort	2	offset to downstream station in RTDB (to get occupancy)
	code	1	roadway_type (NB, NC, NH, SB, SC, or RV of form *_ROADWAY)
	ulong	4	col_ptr to cabinet in RMDB (to get k1, k2, and k3 parameters)
	byte	1	ID_SET_END -- code for end of set
<i>(end of a set)</i>			
<i>(begin next set)</i>			
Repeat for each set (every incident detection site)			
<i>(end of all sets)</i>			
<i>(label end of table)</i>			
	code	1	TABLE_END
	short	2	check sum -- # of bytes in table
<i>(end of table)</i>			

Incident Detection

inc_detect [./rt_skeleton/inc_detect.c] starts up main, builds incident detection table, and then waits for event flag from real time skeleton to do test

- **general_process_startup**
 - **connect_to_mailbox**
 - **write_to_crash_log**
 - Associate to event flag cluster
 - Clear all event flags
 - **log_tms_event**
 - **log_tms_common**
 - **write_to_crash_log**
 - get time
 - compose message
 - **write_to_mailbox_nowait**
 - **map_to_RTDB**
 - **map_to_global_section**
 - **init_rtdb_tl**
 - **map_to_RMDB**
 - **map_to_global_section**
 - **init_rmdb_tl**
 - **build_inc_det_table** (see below)
-
- **While SHUTDOWN_TMS event flag is clear**
 - Wait for inc_detect start event flag with SYS\$WAITFR --
This flag gets set in rt_skeleton by SYS\$SETEF
 - Clear inc_detect start flag with SYS\$CLREF
 - **calc_inc_det** (see below)
 - Set inc_detect done flag with SYS\$SETEF
 - Test shutdown flag with SYS\$READDEF
cond_code of SS\$_WASCLR means don't shutdown
cond_code of SS\$_WASSET means shutdown

build_inc_det_table - Parses the incident equations input file, searches for indices to data for upstream and downstream stations (to get occupancies later) in RTDB and to cabinet (to get threshold parameters later) in RMDB. Writes action codes and data pointers to incident table (see description on next page).

- Initial memory allocation for incident table
- Write header to table: start table label, number of sets, table size, date/time (see description of functions that write to table)
- Open incident detection equation file
- Initialize counters
- While not EOF, read next line of equation file

Example of 1 set:

- *ES-158R:MMS_Inc = ES-158R:MMS_Stn & ES-156R: MS_Stn*
- Get index in rmdb for cabinet name (7 chars) with **find_fddb_cl_name** -- Used later to get k1, k2, and k3 thresholds for that cabinet
- **classify_roadway** returns roadway_type (7 possibilities of form *_ROADWAY) by comparing 3 chars after : (MMS in this case)
- **check_reversible** returns rv_type (3 possibilities of form *_SET_START) by comparing chars [9,10] of line buffer (MS in this case is nonreversible)
- Get this station name from line buffer (15 chars)
- Get index in RTDB for this station name with **search_rtdb_name_table** -- this index will later be used to get occupancy data from this upstream station
- Get downstream station name from line buffer (15 chars)
- Get index in RTDB for downstream station name with **search_rtdb_name_table** -- this index will later be used to get occupancy data from this downstream station
- Check for errors
- Calculate bytes in set
- If not enough memory for new set in table, allocate more memory the size of ADDTL_ALLOC
- **Write reversible type** (of form *_SET_START) and **bytes in set** to table
- Calculate offset for this station in RTDB
- **Write ID_THIS_STN code and offset in RTDB** to table
- Calculate offset for downstream station in RTDB
- **Write ID_DWNSTR_STN code and offset in RTDB** to table
- Close incident equation file
- **Write # of sets to table**
- **Write table size to table**
- **Write TABLE_END code to table**
- Calculate **checksum** (make sure table size matches calculated size) and write to table
- Trim table size
- Calculate table base

calc_inc_det -- processes the incident detection table 1 line at a time to obtain 3 threshold parameters, the previous incident detection state, and volumes for upstream and downstream stations.

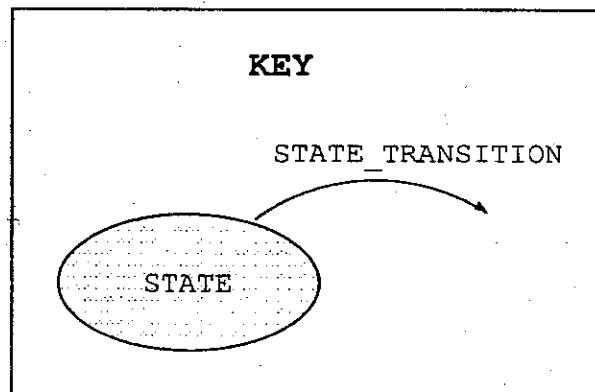
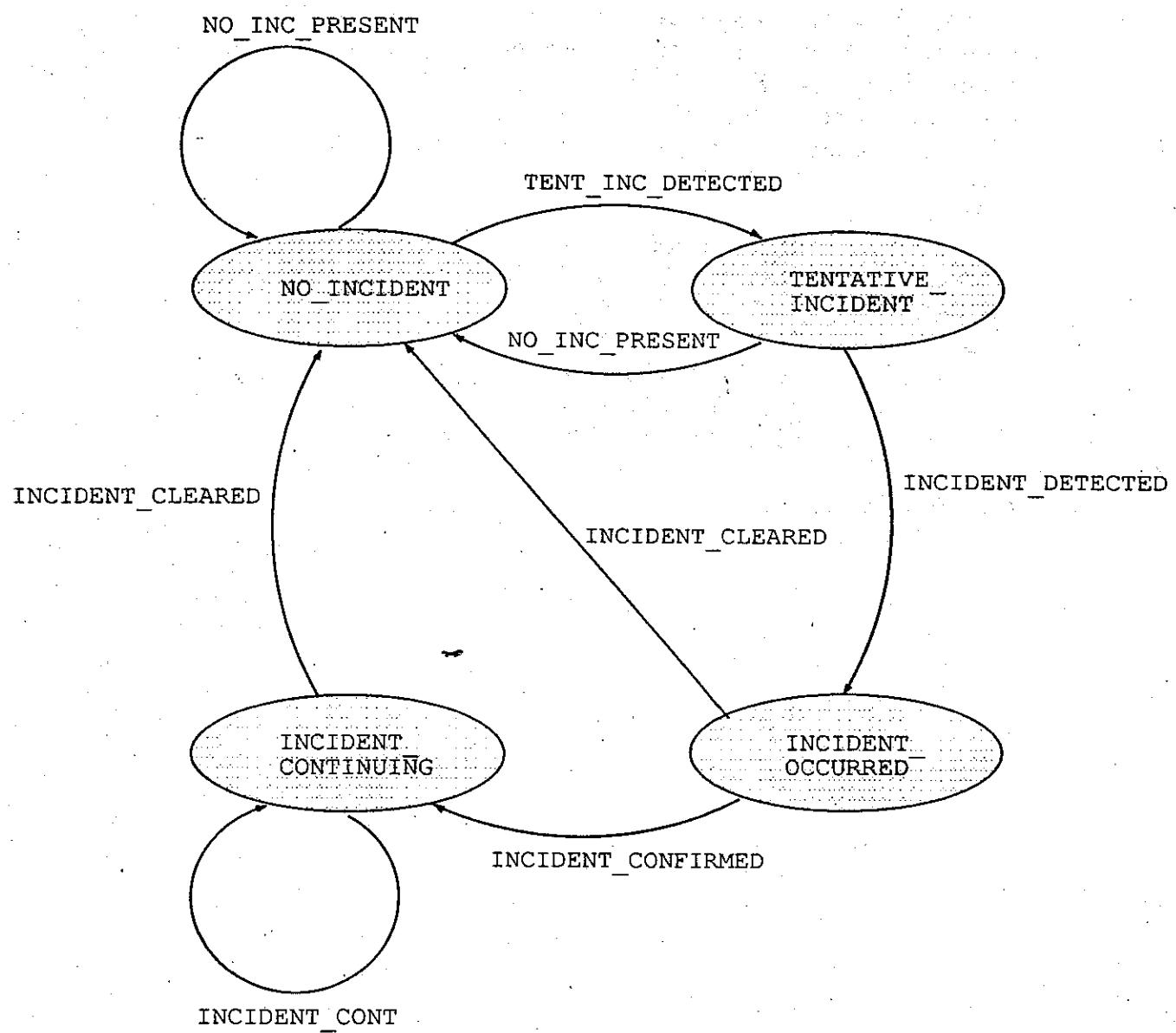
- Initialize pointer to beginning of incident table
- Skip past table header
- While not TABLE_END
 - Get action code from table
 - Switch for action code:
 - 1) Rv_type of form ID_*_SET_START (3 types) -- This is the beginning of a set. Skip past bytes_in_set and continue.
 - 2) Roadway type (7 types) -- get RMDB parameters. The structure element name varies for different road types, which is why there is a different action code for each road type.
 - Get index for current station
 - Get occupancy difference parameter
 - Get relative occupancy difference parameter
 - Get downstream occupancy threshold parameter
 - 3) ID_THIS_STATION
 - Get pointer to "new" data column in RTDB
 - Get new this_station occupancy
 - Get pointer to "current" data column in RTDB
 - Get PrevIncDetState from current data column using **unpack_rtdb_loop_stn**
 - Get current this_station occupancy
 - Get previous incident detection state
 - Get pointer to "minus20" data column in RTDB
 - Get minus20 this_station occupancy
 - Calculate average occupancy from past three samples
 - 4) ID_DWNSTR_STN
 - Get pointer to "new" data column in RTDB
 - Get new dwnstr_station occupancy
 - Get pointer to "current" data column in RTDB
 - Get current dwnstr_station occupancy
 - Get pointer to "minus20" data column in RTDB
 - Get minus20 dwnstr_station occupancy
 - Calculate average occupancy from past three samples
 - 5) ID_SET_END
 - Calculate k1, k2, and k3 -- incident detection parameters
 - Get state change from
 - state_change = **incident_detect** (ThisStnOcc, DwnstrStnOcc,
PrevIncDetState, &NewIncDetState, k1, k2, k3);
NewIncDetState is set by incident_detect.
 - Get var_ptr for station in RTDB, new data column
 - Put the new state into RTDB using **pack_rtdb_inc_det**
 - Get index to name of station given offset in RTDB --
 - using **search_rtdb_offset_list**
 - Switch based on state transition

- 1) NO_INC_PRESENT (stay in NO_INC state)
Do nothing
- 2) TENT_INC_DETECTED (from NO_INC to TENT_INC)
Write state change and message to log (compose message that contains roadway, direction, milepost, location, and station name)
- 3) INCIDENT_DETECTED (from TENT_INC to INC_OCCURED)
Write state change and message to log
- 4) INCIDENT_CONFIRMED (from INC_OCCURED to INC_CONT)
Write state change and message to log
- 5) INCIDENT_CONT (stays in INC_CONT)
Write state change and message to log
- 6) INCIDENT_CLEARED (INC_CONT to NO_INC)
Write state change and message to log

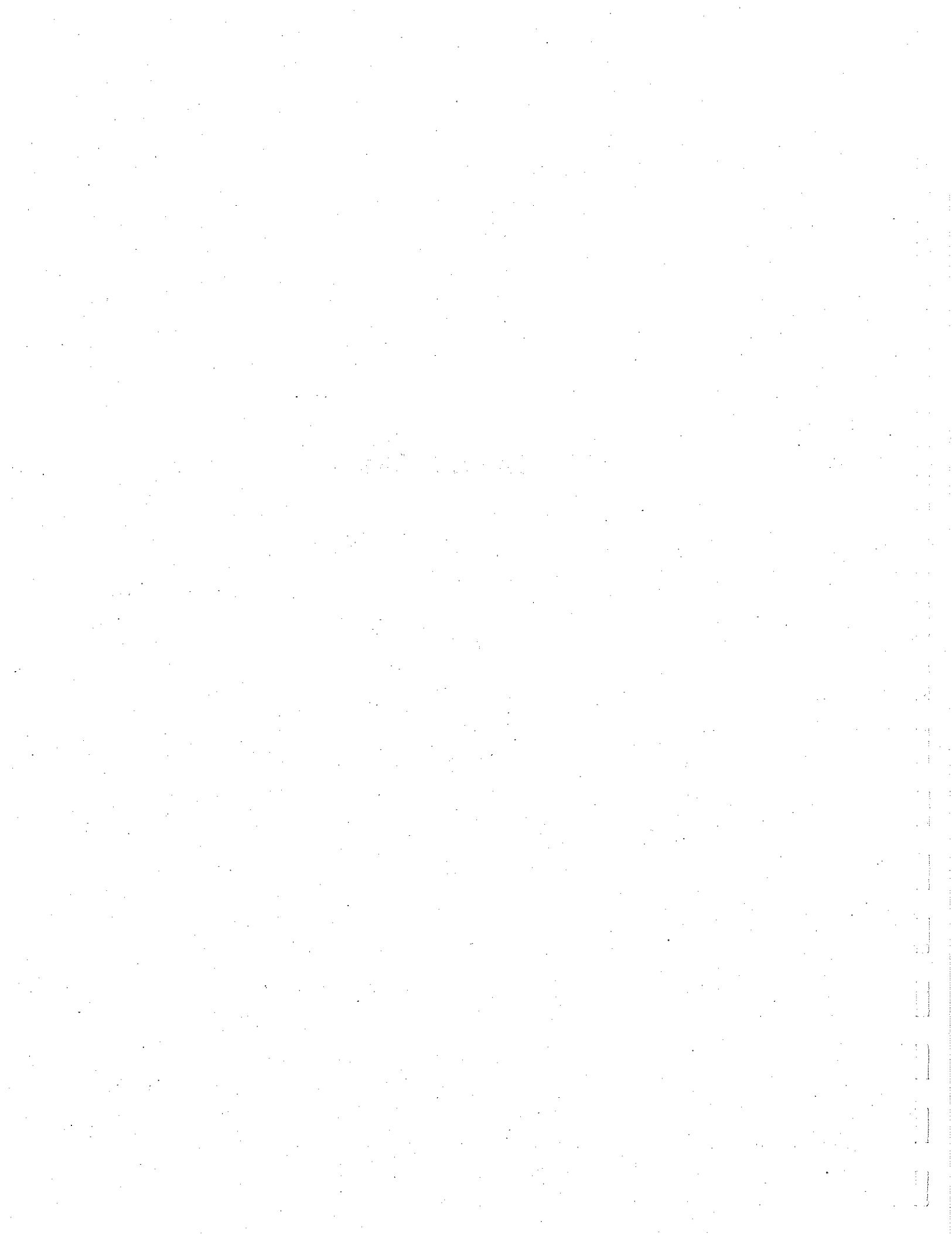
incident_detect -- Performs actual incident detection algorithm. Three checks to see if an incident has occurred. Updates state (the pointer to new state was passed so change is effective in calc_inc_det, although it still needs to be put into RTDB) and returns transition (see state transition diagram)

- Calculate occupancy_difference between stations = upstream - downstream
- If occupancy_difference > k1 (upstream greater than downstream by 10%)
 - Calculate relative_occupancy = occupancy_difference/this_stn_occupancy
 - If relative_occupancy > k2 (40)
 - If previous state was NO INCIDENT
If relative_occupancy < k3 -- Extra qualification to reach TENTATIVE
new state = TENTATIVE_ACCIDENT
 - Else
new state = NO INCIDENT
 - Else if previous state was TENTATIVE_INCIDENT
new state = INCIDENT_OCCURRED
 - Else if previous state was INCIDENT_OCCURRED
new state = INCIDENT_CONTINUING
 - Else previous state was INCIDENT_CONTINUING
new state = INCIDENT_CONTINUING
- Else
new state = NO INCIDENT

INCIDENT DETECTION STATE TRANSITION DIAGRAM



V. DATABASES



DATA BASE BUILD

To build all of the data bases in Table 1, run build_all_db.com

Table 1: Files Related to Off-line Data Base Builds

Mains for Offline Builds	Input file that is Used	Temp File that is Created
build_cctvdb	cctvdb_input.fil	
build_gbldb	gbldb_input.fil	
build_gcdb	gcdb_input.fil	
build_oprtvdb	oprtvdb_input.fil	
build_rmdb	rmdb_input.fil	loop_names.lst speed_traps.lst station_names.lst btl_neck.eqn* inc_det.eqn* stn_aggr.eqn* actv_anal.eqn* rtfmdbname.srt
build_scheddb	scheddb_input.fil	
build_vaxportdb	none	
build_vmsdb	vmsdb_input.fil	

*Note: these files are used by the Traffic Analysis Programs (bottleneck, incident detection, and station aggregation) to create a table of pointers to the RTDB.

Table 2: Files Related to On-line Data Base Builds

Main for Online Build	Function that calls it	Temp File that is Used
build_fmdb	tms_startup	rtfmdbname.srt
build_rtbd	tms_startup	rtfmdbname.srt

DEFAULT SPECIFICATION IN RMDB

There are two stages of default specification:

- 1) Upon compilation, the Default data column in /fdedb/rmdb/rmdb_tbl.c is initialized into global memory (likewise for the other "Predefined" data columns which include Minimum, Maximum, and EP_MASK). The fdedb_name_table is initialized here as well. The fdedb_name_table structure contains default values, but these simply references the ones in the Default data column.
- 2) During off-line run of build_rmdb, the rmdb_input.fil is processed. The beginning of this file contains "User Defined Default" columns (see next page). Only the exceptions to the Default data column (from rmdb_tbl.c) need to be specified; which reduces the complexity of the rmdb_input.fil. The values listed here supercede those of the Default data column. If "MAXIMUM, MINIMUM, DEFAULT, or EP_MASK" columns are defined, these values supercede those that were compiled in step 1), and are used as the Predefined for all data columns.

"User Defined Default" columns (see list on next page) are then used as defaults for the remaining cabinets in the rmdb_input.fil (to make it easier to specify parameters for a similar group of cabinets). Only the exceptions to the "User Defined Default" columns needs to be specified. These exceptions supercede those listed in the "User Defined Default" columns. Only the group names for which there are exceptions *need* to be listed; however it is OK to list all of them. Group names must be in the order specified in the HNTB documentation (pg 121). The RMDB is built from the rmdb_input.fil, so the cabinet names must be given even if no exceptions are listed.

Defaults Columns used in RMDB_INPUT.FIL

This table indicates which data columns use which user-defined default column. The user defined default columns appear without an equal sign, such as { PM-020 } while a column which uses a default will appear with an equal sign such as {ES-TR1R = PM-020}.

Note: these are not permanent -- this table is from 4/96.

<u>User Defined</u>	<u>Default Column</u>	<u>Columns which use it for default parameters</u>
	PM-003	none
	PM-020	ES-TR1R ES-116R ES-117R ES-137 ES-146R ES-504Rn ES-506R ES-822 ES-881 ES-889R
	PM-023	none
	PM-123	ES-TR8R ES-TD1R PM-123 ES-TR8R ES-TD1R ES-TD1R
	PM-103	ES-TR7R ES-TR7R
	PM-120	ES-TR6R ES-188
	PM-100	ES-TR4R ES-WS2D ES-139R ES-151R ES-168R ES-175R ES-825R
	A45-123	ES-689R ES-730R
	A45-120	ES-716R ES-726R
	A45-100	ES-681R ES-710R ES-740R
	AM-003	ES-TR2R
	AM-020	ES-134R ES-136R ES-141R ES-156R ES-158R ES-163R ES-174R ES-187R ES-207R
	AM-023	none
	AM-100	ES-149R
	AM-103	none
	AM-120	ES-181R ES-193R ES-203R
	AM-123	none

NoMeter	ES-TR3D ES-TD1D ES-TD2D ES-TD3D ES-WS1D ES-059D ES-068D ES-069D ES-074D ES-079D ES-086R ES-087R ES-088D ES-090D ES-092R ES-093D ES-094R ES-098D ES-100R ES-101R ES-102R ES-104D ES-105D ES-108D ES-111R ES-118R ES-121R ES-123D ES-124D ES-125R ES-126D ES-128D ES-130D ES-132D ES-143D ES-145D ES-148D ES-152D ES-154D ES-159R ES-161D ES-165D ES-167D ES-170D ES-172R ES-177D ES-179D ES-186D ES-189D ES-201D ES-213D ES-264D ES-265D ES-502D ES-514D ES-516R ES-519R ES-520D ES-521R ES-522R ES-524R ES-525R ES-528D ES-531R ES-533D ES-535D ES-538D ES-539D ES-540D ES-541D ES-542D ES-543D ES-544D ES-610D ES-612D ES-614D ES-616R ES-617D ES-619D ES-621D ES-622D ES-623D ES-625D ES-626D ES-628D ES-630D ES-632D ES-633D ES-634R ES-638R ES-642R ES-647R ES-651D ES-653R ES-660D ES-662R ES-667D ES-672D ES-676D ES-677D ES-678D ES-682R ES-684D ES-687R ES-693R ES-694R ES-696D ES-698D ES-704D ES-706D ES-708D ES-709D ES-711R ES-717R ES-720D ES-722D ES-724D ES-731R ES-734D ES-736D ES-738D ES-739D ES-741R ES-742D ES-761D ES-810D ES-812D ES-814D ES-818D ES-820D ES-826D ES-827D ES-852D ES-854D ES-855D ES-857D ES-858D ES-860D ES-861D ES-878D ES-883D ES-885D ES-891D ES-896D ES-900R ES-903D ES-910D ES-912D ES-916D ES-924D ES-928D ES-932D ES-940D ES-992D ES-994D
I90-100	ES-920R
I90-020	ES-863R ES-876R ES-879R ES-887R ES-893R
I90-120	ES-900R ES-908R ES-935R ES-945R

STRUCTURE OF RMDB (Ramp Metering Data Base)

<u>STRUCTURE NAME</u>	<u>ARRAY SIZE</u>	<u>DESCRIPTION</u>
1) rmdb_offsets		Offsets to other items in RMDB
2) rmdb_parameters		Contains parameters for all groups Approx. 300 parameters in each data column
3) fddb_line_buffer		Contains pointer to current source Position in source file, position of comments, work_buffer
4) fddb_group_table	19 group names	Group names Internal_Controls, 170_Data, 170_Parameters_at_VAX, 170_Global_Parameters, Data_Validity_Parameters, Loop_table, Speed_traps, Lane1_parameters, Lane2_parameters, Lane3_parameters, Station_aggregation_parameters, Incident_detect_parameters, Bottleneck_analysis_parameters, Activation_analysis_parameters
5) fddb_output_list	approx 300	Group names and parameter names
6) fddb_data_col_list	n_data_columns	Name of each field device (beginning with min, max, default, and ep_mask), index to default data column, file location of data column, line number in file, relative pointer to data column num_nt_entries Names of elements in name table, type, size, flag, and offset to this data element
7) fddb_name_table		Contains each element in data column (approx 300)
8) rm_dc_data_col	n_data_columns	Array of loop names for each 170
9) loop_name_list	n_data_columns	

BUILD_RMDB

Build_rmdb opens and reads rmdb_input.fil, builds RMDB, creates temporary files (loop_names.lst, inc_det.eqn, btl_neck.eqn, speed_traps.lst, stn_aggr.eqn, station_names.lst, actv_anal.eqn) which are later used to build tables in global memory for traffic analysis programs. It also sorts names, loops, stations and speeds traps and writes them to "rtfdb dbname.srt" to be used for later creation of RTDB and FMDB. Build_rmdb appears deceptively simply, but in fact, starts a long chain of events, calling function upon function.

BUILD_RMDB [/fdedb/rmdb/build_rmdb.c]

- Declare variables -- Most of these are globals defined in fdedb.h or rmdb.h
- The three fdedb_state_table structures (**fst_cnt_dcol**, **fst_id_param**, **fstr_read_jnl**) are initialized in the file fdedb_fst.c upon compilation. These are state transition tables that contain pointers to functions called from rmdb_fddb_file! The other two tables (**fst_tbl_name** and **fst_grp_cmt**) are used from **patch_rmdb** when the user chooses to write rmdb_output.fil.
- Initialize global sort buffer parameters. Sort_bfr, max_sort_bfr, n_loops, n_stations, n_speed_traps, and n_names are globals that are written by **write_to_sort_bfr** [/fdedb/rmdb/rmdb_sub.c] when called from **write_speed_traps_to_file**, **get_stn_aggr_eqn**, and **write_loop_name_list_to_file**.
- Open rmdb_error.fil, which is later written to in sub-functions called from **read_fddb_file**.

Note: There are several fopen parameters unique to VMS. "mbf=2" is for multi-buffer count. This equals 2 for all files. "mbc=69" is for multiblock count, the number of blocks varying from file to file. "rop=wbh" is for record processing operations, and WBH means write behind. In files that are read from, "rop=RAH" for read ahead.

- Open the rmdb_input.fil
- Initialize the file pointer in the file_list (a global struct in rmdb.h but not actually part of RMDB. However, the rmdb_table_list does point to this structure, so this information is available wherever rmdb_t1 is passed.)
- Initialize the pointers in **rmdb_t1** (often called **t1** in sub-processes) of struct type **rmdb_table_list**. This global struct defined in rmdb.h contains *absolute* pointers to each *table* within the RMDB. Although rmdb_t1 is not actually part of the RMDB, it is essentially a quick pointer reference to everything in RMDB except rmdb_offsets and rm_dc_data_col. It contains 3 pointers to parts not in the RMDB, including rmdb_base, rmdb_file_pointers, and rmdb_stn_aggr_table. (The **rmdb_offsets**, on the other hand, are the 1st part of the RMDB and contain *relative* offsets to each part of the RMDB)
- The # of data columns is initialized to 4 to account for the Predefined data columns (min, max, default, and ep_mask). These rm_dc_data_col are predefined upon compilation, initialized in rmdb_tbl.c. The number of data columns is later incremented in the function **new_column** of /fdedb/fddc_sub.c.
- Initialize the temporary line buffer. This struct, **fdedb_line_buffer**, is the 3rd item in the RMDB. It's used in fdedb/rmdb/read_fddb_file and get_next_line to parse the current line of rmdb_input.fil

- Count the # of data columns in rmdb_input.fil with call to **read_fddb_file** (see process description for details). The argument **fst_cnt_dcol** is what tells the **read_fddb_file** to simply count data columns rather than load them. Whenever a curly brace is found, **fst_cnt_dcol** points to the function **cnt_data_col**, where the **data_col_list_size** is incremented. This is the first of 3 calls to **read_fddb_file**, passing a different function state table (**fst**) each time.
- Determine total size of RMDB in bytes. Three structs in the RMDB are an array of size **n_data_columns**: **fddb_data_col_list**, **rm_dc_data_col**, and **loop_name_list**. The remaining structs in the RMDB are of fixed size upon compilation. (see document “Structure of RMDB” for description).
- **map_to_global_section** [/tms_library/global_sub.c] to see if RMDB already exists in global memory. **map_to_global_section** builds descriptors for global section name, then maps to global system using system utility **SYS\$MGBLSC**
`SYS$MGBLSC(inadr, gbl_addr, PSL$C_USER, flags, &gblnam, &ident, 0);`
inadr is start and end of address to map to. *gbl_addr* is the start and end address that were mapped to. *ident* is the version # of global section, where an exact match is required. If returned *cond_code* == **NORMAL**, then RMDB already exists, in which case **unmap_global_section** and **delete_all_global_section**. **unmap_global_section** deletes virtual access space using system call
`cond_code = SYS$DELTVA(gbl_addr, retadr, PSL$C_USER);`
gbl_addr is beginning and end address to be deleted, and *retadr* returns beginning and end addresses of section actually deleted. **PSL\$C_USER** sets access mode so that privileges are needed to delete global section. **delete_all_global_section** builds descriptors and uses system utility **SYS\$DGBLSC** to mark global section for deletion.
- **create_global_section** [/tms_library/global_sub.c]
`cond_code = create_global_section(section_name, bytes_required, gbl_addr);`
The section name was initialized to “TMS_RMDB.” **create_global_section** builds descriptors for global section name and creates RMDB using system utility **SYS\$CRMPSC**, which associates a specified section of address space with specified physical addresses (both creates and maps). *gbl_addr* is an array returned by **SYS\$CRMPSC** containing actual beginning and ending address of RMDB in global memory. If returned *cond_code* == **SS\$NORMAL**, then the global section already existed and was mapped to -- the previous delete did not work properly because there is another process that mapped to the old RMDB. Successful creation returns **SS\$CREATED**.
- Calculate the *bytes_provided* from returned *gbl_addr*, and make sure it is less than *bytes_required*.
- If **TEST_RMDB** is defined, create RMDB in process memory only (not global), so that it's deleted when application finishes running.
- Initialize the RMDB using the absolute pointer in **rmdb_tl**
 - `init_rmdb_tl(gbl_addr[0], &offsets, &rmdb_tl);` [in /tms_library/fddb_lib.c]
Calculates *absolute* pointers to each table in RMDB. *base_address* is passed in as *gbl_addr[0]*. Absolute pointers = *base_address* + relative offsets (which were previously calculated when determining size of RMDB).
 - `memcpy(gbl_addr[0], &offsets, sizeof(struct rmdb_offsets));`

Relative offsets were previously calculated when determining size of RMDB.

Copy offsets struct to beginning of RMDB. This is 1st item in RMDB.

- `init_rmdb_params(&rmdb_tl, n_data_columns, num_nt_entries);`
(these are `rmdb_tl->pm_ptr->...`). Item 2 in RMDB.

Call `init_fddb_params` to handle FDDB parameters.

Initialize parameters unique to RMDB:

 data base type = RMDB_TYPE, sum_metered_lanes = 0

 For RTDB and FMDB, set n_names, n_loops, n_stations, and n_speed_traps to 0.

- `memcpy(rmdb_tl.gt_ptr, &group_table, size_group_table);`
Global array of `group_table` was initialized in `rmdb_tbl.c` upon compilation.
 - Copy group table into RMDB. Item 4 in RMDB.
 - `memcpy(rmdb_tl.ol_ptr, &output_list, size_output_list);`
Global array of `output_list` was initialized in `rmdb_tbl.c` upon compilation.
 - Copy `output_list` into RMDB. Item 5 in RMDB.
 - `init_rmdb_data_col_list(&rmdb_tl, rmdb_db_base);`
For each field device (`data_col_list_size` was previously determined by 1st call to `read_rmdb_file`), set the name to null, dft_column, file_loc, and lineno to -1. The name and `dft_column` are updated later in `new_column`. Calculate offset to the `rm_dc_data_col` corresponding to each field device.
 - `memcpy(rmdb_tl.nt_ptr, &name_table, size_name_table);`
Name_table is initialized in `rmdb_tbl.c` upon compilation.
 - Copy info for each element into RMDB. Item #7 in RMDB.
 - `memset(rmdb_tl.ln_ptr, NUL, n_data_columns * sizeof(struct loop_name_list));`
Calculate size of `loop_name_list` and set to NULL in RMDB. The loop name table for each data column is later updated in `get_pin_assignments`. Item #9 in RMDB.
 - Skip `rm_dc_data_col` for now. They are subsequently built by sub-function `new_column`.
 - Reinitialize line buffer. `rmdb_tl->lb_ptr` was already set to `temp_lb` for 1st call to `read_fddb_file`. Item #3 in RMDB.
-
- **calc_offsets** [in /fddb/fddb_sub.c]
converts the compile time pointers to relative offsets (stored in `rmdb_tl.nt_ptr[i].offset`) for each name table element by subtracting out `ct_default` (base address of default `rm_dc_data_col` at compile time)
 - Copy the Predefined data columns (`ct_minimum`, `ct_maximum`, `ct_default`, and `ct_ep_mask` exists as global variables initialized in `rmdb_tbl.c` at compile time) into the RMDB data columns. Initialize the global pointers `ct_minimum_ptr`, `ct_maximum_ptr`, `ct_default_ptr`, and `ct_ep_mask_ptr`. The number of data columns is updated to its minimal 4, and is later incremented during `new_column` function.
 - Open the temporary files that will be written to by `read_rmdb_file`'s subfunctions. These files include `LOOP_NAMES.LST`, `STATION_NAMES.LST`, `SPEED_TRAPS.LST`, `STN_AGGR.EQN`, `INC_DET.EQN`, `BTL_NECK.EQN`, and `ACTV_ANAL.EQN`. The equation files are later used during startup of the traffic analysis programs to build tables in memory.
 - `read_fddb_file` (see function description for details) reads `rmdb_input_file`, creates data columns and writes to temporary files. The argument `fst_id_parameters` contains the function

- state tables, indicating what function to call given the delimiter type returned by `get_next_line` [/fddb/fddb_sub.c]. This is the 2nd call to `read_fddb_file`.
- `load_name_table_ndx` [/fddb/fddb_sub.c] calls `find_fddb_nt_name` to get the offset to the “`_NameTableNdx`” field in the `fddb_name_table` (the RMDB struct that contains element names and relative offsets to them). For each data column, the “`_NameTableNdx`” field is set to the numerical index for that data column (between 0 and `n_data_col_list`).
 - Close temporary files.
- Open file `RMDB_JOURNAL.FIL`, which contains real-time changes that were made to the RMDB from the operator console during previous operation of TSMC.
- Call `read_fddb_file` to read `rmdb_jnl_file`, using function state table `fst_read_jnl`. This is the 3rd call to `read_fddb_file`. `fst_read_jnl` points to `find_col_name` when it encounters a curly bracket. It points to `find_grp_name` when it encounters a square bracket. However, the journal files do not contain group names, only column names and changes made. When it encounters a parameter, it points to `get_param`, which should update the RMDB. However, this code does not work for some reason. `get_param` does work correctly when used with `fst_Id_param`, so the problem is probably not within `get_param`. Possibly the problem is that the current group is incorrect. For further details on journal file, see `rmdb_journal.frm`.
- Set global line buffer input to file terminal (disables generation of debugging output file).
- Call `mem_sort_rtfmdb_names` [/fddb/rmdb/rmdb_sub.c] to sort the station names, speed traps, and loops in ascending order using system call
`qsort(sort_bfr, n_names, (RTDB_NAME_LENGTH+1), rtfm_compare);`
The arguments `sort_bfr` (containing names) and `n_names` are globals that were previously updated by `write_to_sort_bfr` [/fddb/rmdb/rmdb_sub.c] when called from `write_speed_traps_to_file`, `get_stn_aggr_eqn`, and `write_loop_name_list_to_file`. Sorted list is written to “`rtfmdbname.srt`” and used in `build_rtdb.c` to create RTDB.

READ_FDBD_FILE [in build_rmdb.c]

- Called 3 times from build_rmdb main. read_fdbd_file reads each line in rmdb_input.fil with call to `get_next_line`, then searches state table until it finds the current state and line_type. The corresponding function is called. The behavior of read_fdbd_file depends entirely on what input file it is reading and what function state table is used. This documentation will describe the behavior for the load parameter function state table because it is by far the most complex.
- Initialize current state to 0 index of state table (corresponding to NO_COLUMN). The function state table was initialized in fdbd_fst.c upon compilation. The state table name was passed in as an argument.
 - Initialize line buffer (global struct in RMDB).
 - While not end of rmdb_input.fil
 - `line_type = get_next_line(tl, fdbd_file);`
Returns line_type of comment, curly_brace, square_bracket, or parameter. (See function description for details). Line is stored in tl->lb_ptr->line_buffer
 - `setjmp` saves stack environment for later use by `longjmp`. If error occurred in `get_next_btl_line`, then `longjmp` restores environment.
 - While not end of state table
 - Search table until the current state and line_type match those indexed in table. The table contains an array of structs of the form `{state, line_type, err_msg, before, function_pointer, after, and new state}`. *Before* indicates how to process comments before the function call, and *after* indicates how to process comments after the function call. This search is linear, always starting at the beginning of the table and scanning until it finds the state/line_type or until end of table.
 - Process comments before function call based on `fst[i].before`
 - If within comment block, update end of comment block
 - If CMT_PREV, save comment block in previous name/group
 - Otherwise skip comment processing
 - Get pointer to function from state table.
 - Call function. For `fst_id_parameters`, this can be `new_column`, `complete_new`, `grp_name_fnd`, `get_param`, or `complete_col`. See function descriptions for details.
 - Process comments after function call based on `fst[i].after`
 - If INIT_CMT, save location as start of a comment block
 - If CLR_CMT, clear comment block variables
 - Otherwise no comment processing
 - Get new state from function state table. States include NO_COLUMN, NOCOL_CMT, COL_NAME, GRP_NAME, PARAM_ST, CMT_STATE, CMT_BLANK, FF_STATE, and END_TABLE.
 - Save last line type in `line_buffer->prior_line`
 - If current state is DONE_FIL, return. Otherwise exit inner loop (go read next line). DONE_FIL becomes next state if the line_type encountered is end of file.

GET_NEXT_LINE [/fdbd/fdbd_sub.c]

Called from read_fdbd_file to read next line of input file. Returns line_type: END_FILE,

FORM_FD, BLANK_L, COMMENT, CURLY_BR, SQ_BKT, or PARAM. The line_type constants [defined in /fdedb/fddb.h] consist of 8 bits (constants 0x11 through 0x17). If the 8th bit is set, this indicates a truncated record.

- Get current line of input file and store in line buffer
- Increment line number in rmdb_tl->lb_ptr->line_no
- Initialize work buffer pointer and offset
- Check for end of file
- Calculate length of line
- Check line byte by byte for a form feed.
 - If "f" is found, store the FF position in file
 - If FF is first byte in line, adjust file position to continue after FF and return FORM_FD
 - Else adjust file position to reread FF and continue processing
- Check to see if record is truncated --
 - If line doesn't end with "\n", record is bad -- set truncated_record bit flag to TRUNC_REC
 - Otherwise, line is not truncated -- set flag to 0x00.
- BLANK_L returned if line is null
- COMMENT returned if line's first nonblank char is ";"
- CURLY_BR returned if line's first nonblank char is "{"
- SQ_BKT returned if line's first nonblank char is "["
- PARAM returned otherwise

FUNCTIONS CALLED FROM READ_FDDB_FILE

Read_fddb_file calls several functions. Which functions are called depends on the function state table, an argument passed to read_fddb_file from build_rmdb. The function state table provides a pointer function to process the current state and line_type from the input file that is being read.

The table below indicates the name of each function state table, what function calls read_fddb_file, and what functions that table uses.

<u>Table Name</u>	<u>Called from</u>	<u>Functions called</u>
fst_cnt_dcol	build_rmdb	cnt_data_col
fst_ld_param	build_rmdb	new_column, complete_new, grp_name_fnd, get_param, complete_col
fst_tbl_name	build_cctvdb, build_gbldb, col_name_fnd build_gcdb, build_oprtvdb build_vmsdb	
fst_grp_cmt	build_cctvdb, build_gbldb build_gcdb, build_oprtvdb build_vmsdb	chk_col_name, grp_name_fnd
fst_read_jnl	build_rmdb	find_col_name, find_grp_name, get_param

DESCRIPTION OF FUNCTIONS CALLED FROM READ_FDDB_FILE

cnt data col [/fddb/fddb_sub.c] -- increments the # of data columns whenever a line begins with “{“

- Call **column_name_line** [/fddb/fddb_sub.c] to crack ‘table_name = default’ into its three parts. Returns either P_NAME_FOUND, P_NAME_EQUAL_FOUND, or P_NAME_VALUE_FOUND
- Increment data_col_list_size (a parameter in RMDB) unless it is one of the four Predefined data columns.

new column [/fddb/fddb_sub.c] -- creates new data column, loads default values whenever line begins with a “{“, and increments n_data_col_list_size.

- Call **column_name_line** [/fddb/fddb_sub.c] to crack ‘table_name = default’ into its three parts. Returns either P_NAME_FOUND, P_NAME_EQUAL_FOUND, or P_NAME_VALUE_FOUND
- If data column is one of 4 Predefined data columns, don’t create a new data column
 - set current_tl_index to predefined index

- **init_group_table** to predefined data column (from current_tl_index) -- see function description below.
- **new_column_special_case** -- Initializes station aggregation list and sets DataSwitch for current column (see function description below for details)
- Make sure that n_data_col_list (counted in this function) is less than the n_data_col_list (previously counted in 1st call to read_fmdb_file)
- Copy table_name (of form ES-*****) into data column
- Update current_tl_index to n_data_col_list
- **init_group_table** -- This is a temporary table to store location of group names in rmdb_input.fil. It is reused for each data column and contains data for only one data column.
 - Copy current column name (last read in new_column) into the group table gt[0].name.
 - Set current group index to zero -- this points to current column name
 - For cabinet (gt[0]), set before_pos, name_pos, and after_pos to file position in line buffer
 - Initialize group names (1 through end of group table)
 - Set before_pos, name_pos, and after_pos to -1
- Find default_index
 - If no default name was given, use Predefined default.
 - Otherwise, search for the provided default_table_name using **find_fddb_el_name**, which returns an index to a data column. Set default index to the user specified data column.
- Specify default_index in data column and copy defaults to data column.
- Increment n_data_col_list
- **new_column_special_case** -- Initializes station aggregation list and sets DataSwitch for current column
 - **init_stn_aggr_list** [/fdb/rmdb/rmdb_sub.c]
 - Get pointer to station aggregation list
 - For each station of list
 - Set station name to null
 - Set # of loops to 0
 - Set all loop names to 0 for that station
 - Calculate pointer to current data column
 - If one of Predefined data columns, set data switch to NOT_RM_DC and return
 - Set data switch to RAMP_MTR, DATA_STN, or NOT_RM_DC based on cabinet name
 - Get cabinet name from line buffer. Check char 6 to determine data switch type.
 - Set col_ptr->_DataSwitch

complete_col [/fdb/rmdb/rmdb_sub.c]

Called when another "{" is found or end of file -- finishes previous data column

Return if Predefined column

Otherwise, call several functions (for details, see function descriptions elsewhere):

- **write_loop_name_list_to_file**
Get pointer to current data column
If the data column is for a ramp meter/data station,
- **count_active_loops** -- counts NActiveLoops for a data column
For all loops in 170
- **count_metered_lanes** -- counts NMeteredLanes and MLanesMask for a data column

- **set_roadway_type_bits** -- set bit of RoadTypeMask according to roadway type
- **count_speed_traps** -- count the # of active speed traps for a given data column and write to _NSpeedTraps
- **match_speed_loops** [/fdedb/rmdb/rmdb_sub.c] -- For all speed traps in a data column, find matching upstream loop for a given downstream loop and write to data column
- **write_speed_traps_to_file** -- For a particular data column, writes speed traps to file speed_traps.lst and to global sort buffer.
- **build_dft_stn_aggr_eqns** -- fills in the struct rmdb_stn_aggr_table for a particular data column. This consists of copying in the station name, # of loops in 170, and the loop names.
- **write_dft_stn_aggr_eqns_to_file** -- For each station in a given data column, copy station and loops from station aggregation table to temp buffer. Then write entire equation to stn_aggr.eqn.

complete_new [/fdedb/fddb_sub.c] -- combined functions of complete past column and start new column. Called when next "{" is encountered in rmdb_input.fil

- **complete_col** (see above description)
- **new_column** (see above description)

grp_name_fnd [/fdedb/rmdb/rmdb_sub.c] -- Called when a "[" is encountered in rmdb_input.fil. Searches for group name in fddb_group_table (which contains 19 group names) to obtain group index, stores *current* group index in pm->grp_ndx, and stores file position for comment block. This is later used in **process_input_special_case** (see **get_param** below) to get group type.

- Call **get_token** to extract group name from square bracket string
- Call **find_fddb_gt_name** to return group index for given group name
- Set gt[group_index].before_pos to to start of comment block before group name
- Set gt[group_index].name_pos and .after_pos to current file position in line buffer
- Call **grp_name_special_case** -- this function does nothing for BUILD_RMDB

get_param [/fdedb/fddb_sub.c] -- Called to process parameter line. Loads parameter(s) in data column of RMDB.

- While not end of line, process parameters
 - Call **get_token** to find_next_parameter in line buffer -- this just skips over blanks and commas to point to parameter
 - Call **process_input_special_case** to check for special cases based on current grp_index and return ld_result:
 - If PIN_ASSIGNMENTS, **get_pin_assignments** (see function description)
 - If SPEED_TRAP_PARAMS, **get_speed_trap_params**
 - If TIME_OF_DAY_TABLE, **get_tod_entry** (see function description)
 - If STN_AGGR_EQNS, **get_stn_aggr_eqn** (see function description)
 - If INC_DETECT_EQNS, **get_inc_det_eqn** (see function description)
 - If BTLNECK_ANAL_EQNS, **get_btl_neck_eqn** (see function description)

- If ACTIVATION_ANAL_EQN, get_actv_anal_eqn
- Note: In future code, If FUZZY_PARAMETER, get_fuzzy_params
- Note: In future code, If FUZZY_EQNS, get_fuzzy_eqn
- Otherwise not a special case
- If not a special case, call load_param to put parameter into data column (see function description)
- If successful load (ld_result == DONE), continue processing line
- Otherwise, handle error

find_col_name [/fddb/fddb_sub.c] -- Called during journal file read. Parses a column name from the input line, searches for index to that column in RMDB, and updates pm->current_tl_index.

- Call column_name_line to crack input line into 3 parts: 'Column_Name = Default'
- Search for column_name in RMDB and return index to it with call to find_fddb_cl_name.
- Update current index

find_grp_name [/fddb/fddb_sub.c] -- Called during journal file read. Parses a group name from the input line, searches for the group name in group table, updates current group index, pm->grp_ndx.

- Call get_token to extract group name from square brackets of input line.
- Search for group name in group table (which contains 19 names) with call to find_fddb_gt_name and return index to group
Note that not all of names in group table are used in RMDB. Stn_aggr_eqns, inc_detect_eqns, btlneck_anal_eqns, and activation_anal_eqns are groups not included in RMDB.
- Update current group index
- Call grp_name_special_case -- Does nothing for BUILD_RMDB

DESCRIPTION OF SUBFUNCTIONS CALLED BY COMPLETE_COL

- **write_loop_name_list_to_file** [/fdedb/rmdb/rmdb_sub.c] -- Write loop name to file "Loop_names.lst" and to global sort buffer.
 - Get pointer to current data column
 - If the data column is for a ramp meter/data station,
 - For all loops in the 170
 - Write the loop name to file_list.loop_name_file (called "LOOP_NAMES.LST")
 - Write the loop name to global sort buffer, sort_bfr with call to **write_to_sort_bfr**
- **count_active_loops** [/fdedb/rmdb/rmdb_sub.c] -- counts NActiveLoops for a data column
 - For all loops in 170
 - Check if function exists for that loop in col_ptr->loop_table[i].function
 - If nonzero, then increment # of active loops
- **count_metered_lanes** [/fdedb/rmdb/rmdb_sub.c] -- counts NMeteredLanes and MLanesMask for a data column
 - Get pointer to current data column
 - For all loops in 170
 - If 4th and 5th character of loop name are " _P"
 - Increment # of metered lanes
 - Check if char 6 of loop name is 1, 2, 3
 - set corresponding bit in MLanesMask (an 8 bit flag)
- **set_roadway_type_bits** [/fdedb/rmdb/rmdb_sub.c] --set bit of RoadTypeMask according to roadway type
 - Get pointer to data column
 - Initialize RoadTypeMask to 0x00
 - For all loops in 170
 - **classify_roadway** type for loop (7 possible types of form **_ROADWAY)
 - Set bit in col_ptr->_RoadTypeMask for roadway_type -- each roadway type has a corresponding bit in _RoadTypeMask
- **count_speed_traps** [/fdedb/rmdb/rmdb_sub.c] -- count the # of active speed traps for a given data column and write to _NSpeedTraps
 - Get pointer to data column
 - For all speed traps in that data column
 - If speed trap has a nonzero downstream loop
 - Increment # of speed traps
 - Write # of speed traps to col_ptr->_NSpeedTraps
- **match_speed_loops** [/fdedb/rmdb/rmdb_sub.c] -- For all speed traps in a data column, find matching upstream loop for a given downstream loop and write to data column
 - Get pointer to data column
 - For all speed traps in data column
 - Search through loop in 170 until find upstream loop that corresponds to a given down-

- stream loop.
- Write matched upstream loop to col_ptr->speed_traps[i].upstr_loop
- **write_speed_traps_to_file** [/fdedb/rmdb/rmdb_sub.c] -- For a particular data column, writes speed traps to file speed_traps.lst and to global sort buffer.
 - Get pointer to data column
 - Make sure data column is of type RAMP_MTR or DATA_STN
 - For all speed traps in that data column, write to file "speed_traps.lst"
 - Write station and loop name, "speed_trap", and index for that speed trap within 170.
Example: "ES-TD1R:MMN_T1 SPEED_TRAP ; Speed Trap 1"
 - Write speed trap name to global sort buffer with call to **write_to_sort_bfr**
- **build_dft_stn_aggr_eqns** [/fdedb/rmdb/rmdb_sub.c] -- fills in the struct rmdb_stn_aggr_table for a particular data column. This consists of copying in the station name, # of loops in 170, and the loop names.
 - Get pointer to data column
 - Make sure data column is a RAMP_MTR or DATA_STN, otherwise return.
 - For all loops in 170
 - **classify_roadway** type
 - Convert the roadway_type from to an index by subtracting NB_ROADWAY, the first constant in listing of roadway types
 - If the current loop_name is a speed trap (char 5 equals 'S'), don't include it
 - If num_loops is set to -1, this means equation was specified in rmdb_input.fil.
Don't build a default equation for this station.
 - If num_loops equals zero, copy station name from line buffer to station aggregation table
 - Copy loop name from line buffer to station aggregation table
 - Increment # of loops for this station
 - Error if # of loops exceeds LOOPS_PER_STATION, the max allowed.
- **write_dft_stn_aggr_eqns_to_file** [/fdedb/rmdb/rmdb_sub.c] -- For each station in a given data column, copy station and loops from station aggregation table to temp buffer. Then write entire equation to stn_aggr.eqn.
 - Get pointer to data column
 - Make sure data column is RAMP_MTR or DATA_STN, otherwise return
 - For all stations (for a given point on the roadway, it is possible to have up to seven roadway types: NB, NC, NH, SB, SC, SH, and RV)
 - For all loops in that station
Copy loop_name from station aggregation table to a temp buffer
 - Write station name (15 chars) =loop_name1+loop_name2, etc...

Example: "ES-TR1R:MMN_Stn=MMN_1+MMN_2+MMN_3+MMN_4"

DESCRIPTION OF SUBFUNCTIONS CALLED BY GET_PARAM

get_pin_assignment [/fddb/rmdb/rmdb_sub.c] -- Parses line of form 'pin_name =parameter_value' and stores loop name, loop function, and pin # in loop name table for that data column.

- Get pointer to current data column (current_index was set by new_column)
- Call **get_token** to get parameter name from input line (will be "pin1" through "pin40")
- Handle special case for EP_MASK
- Call **get_token** to get parameter_value (7 chars such as "_MMNRA_2") from input line
- Return if current data column is one of Predefined -- pin assignment not valid
- Convert pin # to from ascii to integer
- If param_value is "NONE"
 - set loop name to null
 - set pin_function to 0
 - Set speed trap loops to 0
- Check for duplicate pin #
- Copy param_value from line buffer into loop name table for that data column (this writes over null initialized in build_rmdb)
(tl->line_ptr into loop_name[pin_no][0])
- Set loop function and pin # within that data column
- If DISABLED, set loop_status to DISABLE
- If pin is for speed trap, make sure speed trap is OK.

get_tod_entry [/fddb/rmdb/rmdb_sub.c] -- processes a tod entry of rmdb_input.fil and writes into data column

- Get parameter name from input line with **get_token**
- Obtain index to parameter name with call to **find_fddb_nt_name** (it returns the relative offset to that element from the col_ptr)
- Calculate pointers to data column, tod_ptr, min_tod_ptr, and max_tod_ptr for that data column
- Get equal sign from input line with **get_token**
- If current index equals the Predefined data column EP_MASK_INDEX
 - Get ep_mask
 - set tod_ptr->hour to ep_mask
- Get time parameter (hour and minute) from line
- Make sure time parameter is greater than min_time and less than max_time
- get "on" from input line
- get DOW (day of week) from input line
- get "@"
- get rate
- If rate_buffer equals "OFF", set rate to 0x00.
- Else if rate_buffer equals "Traffic", set rate to 0xFF

- Else set rate to rate_buffer converted from ascii to integer
- Set hour, min, dow, and rate for tod entry in data column.

get_stn_aggr_eqn [/fddb/rmdb/rmdb_sub.c] -- processes a station aggregation equation from rmdb_input.fil and writes it to files "stn_aggr.eqn", "station_names.lst", and to global sort buffer. It doesn't write these to RMDB. Stn_aggr builds the station aggregation table from "stn_aggr.eqn".

Get pointer to data column

- Make sure data column is of type RAMP_METER or DATA_STN. Otherwise return.
- Check that station name in aggregation equation matches the one give for this data column
- **classify_roadway** given station name -- returns 1 of 7 road_types of form **_ROADWAY
- Convert roadway type to an integer index by subtracting NB_ROADWAY, the first in listing of roadway type constants
- Initialize # of loops in station aggregation of data column to -1 to prevent default equation from being built later. This is later incremented in **build_dft_stn_aggr_eqns**, which is called from **new_column**
- Get equal sign from input line
- DONE if no loops
 - Read each loop name until end of line
 - Check for errors. Return if too many loops, loop name doesn't match those given under pin assignments for that cabinet, or not at least 1 loop in equation.
 - Build equation (format is same as in rmdb_input.fil) to a temp buffer
- Write temp buffer to file "stn_aggr.eqn"
- Write station name to file "station_names.lst"
- Write station name to global sort buffer with call to **write_to_sort_bfr**

get_inc_det_eqn [/fddb/rmdb/rmdb_sub.c] -- processes incident detection equation in rmdb_input.fil and writes it to file inc_det.eqn. These parameters are not written to RMDB. Instead, inc_detect builds the incident detection table from "inc_det.eqn."

Get pointer to current data column

- Parse input line to get cabinet name with call to **get_cab_loop_name**
- Parse input line to get equal sign with call to **get_token**
- Parse input line to get this_stn with call to **get_next_loop_name**
- Check that station name matches the one that was given in cabinet line of input file
- Get dwnstr_stn of incident detection pair from input line with call to **get_cab_loop_name**
- Build equation in temp buffer (same format as that in rmdb_input.fil)
- Print temp buffer to file "inc_det.eqn"

get_btl_neck_eqn [/fddb/rmdb/rmdb_sub.c] -- read bottleneck equation from input line and write bottleneck and weight equations to file "btl_neck.eqn". These parameters are not written to RMDB. Instead, bottleneck builds the bottleneck table from "btl_neck.eqn."

Get pointer to current data column

- Initialize indices for loops and weights to zero
- Get bottleneck station from input line (15 chars)
 - Get cabinet name with call to `get_token`
 - Get loop name with call to `get_token`
- Make sure the bottleneck station name matches the one most recently read from `new_column`
- Get equal sign and put in loop operator
- Read all loops from bottleneck equation (see diagram "Read Bottleneck Equation")
- Read all weights from weights equation (see diagram "Read Bottleneck Equation")
- In case of `END_LINE`, `get_next_btl_line`
- Write bottleneck equation to "btl_neck.eqn"
- Write weight equation to "btl_neck.eqn"

load_param [/fddb/fdadb_sub.c] -- called from `get_param` if not one of the special cases. Gets parameter name and value from input line and puts into data column

- Get parameter name from input file with call to `get_token`
- Search for parameter name in element name table with call to `find_fdadb_nt_name` and return offset relative to beginning of data column
- Calculate pointer to parameter in data column by adding the column pointer (which was passed in as an argument) to relative offset
- Select parsing table based on field type. Field types (0x00-0x49) are found in `tms_system.h`

NOTE: The field type names containing a 1 are scaled by 0.1 when they're used. These include `UBYTE1`, `UBYTE1P`, `SBYTE1P`, `USHORT1`, `USHORT1P`, `SSHORT1`, and `SSHORT1P`. This way they save memory because they're not stored as floats. Although elements of this type are premultiplied by 10 in the Predefined Data Columns in `rmdb_tbl.c`, they are entered as floats in `rmdb_input.fil`. This factor of 10 difference is accounted for in the parse table struct (see `tok_tabl.c`). The parse table for these field types tells `get_token` to write over the decimal point, modifying the input line buffer. The effect is multiplying the input by 10 and converting from a float to an integer. That's why the subsequent call to `atol` in `load_param` is able to convert from ascii to an integer.

- Get parameter value from input line with call to `get_token` with appropriate parse table
- Process parameter -- switch based on parameter type
 - Cast parameter name's pointer to proper type
 - Make sure parameter value is within acceptable range with call to `range_check`. `Range_check` compares with the Minimum and Maximum Predefined Data Columns for that element (not specific to field type).
 - Write parameter value to RMDB

STRUCTURE OF RTDB (Real Time Data Base)

STRUCTURE NAME	ARRAY SIZE	DESCRIPTION
1) rtdb_offsets	1	Offsets to other items in RTDB
2) rtdb_params	1	Creation time, db_type, # processes, # of name table entries, etc
3) rtdb_reg_process	1	Process ID's, bit masks to track processes
4) unsigned long	NUM_RTDB_COLUMNS	offsets for RTDB data ptrs
5) rtdb_name_table	rtfm_n_names	element name, type, size, and offset to data
6) data_col	NUM_RTDB_COLUMNS	

BUILD_RTDB

BUILD_RTDB maps to RMDB, calculates size of RTDB, allocates memory for RTDB, and initializes RTDB. See description of RTDB structure in rtdb_struct.frm. The structures are initialized in tms_include/rtdb.h.

BUILD_RTDB [/build_rtdb.c]

- Started from tms_startup and executed once
- maps to RMDB
- Calculate n_bytes_data_elems (total # of data bytes/sample) by added # of loops, # of stations, and # of speed traps, which were tallied by BUILD_RMDB.
- Calculate bytes required for RTDB and offsets to items in RTDB
 1. rtdb_offsets to other items in RTDB
 2. rtdb_params -- oddball creation parameters
 3. rtdb_reg_process -- bit masks
 4. array of offsets to RTDB data_cols
 5. rtdb_name_table -- array of element names, type, size and offset
 6. data_col -- real time data
- map_to_global_section [/tms_library/global_sub.c] to create RTDB
- create_global_section for RTDB
- Make sure enough memory to fit RTDB
- Initialize RTDB
 1. copy offsets (that were calculated previously) into RTDB
 2. Initialize the FMDB parameters
 3. Set bit masks
 4. Calculate offsets for each data column
 5. Call load_rtdb_name_table [in build_rtdb.c]. Opens "rtfmdbname.srt", which was created by BUILD_RMDB and contains a sorted list of every loop, station, and speed trap. Initializes the rtdb_name_table. For each element, initializes type, element size, element offsets, and element name.
 6. Zero out all of the data columns

Related Functions:

- The NEW data column is updated when run_polling_processes [in rt_skeleton/rt_skeleton.c] calls multi_rmdc_comm [in comm_prot/rmdc_comm]
- scroll_rtdb_col_offsets [in tms_library/rtdb_lib.c] makes room for the latest data by moving data columns down one. It is called from rt_skeleton after TAPS have completed.
 - rtdb_cl is an array of offsets to data columns. Rather than moving the data itself, the data offset is incremented by one. The offset to the last data column is relocated to the first offset so that the same memory can be reused.
 - The first data column is initialized to zero (not the offset itself, but where it points to).
 - read_rtdb.c [rt_skeleton/read_rtdb.c] is called from rt_skeleton every 20 seconds, but does nothing! It was put there in anticipation of future processing.