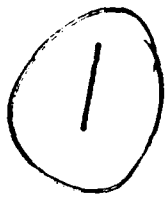
AD-A244 976



SOFTWARE DESIGN DOCUMENT
SAF Simulation Host CSCI (8)

Volume 2 of 2 Sections 2.7.1 - 2.15 and Appendices

DTIC
ELECTE
JAN 09 1992
S D
D

June, 1991

Prepared by:

BBN Systems and Technologies,
A Division of Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, MA 02138
(617) 873-3000 FAX: (617) 873-4315

Prepared for:

Defense Advanced Research Projects Agency (DARPA)
Information and Science Technology Office
1400 Wilson Blvd., Arlington, VA 22209-2308
(202) 694-8232, AUTOVON 224-8232

Program Manager for Training Devices (PM TRADE)
12350 Research Parkway
Orlando, FL 32826-3276
(407) 380-4518

92-00248

92 1 6 056

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

SOFTWARE DESIGN DOCUMENT
SAF Simulation Host CSCI (8)

Volume 2 of 2 Sections 2.7.1 - 2.15 and Appendices

June, 1991

Prepared by:

BBN Systems and Technologies,
A Division of Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, MA 02138
(617) 873-3000 FAX: (617) 873-4315

Accesion For
NTIS CRA&I
DTIC TAB
Unannounced
Justification
By
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

Prepared for:

Defense Advanced Research Projects Agency (DARPA)
Information and Science Technology Office
1400 Wilson Blvd., Arlington, VA 22209-2308
(202) 594-8232, AUTOVON 224-8232

Program Manager for Training Devices (PM TRADE)
12350 Research Parkway
Orlando, FL 32826-3276
(407) 380-4518

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

REPORT DOCUMENTATION PAGE

Form Approved
OPM No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Information and Regulatory Affairs, Office of Management and Budget, Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1991	Software Design Document

4. TITLE AND SUBTITLE

Software Design Document SAF Simulation Host CSCI (8)

5. FUNDING NUMBERS

Contract Numbers:
MDA972-89-C-0060
MDA972-89-C-0061

6. AUTHOR(S)

Author not specified.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Bolt Beranek and Newman, Inc. (BBN)
Systems and Technologies; Advanced Simulation
10 Moulton Street
Cambridge, MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER

Advanced Simulation #
9111

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency (DARPA
3701 North Fairfax Drive
Arlington, VA 22203-1714

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

DARPA Report Number:
None.

11. SUPPLEMENTARY NOTES

None

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Statement A: Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

Distribution Code:
A

13. ABSTRACT *(Maximum 200 words)*

A Simulation Network (SIMNET) project Software Design Document that describes the Semi-Automated Forces (SAF) Simulation Host Computer Software Configuration Item (CSCI number 8) of the SIMNET hardware and software training system for vehicle crew training and operational training.

14. SUBJECT TERMS

SIMNET Software Design Document for the SAF Simulation Host CSCI (CSCI 8).

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Same as report.

2.7.1 remote.c CSC

/simnet/src/host/remote.c

This CSC contains all the code for handling remote vehicles, including some of the stealth code. In addition to the CSUs, there is a constant definition, as shown in the following table.

Constant	Value	Units
MAX_TIME_OFF_ROAD	10000	msecs

Table 2.7-1: remote.c Constant Definition

2.7.1.1 create_remote CSU

This CSU creates a remote vehicle by allocating and initializing a REMOTE_VARS structure and allocating a SAF object structure. It also initializes the SAF object structure, creating the tick, entity, and vehicle information for that structure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
id	unsigned int	Standard
stealthp	unsigned short	Standard
force	ForceID	basic.h
ReturnValues		
Return Value	**Type**	**Meaning**
safobj	pointer to SAF_OBJECT	Pointer to remote vehicle
Calls		
Function	**Where Described**	
allocate_remote	Sec. 2.9.1.2 Appendix A	
allocate_safobj	Sec. 2.2.2.2	
create_tickable	Sec. 2.2.2.2.1	
remote_tick	Sec. 2.7.1.13	
create_entity	Sec. 2.9.2.1.1	
create_vehicle	Sec. 2.9.2.2.1	
init_grid_entry_list	Sec. 2.9.3.1.11	

Table 2.7-2: create_remote CSU [8.7.1.1]

2.7.1.2 remote_init_vars CSU

This CSU initializes a SAF remote vehicle's state, including REMOTE_VARS, ENTITY_VARS, and VEHICLE_VARS structures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	SAF_OBJECT	Sec. 2.9.1.1
stealthp	unsigned short	Standard
force	ForceID	basic.h
Calls		
Function	**Where Described**	
init_grid_entry_list	Sec. 2.9.3.1.11	

Table 2.7-3: remote_init_vars CSU [8.7.1.2]

2.7.1.3 remote_go_away CSU

This CSU causes a remote SAF vehicle to completely disappear by destroying all of its structures and any sign of its appearance through a series of calls.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
cancel_fncl_group	Sec. 2.2.1.1.5	
destroy_tickable	Sec. 2.2.2.2.2	
destroy_entity	Sec. 2.9.2.1.2	
destroy_grid_entry_list	Sec. 2.9.3.1.13	
destroy_vehicle	Sec. 2.9.2.2.2	
buffer_deallocate	Sec. 2.14.4.2.14	
deallocate_remote	Sec. 2.9.1.2	
deallocate_safobj	Sec. 2.2.2.1.2	
unhash_saf_id		

Table 2.7-4: remote_go_away CSU [8.7.1.3]

2.7.1.4 remote_show CSU

This CSU determines the state of the remote vehicle and prints that state to the simulation host parser.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
flags	int	Standard

Calls	
Function	**Where Described**
entity_show	Sec. 2.9.2.1.4
vehicle_show	Sec. 2.9.2.2.4
print_routepoint	Sec. 2.10.2.4.16

Table 2.7-5: remote_show CSU [8.7.1.4]

2.7.1.5 remote_start_ticking CSU

This CSU starts the ticking of a remote SAF object. It sets the last update time to zero and calls start_ticking.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
start_ticking	Sec. 2.2.2.2.3	

Table 2.7-6: remote_start_ticking CSU [8.7.1.5]

2.7.1.6 remote_fill_in_appearance_data CSU

This CSU fills out a packet to send to the symbolics telling it about the state of a remote vehicle. It enters entity and vehicle appearance data for a SAF object through calls to entity_fill_in_appearance_data and vehicle_fill_in_appearance_data. It then determines and enters the vehicle's status and sets the vehicle's tactics to zero.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
desc_ptr	pointer to VEHICLE_APPEARANCE_DESCRIPTOR	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
entity_fill_in_appearance_data	Sec. 2.9.2.1.6	
vehicle_fill_in_appearance_data	Sec. 2.9.2.2.6	
is_dead	Sec. 2.13.3.1	

Table 2.7-7: remote_fill_in_appearance_data CSU [8.7.1.6]

2.7.1.7 remote_fill_echelon_data CSU

This CSU fills in the VEHICLE_ECHELON_DECRIPTOR in the VAP packet to be sent to the symbolics structure elements job_desc, port_nember, and superior_id to zero and echelon to g_vehicle_echelon.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
desc_ptr	pointer to VEHICLE_ ECHELON_DESCRIPTOR	Sec. 2.4.1.1

Table 2.7-8: remote_fill_in_echelon_data CSU [8.7.1.7]

2.7.1.8 remote_fill_in_position_data CSU

This CSU fills in the remote SAF vehicle's position descriptor through calls to entity_fill_in_position_data and vehicle_fill_in_position_data.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
desc_ptr	pointer to VEHICLE_ POSITION_DESCRIPTOR	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
entity_fill_in_position_data	Sec. 2.9.2.1.6	
vehicle_fill_in_position_data	Sec. 2.9.2.2.5	

Table 2.7-9: remote_fill_in_position_data CSU [8.7.1.8]

2.7.1.9 remote_next_road_point CSU

This CSU returns the next road point toward which a remote SAF vehicle is heading, if the vehicle is on a road.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
NULL	pointer to ROUTEPOINT	Vehicle not on road
remote->last_road_point	pointer to ROUTEPOINT	Last point on road of vehicle

Calls	
Function	**Where Described**
remote_start_being_watched	Sec. 2.7.1.10
OBJ_VELOCITY	Sec. 2.9.1.1
abs	Sec. 2.13.3.2
OBJ_DIRECTION	Sec. 2.9.1.1
next_road_point	Sec. 2.12.1.15.5
heap_allocate	Sec. 2.14.2.1.1
vec_sub	Sec. 2.6.2.65.1 Vehicles CSCI SDD
vec2_dot	Sec. 2.14.3.5.24

Table 2.7-10: remote_next_road_point CSU [8.7.1.9]

2.7.1.10 remote_start_being_watched CSU

This CSU starts watching a SAF remote vehicle by setting the tick rate, setting the REMOTE_VARS being_watched element to TRUE, and setting its last_watched_at time to the current time.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
change_tick_rate	Sec. 2.2.2.2.5	

Table 2.7-11: remote_start_being_watched CSU [8.7.11]

2.7.1.11 remote_stop_being_watched CSU

This CSU stops watching a SAF remote vehicle by setting the tick rate to the normal tick rate (in the REMOTE_VARS structure) and setting the REMOTE_VARS being_watched element to FALSE.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
change_tick_rate	Sec. 2.2.2.2.5	

Table 2.7-12: remote_stop_being_watched CSU [8.7.1.11]

2.7.1.12 remote_new_appearance CSU

This CSU receives an appearance packet from a remote SAF vehicle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
spdu	pointer to SimulationPDU	p_sim.h
Calls		
Function	**Where Described**	
buffer_deallocate	Sec. 2.14.4.2.13	
buffer_up_refcnt	Sec. 2.14.4.2.16	

Table 2.7-13: remote_new_appearance CSU [8.7.1.12]

2.7.1.13 remote_tick CSU

This CSU updates a SAF remote vehicle, based on information in an appearance packet. If the remote vehicle is a stealth, its position is sent to the workstation if that workstation is controlling it.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	Remote has unexpected packet in receive queue	
ERROR_OUT	Appearance of unknown vehicle class	
Calls		
Function	**Where Described**	
tickable_note_start_tick	Sec. 2.2.2.2.6	
remote_stop_being_watched	Sec. 2.7.1.11	
buffer_simple_dequeue	Sec. 2.14.4.2.19	
remote_deactivate	Sec. 2.7.1.15	
buffer_simple_flush	Sec. 2.14.4.1.16	
ERROR_OUT	Sec. 2.5.2.2	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
buffer_deallocate	Sec. 2.14.4.2.14	
vec_copy	Sec. 2.6.2.59.1 Vehicles CSCI SDD	
icon_from_object_type	Sec. 2.4.3.1.9	
get_soil_type	Sec. 2.14..1.2.1	
update_grid_entry_list	Sec. 2.9.3.1.12	
fmat_to_rmat	Sec. 2.14.3.5.16	
copy_matrix_row_to_vector	Sec. 2.14.3.5.18	
change_tick_rate	Sec. 2.2.2.2.5	
broadcast_pae_data	Sec. 2.4.3.2.74	
broadcast_appearance_data	Sec. 2.4.3.2.73	
buffer_allocate	Sec. 2.14.4.2.12	
fill_sbx_opfor_header	Sec. 2.4.3.2.18	
s_atan2	Sec. 2.14.3.9 See Appendix A	
sbx_connection_send_to_port	Sec. 2. 4.3.2.19	
buffer_deallocate	Sec. 2.14.4.2.14	

Table 2.7-14: remote_tick CSU [8.7.1.13]

2.7.1.14 send_stealth_gone_msg CSU

This CSU informs the workstation controlling a stealth that the stealth has left the network.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
port	int	Standard

Calls	
Function	**Where Described**
buffer_allocate	Sec. 2.14.4.2.12
fill_sbx_opfor_header	Sec. 2.4.3.2.18
sbx_connection_send_to_port	Sec. 2.4.3.2.19
buffer_deallocate	Sec. 2.14.4.2.15

Table 2.7-15: send_stealth_gone_msg CSU [8.7.1.14]

2.7.1.15 remote_deactivate CSU

This CSU sets the appropriate parameters of a remote vehicle when that vehicle becomes deactivated.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
DEBUG_REMOTE	Sec. 2.5.2.2	
send_stealth_gone_msg	Sec. 2.7.1.14	
sbx_printf	Sec. 2.4.3.2.8	
RETYPE_VEHICLE	Sec. 2.9.3.2	
destroy_grid_entry_list	Sec. 2.9.3.1.13	
FOR_VEHICLES_DO	Sec. 2.9.3.2	
remove_vehicles	Sec. 2.14.1.1.5	
stop_ticking	Sec. 2.2.3.4	
broadcast_vehicle_is_gone	Sec. 2.4.3.2.75	
buffer_deallocate	Sec. 2.14.4.2.15	

Table 2.7-16: remote_deactivate CSU [8.7.1.15]

2.7.1.16 remote_change_stealth_controlling_port CSU

This CSU checks to see if the remote SAF stealth vehicle is currently connected to a controlling port and if the port is different from the new one assigned. If so, the CSU sends a message to the old remote port that the stealth vehicle has left that workstation and assigns the vehicle to its new controlling port.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
send_stealth_gone_message	Sec. 2.7.1.14	
sbx_printf	Sec. 2.4.3.2.8	

Table 2.7-17: remote_change_stealth_controlling_port CSU [8.7.1.16]

2.7.1.17 create_remote_vehicle CSU

This CSU checks if an inactive remote SAF vehicle exists. If it does, the CSU initializes it. If it does not, the CSU creates it. In either case, the CSU then starts it ticking and returns a pointer to it.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
spdu	pointer to SimulationPDU	p_sim.h
id	unsigned int	Standard
stealthp	unsigned int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
safobj	pointer to SAF_OBJECT	Points to remote SAF vehicle
Calls		
Function	**Where Described**	
LOOKUP_VEHICLE	Sec. 2.9.3.2	
DEBUG_REMOTE	Sec. 2.5.2.2	
RETYPE_VEHICLE	Sec. 2.9.3.1	
remote_init_vars	Sec. 2.7.1.2	
create_remote	Sec. 2.7.1.1	
remote_start_ticking	Sec. 2.7.1.5	

Table 2.7-18: create_remote_vehicle CSU [8.7.1.17]

2.8 UNITS CSC

The Units CSC simulates the organizational structure of the simulated forces (the local vehicles). The vehicles are organized into a hierarchy containing entities such as platoons, companies, and flights. The units code is called by the SAF command interface to distribute unit commands down to the subunits and the vehicles. When called by the scheduler, it updates unit positions based on vehicle positions, processes packets delivered to it, generates unit reports, determines the status of units, and interprets orders.

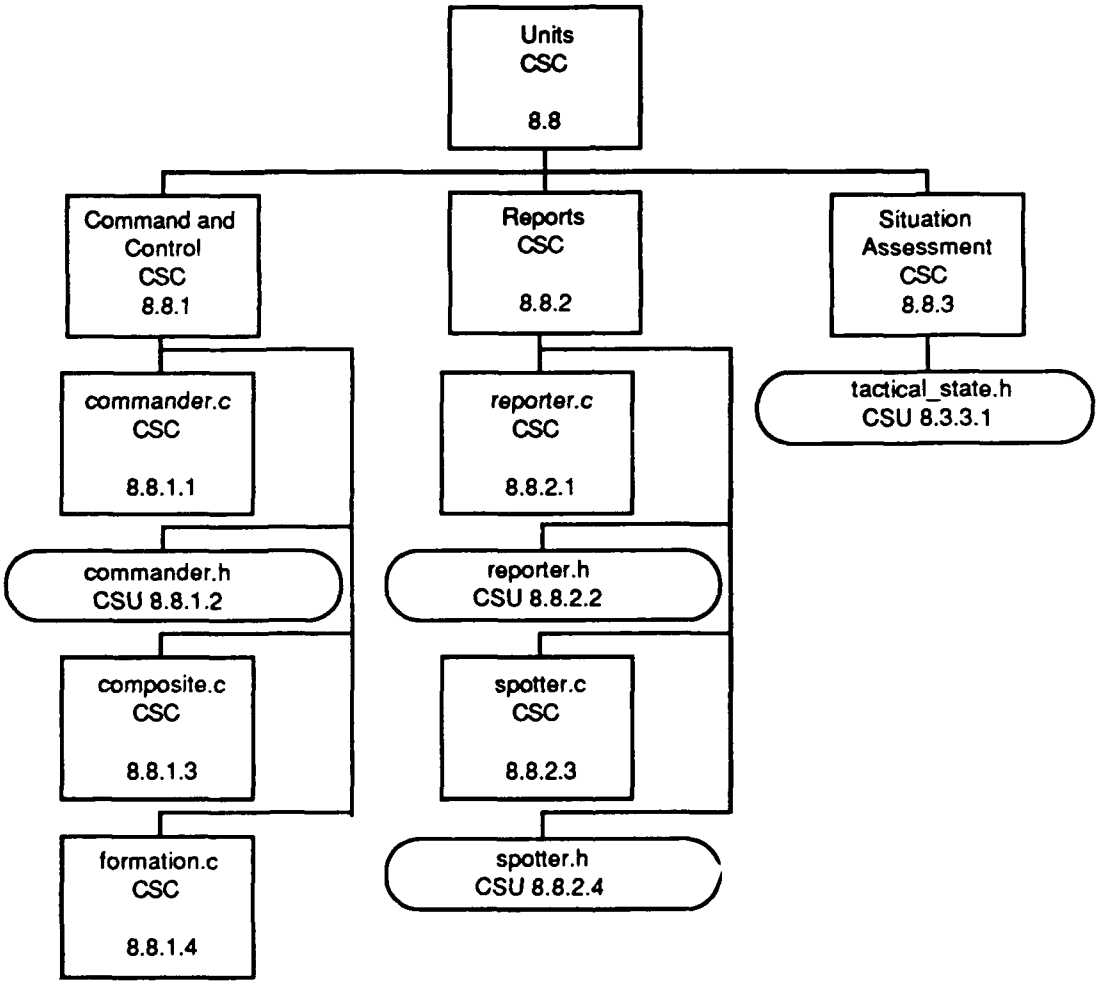


Figure 2.8-1: Units CSC Structure

2.8.1 Command and Control CSC

The Command and Control CSC [8.8.1] contains three CSCs and a CSU which handle the following responsibilities: commander.c [8.8.1.1] provides instructions for composite vehicles; commander.h [8.8.1.2] is the header CSU for commander.c; composite.c [8.8.1.3] organizes groups of vehicles and other composites; and formation.c [8.8.1.4] places individual vehicles into their proper formation positions.

2.8.1.1 commander.c CSC

/simnet/src/host/commander.c

The code in this CSC is the source of composite vehicle instructions. In any composite, there is one vehicle (the "leading_vehicle") which is in control of what all of the other vehicles are doing (unless sub units have been told to perform other tasks). This vehicle is the "brain" of a composite. When a composite is told what to do through immediate interventions or combat instruction sets, the information is retained in this vehicle's driver or pilot state, and in the composite's commander variable, and the other vehicles obtain their information from there. The code for switching the leading vehicle of a composite also resides in this CSC.

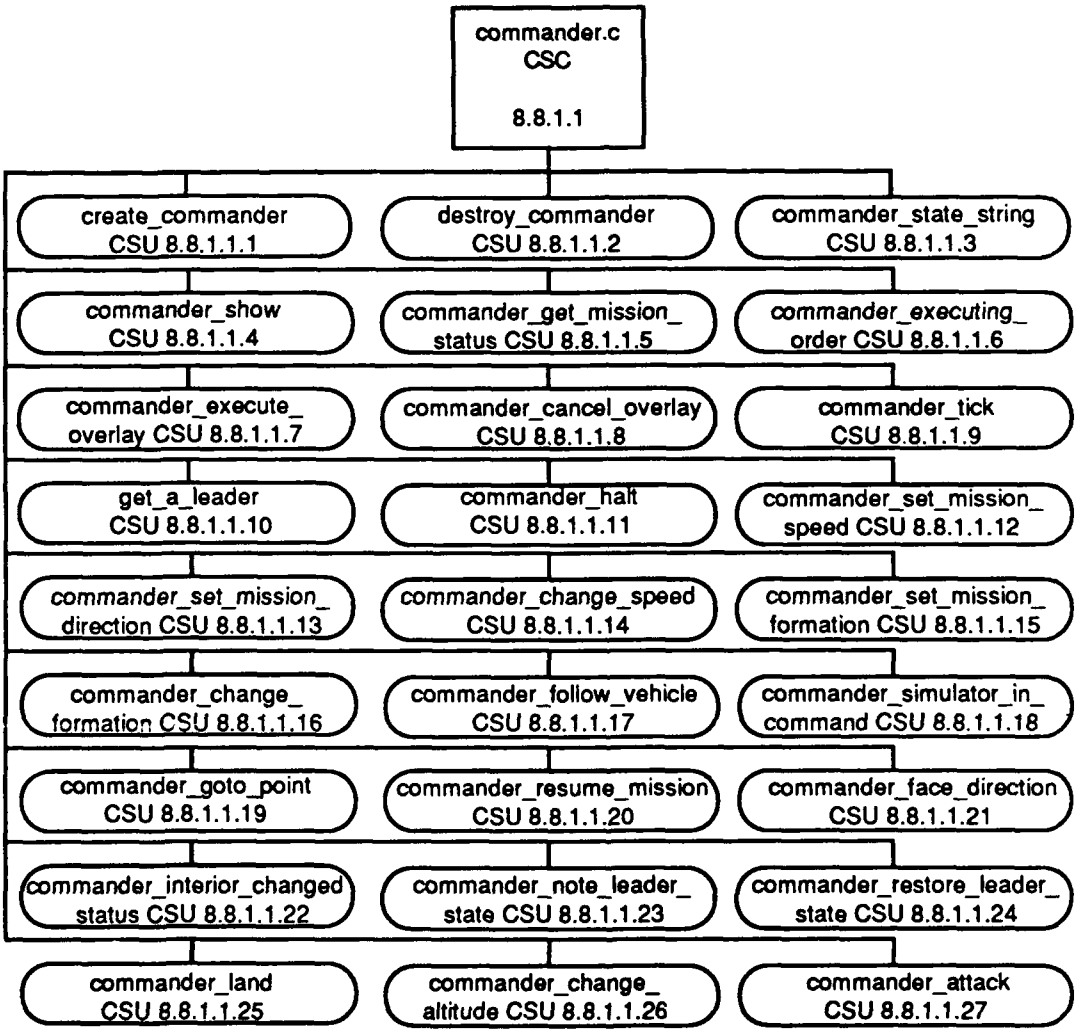


Figure 2.8-2: commander.c CSC Structure

2.8.1.1.1 create_commander CSU

This CSU allocates memory to build a COMMANDER_VARS structure, initializes its elements, and returns a pointer to the commander.

ReturnValues		
Return Value	**Type**	**Meaning**
commander	pointer to COMMANDER_VARS	Sec. 8.9.1.2 veh_storage.h
Calls		
Function	**Where Described**	
allocate_commander	Sec. 2.9.1.2 See Appendix A	

Table 2.8-1: create_commander CSU [8.8.1.1.1]

2.8.1.1.2 destroy_commander CSU

This CSU calls deallocate_commander to release the commander's memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
commander	pointer to COMMANDER_VARS	Sec. 8.9.1.2 veh_storage.h
Calls		
Function	**Where Described**	
deallocate_commander	Sec. 2.9.1.2 See Appendix A	

Table 2.8-2: destroy_commander CSU [8.8.1.1.2]

2.8.1.1.3 commander_state_string CSU

This CSU converts a commander state token into a character string, returning a pointer to the string.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
state	unsigned short	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
"idle"	Pointer to char	Commander state
"executing mission"	Pointer to char	Commander state
"immediate intervention"	Pointer to char	Commander state
"unknown"	Pointer to char	Unknown token

Table 2.8-3: commander_state_string CSU [8.8.1.1.3]

2.8.1.1.4 commander_show CSU

This CSU displays the commander's state.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
commander	pointer to COMMANDER_VARS	Sec. 8.9.1.2 veh_storage.h
Calls		
Function	**Where Described**	
commander_state_string	Sec. 2.8.1.1.3	

Table 2.8-4: commander_show CSU [8.8.1.1.4]

2.8.1.1.5 commander_get_mission_status CSU

This CSU creates a character string buffer for displaying that contains the commander's state and the leading and useable subordinate vehicles in the commander's composite.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
buf	pointer to char	Created buffer
Calls		
Function	**Where Described**	
saf_vehicle_useless	Sec. 2.6.1.1.41	

Table 2.8-5: commander_get_mission_status CSU [8.8.1.1.5]

2.8.1.1.6 commander_executing_order CSU

This CSU determines if the commander's state is not idle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
command	pointer to COMMANDER_VARS	Sec. 8.9.1.2 veh_storage.h
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Commander state not idle.
FALSE	int	Commander state idle.

Table 2.8-6: commander_executing_order CSU [8.8.1.1.6]

2.8.1.1.7 commander_execute_overlay CSU

This CSU sets state variables in the commander to indicate that an overlay is being executed. The leading vehicle is then assigned a route to follow, and the other available vehicles are instructed to follow the leading vehicle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm	pointer to CONTROL_MEASURE	2.10.2.2
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	Cannot execute anything but route cm.	
Calls		
Function	**Where Described**	
ERROR_OUT	Sec. 2.5.2.2 See Appendix A	
get_a_leader	Sec. 2.8.1.1.10	
composite_note_leader_state	Sec. 2.8.1.3.36	
saf_vehicle_useless	Sec. 2.6.1.1.41	
saf_vehicle_reset_station_keeper	Sec. 2.6.1.1.51	
saf_vehicle_set_route	Sec. 2.6.1.1.48	
compcsite_inferior_changed_status	Sec. 2.8.1.3.37	

Table 2.8-7: commander_execute_overlay CSU [8.8.1.1.7]

2.8.1.1.8 commander_cancel_overlay CSU

The CSU sets the commander state variables to indicate that the overlay should no longer be executed. It then calls a similar function on the leading vehicle, instructing it to stop moving.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
saf_vehicle_cancel_overlay	Sec. 2.6.1.1.47	
composite_note_leader_state	Sec. 2.8.1.3.36	
composite_inferior_changed_status	Sec. 2.8.1.3.37	

Table 2.8-8: commander_cancel_overlay CSU [8.8.1.1.8]

2.8.1.1.9 commander_tick CSU

Upon a tick, this CSU checks if the commander's state is executing a mission. The return is TRUE if a mission is being executed, and FALSE otherwise. If a mission is being executed, the CSU calls check_predicates, which checks if any behaviors have been triggered.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Commander executing mission.
FALSE	int	Commander not executing mission.
Calls		
Function	**Where Described**	
check_predicates	Sec. 2.10.1.3.13	

Table 2.8-9: commander_tick CSU [8.8.1.1.9]

2.8.1.1.10 get_a_leader CSU

This CSU checks if there is a leader, and if no leader exists, it tries to assign one.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	There is a leader.
FALSE	int	No leader.
Calls		
Function	**Where Described**	
composite_note_leader_state	Sec. 2.8.1.3.36	
composite_reassign_current_formation	Sec. 2.8.1.3.27	
DEBUG_COMPOSITE	Sec. 2.5.2.2 See Appendix A	

Table 2.8-10: get_a_leader CSU [8.8.1.1.10]

2.8.1.1.11 commander_halt CSU

This CSU triggers a halt TAC/E which causes the leading vehicle to temporarily suspend movement.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
type	int	Standard
Calls		
Function	**Where Described**	
get_a_leader	Sec. 2.8.1.1.10	
saf_vehicle_stop_mimicing	Sec. 2.6.1.1.19	
composite_note_leader_state	Sec. 2.8.1.3.36	
saf_vehicle_useless	Sec. 2.6.1.1.41	
saf_vehicle_reset_station_keeper	Sec. 2.6.1.1.51	
pilot_hold_im	Sec. 2.6.4.2.76	
saf_vehicle_halt	Sec. 2.6.1.1.52	
composite_inferior_changed_status	Sec. 2.8.1.3.37	

Table 2.8-11: commander_halt CSU [8.8.1.1.11]

2.8.1.1.12 commander_set_mission_speed CSU

This CSU sets the speed of the leading SAF vehicle if it is able to find one.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
speed	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
FALSE	int	No leader
saf_vehicle_set_speed(...)	int	Set the speed.
Calls		
Function	**Where Described**	
get_a_leader	Sec. 2.8.1.1.10	
saf_vehicle_set_speed	Sec. 2.6.1.1.49	

Table 2.8-12: commander_set_mission_speed CSU [8.8.1.1.12]

2.8.1.1.13 commander_set_mission_direction CSU

This CSU sets the direction of the leading SAF vehicle if it is able to find one.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
direction	int	Standard
Calls		
Function	**Where Described**	
get_a_leader	Sec. 2.8.1.1.10	
saf_vehicle_set_direction	Sec. 2.6.1.1.50	

Table 2.8-13: commander_set_mission_direction CSU [8.8.1.1.13]

2.8.1.1.14 commander_change_speed CSU

This CSU changes the speed of the leading SAF vehicle if it is able to find one, along with all useable member vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
speed	REAL	sim_types.h
Calls		
Function	**Where Described**	
get_a_leader	Sec. 2.8.1.1.10	
saf_vehicle_change_speed	Sec. 2.6.1.1.53	
saf_vehicle_useless	Sec. 2.6.1.1.41	

Table 2.8-14: commander_change_speed CSU [8.8.1.1.14]

2.8.1.1.15 commander_set_mission_formation CSU

This CSU changes the formation which is to be used while executing an overlay. If the mission formation is not currently overridden, the formation of the unit is changed to the new value. The return value indicates whether such a change was made.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
formation	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Formation changed.
FALSE	int	Formation not changed.

Calls	
Function	**Where Described**
composite_note_leader_state	Sec. 2.8.1.3.36
composite_assume_formation	Sec. 2.8.1.3.26
composite_inferior_changed_status	Sec. 2.8.1.3.37

Table 2.8-15: commander_set_mission_formation CSU [8.8.1.1.15]

2.8.1.1.16 commander_change_formation CSU

This CSU overrides the current formation of the unit, imposing the passed value.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
formation	pointer to char	Standard

Calls	
Function	**Where Described**
composite_note_leader_state	Sec. 2.8.1.3.36
composite_assume_formation	Sec. 2.8.1.3.26
composite_inferior_changed_status	Sec. 2.8.1.3.37

Table 2.8-16: commander_change_formation CSU [8.8.1.1.16]

2.8.1.1.17 commander_follow_vehicle CSU

This CSU causes the leading vehicle to start following the indicated vehicle. All available members of the unit will continue to follow the leading vehicle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
vid	unsigned int	Standard
xoff	int	Standard
yoff	int	Standard

Calls	
Function	**Where Described**
saf_vehicle_stop_mimicing	Sec. 2.6.1.1.19
composite_note_leader_state	Sec. 2.8.1.3.36
saf_vehicle_useless	Sec. 2.6.1.1.41
saf_vehicle_reset_station_keeper	Sec. 2.6.1.1.51
saf_vehicle_follow_vehicle	Sec. 2.6.1.1.54
composite_inferior_changed_status	Sec. 2.8.1.3.37

Table 2.8-17: commander_follow_vehicle CSU [8.8.1.1.17]

2.8.1.1.18 commander_simulator_in_command CSU

This CSU causes the leading vehicle to mimic the passed vehicle (disappear from the Simnet and do exactly what the simulator does).

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
vid	unsigned int	Standard
Calls		
Function	**Where Described**	
get_a_leader	Sec. 2.8.1.1.10	
composite_note_leader_state	Sec. 2.8.1.3.36	
saf_vehicle_useless	Sec. 2.6.1.1.41	
saf_vehicle_reset_station_keeper	Sec. 2.6.1.1.51	
saf_vehicle_mimic_vehicle	Sec. 2.6.1.1.18	
composite_inferior_changed_status	Sec. 2.8.1.3.37	

Table 2.8-18: commander_simulator_in_command CSU [8.8.1.1.18]

2.8.1.1.19 commander_goto_point CSU

This CSU sends the leading vehicle to a point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
x	REAL	sim_types.h
y	REAL	sim_types.h
backwardp	int	Standard
Calls		
Function	**Where Described**	
get_a_leader	Sec. 2.8.1.1.10	
saf_vehicle_stop_mimicing	Sec. 2.6.1.1.19	
composite_note_leader_state	Sec. 2.8.1.3.36	
saf_vehicle_useless	Sec. 2.6.1.1.41	
saf_vehicle_reset_station_keeper	Sec. 2.6.1.1.51	
saf_vehicle_goto_point	Sec. 2.6.1.1.55	
composite_inferior_changed_status	Sec. 2.8.1.3.37	

Table 2.8-19: commander_goto_point CSU [8.8.1.1.19]

2.8.1.1.20 commander_resume_mission CSU

This CSU causes the termination of any outstanding TAC/Es and override formation, and restores the commander state to its value before the TAC/E started (possibly relinquishing control of its member vehicles back to its superior unit).

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1

Calls	
Function	**Where Described**
saf_vehicle_stop_mimicing	Sec. 2.6.1.1.19
saf_vehicle_useless	Sec. 2.6.1.1.41
saf_vehicle_resume_mission	Sec. 2.6.1.1.56
composite_note_leader_state	Sec. 2.8.1.3.36
composite_inferior_changed_status	Sec. 2.8.1.3.37
composite_assume_formation	Sec. 2.8.1.3.26
pop_resumable_cis	Sec. 2.10.1.1.13

Table 2.8-20: commander_resume_mission CSU [8.8.1.1.20]

2.8.1.1.21 commander_face_direction CSU

This CSU causes the leading vehicle to face a direction, and in the case of air vehicles, it causes other vehicles to do so as well.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
mathradians	REAL	sim_types.h

Calls	
Function	**Where Described**
get_a_leader	Sec. 2.8.1.1.10
saf_vehicle_stop_mimicing	Sec. 2.6.1.1.19
composite_note_leader_state	Sec. 2.8.1.3.36
saf_vehicle_useless	Sec. 2.6.1.1.41
saf_vehicle_reset_station_keeper	Sec. 2.6.1.1.51
saf_vehicle_face_direction	Sec. 2.6.1.1.57
composite_inferior_changed_status	Sec. 2.8.1.3.37

Table 2.8-21: commander_face_direction CSU [8.8.1.1.21]

2.8.1.1.22 commander_inferior_changed_status CSU

This CSU is called when a member of a composite becomes available or unavailable. The CSU reassigns the current formation, adjusting all of the member vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
composite_inferior_changed_status	Sec. 2.8.1.3.37	
composite_reassign_current_formation	Sec. 2.8.1.3.27	

Table 2.8-22: commander_inferior_changed_status CSU [8.8.1.1.22]

2.8.1.1.23 commander_note_leader_state CSU

This CSU makes a copy of the current leader state so it can be restored to a new leader, if necessary. If this unit is inactive, this message is passed up the hierachy.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
composite_note_leader_state	Sec. 2.8.1.3.36	

Table 2.8-23: commander_note_leader_state CSU [8.8.1.1.23]

2.8.1.1.24 commander_restore_leader_state CSU

This CSU restores the currently saved leader state to a new leading vehicle. If this unit is inactive, this message is passed up the hierarchy.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
saf_vehicle_stop_mimicing	Sec. 2.6.1.1.19	
saf_vehicle_mimic_vehicle	Sec. 2.6.1.1.18	

Table 2.8-24: commander_restore_leader_state CSU [8.8.1.1.24]

2.8.1.1.25 commander_land CSU

This CSU causes the leading air vehicle to land. Other available member vehicles will do as the leading vehicle does.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
x	REAL	sim_types.h
y	REAL	sim_types.h
Calls		
Function	**Where Described**	
get_a_leader	Sec. 2.8.1.1.10	
composite_note_leader_state	Sec. 2.8.1.3.36	
pilot_land_im	Sec. 2.6.4.2.70	
saf_vehicle_useless	Sec. 2.6.1.1.41	
saf_vehicle_reset_station_keeper	Sec. 2.6.1.1.51	
composite_inferior_changed_status	Sec. 2.8.1.3.37	

Table 2.8-25: commander_land CSU [8.8.1.1.25]

2.8.1.1.26 commander_change_altitude CSU

This CSU changes the altitude of all available member vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
altitude	REAL	sim_types.h
Calls		
Function	**Where Described**	
saf_vehicle_useless	Sec. 2.6.1.1.41	
pilot_change_altitude_im	Sec. 2.6.4.2.66	

Table 2.8-26: commander_change_altitude CSU [8.8.1.1.26]

2.8.1.1.27 commander_attack CSU

This CSU computes the necessary vectors to perform the specific attack profile, and then causes the available units to execute this profile. Each vehicle is given its own separate route to attack based on its own relative position in the formation.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
popup_x	REAL	sim_types.h
popup_y	REAL	sim_types.h
target_x	REAL	sim_types.h
target_y	REAL	sim_types.h

Calls	
Function	**Where Described**
get_a_leader	Sec. 2.8.1.1.10
vec2_set	Sec. 2.14.3.5.21
vec2_sub	Sec. 2.14.3.5.21
vec2_mag2	Sec. 2.14.3.5.27
OBJ_POSITION	Sec. 2.9.1.1 See Appendix A
vec2_copy	Sec. 2.14.3.5.29
OBJ_DIRECTION	Sec. 2.9.1.1 See Appendix A
vec2_norm	Sec. 2.14.3.5.31
composite_note_leader_state	Sec. 2.8.1.3.36
saf_vehicle_useless	Sec. 2.6.1.1.41
pilot_get_asm	Sec. 2.6.4.2.84
p_follower_get_wrld_offset	Sec. 2.6.4.3.7
pilot_hoverattack_im	Sec. 2.6.4.2.75
composite_inferior_changed_status	Sec. 2.8.1.3.37

Table 2.8-27: commander_attack CSU [8.8.1.1.27]

2.8.1.2 commander.h CSU

/simnet/src/host/commander.h

This CSU contains the state symbolic constants for the commander.

Constant	Value
COMMANDER_STATE_IDLE	1
COMMANDER_STATE_EXECUTING_MISSION	2
COMMANDER_STATE_IMMEDIATE_INTERVENTION	3

Table 2.8-28: commander.h Constant Definitions

2.8.1.3 composite.c CSC

/simnet/src/host/composite.c

Composites are used to organize groups of vehicles or other composites. They allow a single command or request for a change to be given to entire units or groups of vehicles, simplifying the task of the person running the vehicles from the workstation. Composites

"move" around the database, taking their sub composites and vehicles with them. (The composite location is the location of the lead vehicle of that composite, or the lead vehicle of the lead composite, etc).

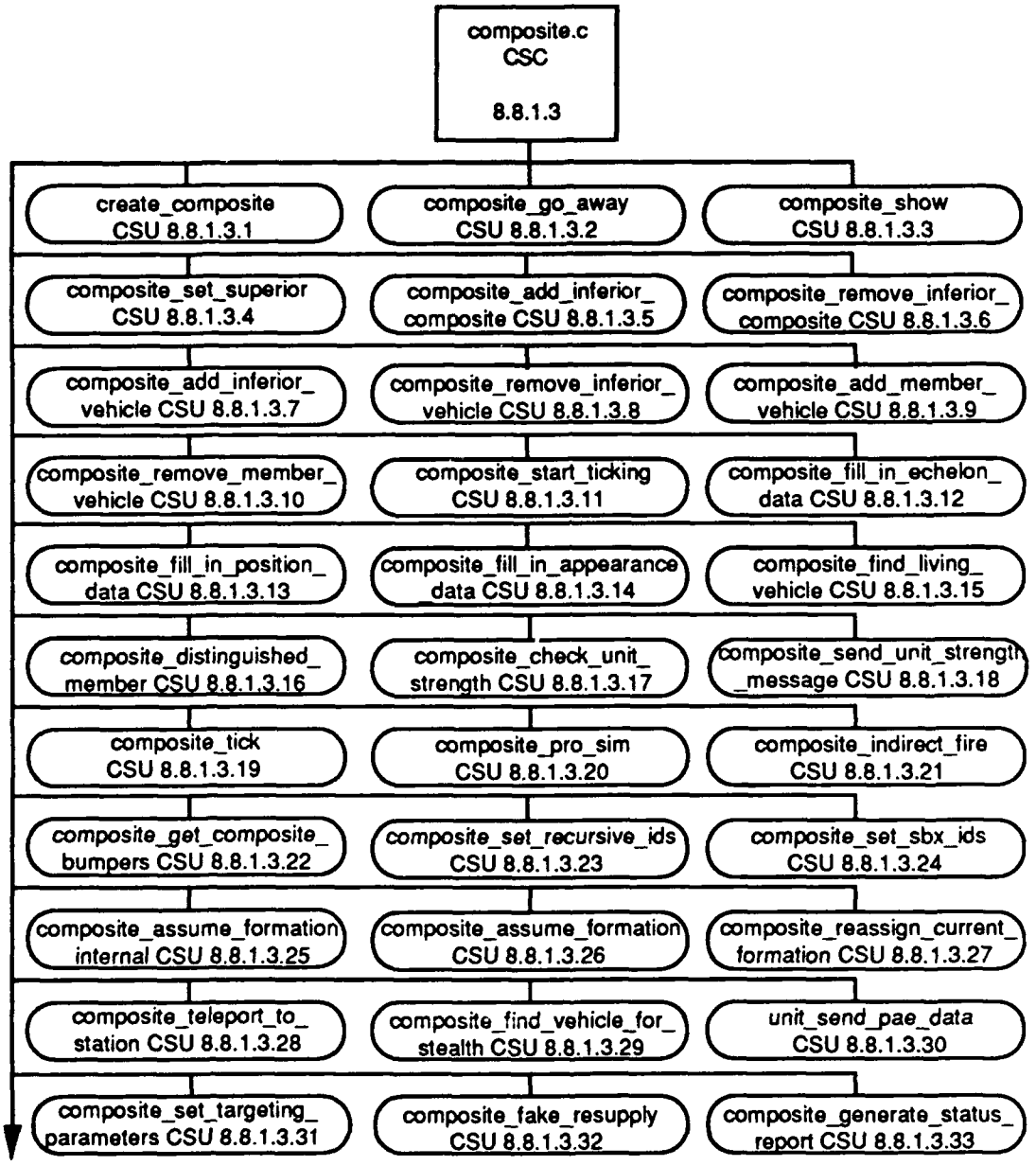


Figure 2.8-3: composite.c CSC Structure Part 1 of 2

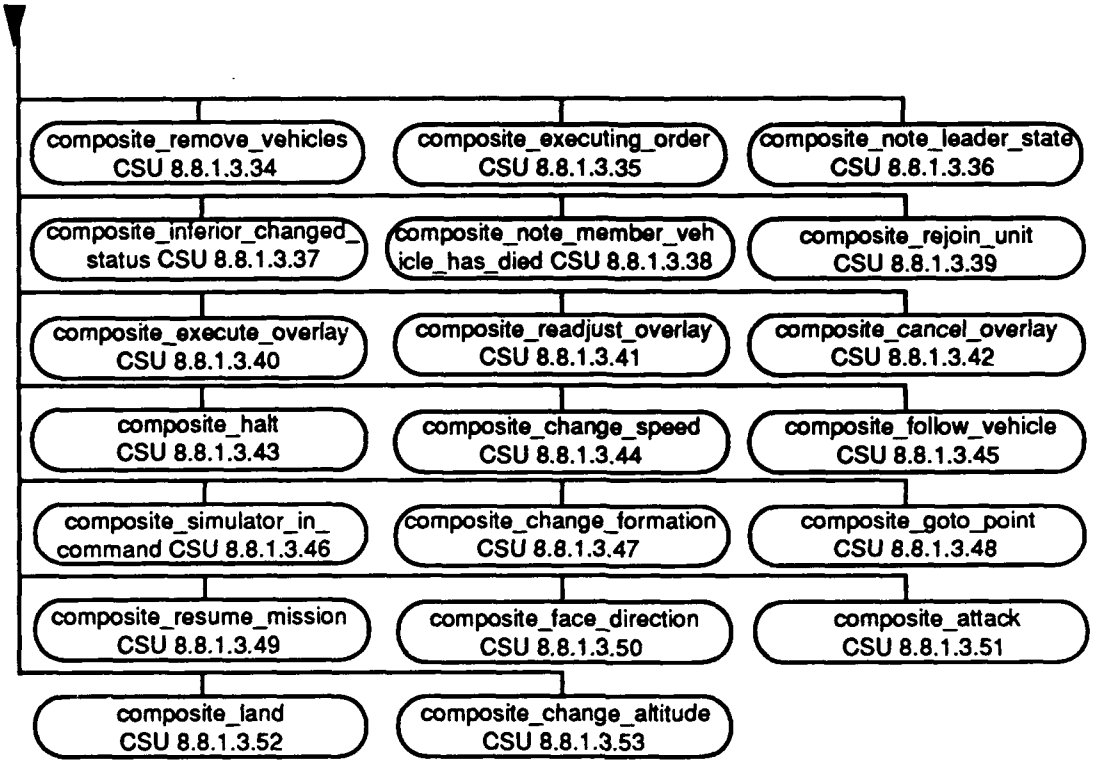


Figure 2.8-4: composite.c CSC Structure Part 2 of 2

This CSC contains all of the main composite CSUs. It creates and destroys composites, adds and subtracts inferiors and superiors in the vehicle/composite hierarchy, sets up the echelon configuration for the composite (see data files echelons), sets up positions relative to other vehicles in the composite and sub composites (see data files formations), initiates composite status reporting, and sets up the starting points for dealing with combat instruction sets and immediate interventions.

2.8.1.3.1 create_composite CSU

This CSU allocates memory to store the composite, and allocates all component objects. It also initializes formation and CIS databases, and other composite variables.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
forceID	ForceID	basic.h
tactics	unsigned char	Standard
echelon	pointer to char	Standard
echelon_type	pointer to char	Standard
heading	REAL	sim_types.h
position	pointer to REAL	sim_types.h
owner	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3

ReturnValues		
Return Value	**Type**	**Meaning**
safobj	pointer to SAF_OBJECT	Composite created.
Calls		
Function	**Where Described**	
composite_tick	Sec. 2.8.1.3.19	
generate_vehicle_id		
allocate_composite	Sec. 2.9.1.2 See Appendix A	
map_echelon_symbol_to_ echelon_number	Sec. 2.4.3.1.3	
find_tag	Sec. 2.1.1.4.3	
create_predicates	Sec. 2.10.1.3.1	
ft_untagged_table	Sec. 2.14.1.2.15	
allocate_safobj	Sec. 2.2.2.2	
create_tickable	Sec. 2.2.3.1	
create_entity	Sec. 2.9.2.1.1	
map_echelon_type_symbol_ to_icon_number	Sec. 2.4.3.1.7	
create_spotter	Sec. 2.8.2.3.1	
create_reporter	Sec. 2.8.2.1.1	
create_commander	Sec. 2.8.1.1.1	
create_navigator	Sec. 2.10.2.3.1	

Table 2.8-29: create_composite CSU [8.8.1.3.1]

2.8.1.3.2 composite_go_away CSU

This CSU frees composite object memories and releases the use of its vehicle id.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
cancel_fncl_group	Sec. 2.2.1.1.5	
destroy_tickable	Sec. 2.2.3.2	
destroy_entity	Sec. 2.9.2.1.2	
destroy_spotter	Sec. 2.8.2.3.2	
destroy_reporter	Sec. 2.8.2.1.2	
destroy_cis	Sec. 2.10.1.1.9	
destroy_predicates	Sec. 2.10.1.3.2	
deallocate_composite	Sec. 2.9.1.2 See Appendix A	
deallocate_safobj	Sec. 2.2.2.3	
unhash_saf_id		

Table 2.8-30: composite_go_away CSU [8.8.1.3.2]

2.8.1.3.3 composite_show CSU

This CSU displays the composite statistics.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
flags	int	Standard
Calls		
Function	**Where Described**	
entity_show	Sec. 2.9.2.1.4	
spotter_show	Sec. 2.8.2.3.12	
reporter_show	Sec. 2.8.2.1.4	
navigator_show	Sec. 2.10.2.3.3	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
print_vehicle_marking	Sec. 2.9.2.2.3	
commander_show	Sec. 2.8.1.1.4	
cis_show	Sec. 2.10.1.1.10	
predicates_show	Sec. 2.10.1.3.9	

Table 2.8-31: composite_show CSU [8.8.1.3.3]

2.8.1.3.4 composite_set_superior CSU

This CSU removes the composite from its current hierarchy and reassigns it to a new hierachy.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
superior	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
DEBUG_COMPOSITE	Sec. 2.5.2.2 See Appendix A	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
composite_remove_inferior_composite	Sec. 2.8.1.3.6	
composite_add_inferior_composite	Sec. 2.8.1.3.5	

Table 2.8-32: composite_set_superior CSU [8.8.1.3.4]

2.8.1.3.5 composite_add_inferior_composite CSU

This CSU adds a new composite to an existing hierarchy.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
inferior	pointer to SAF_OBJECT	Sec. 2.9.1.1

Table 2.8-33: composite_add_inferior_composite CSU [8.8.1.3.5]

2.8.1.3.6 composite_remove_inferior_composite CSU

This CSU removes a composite and its member vehicles from an operational hierarchy.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
inferior	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
DEBUG_COMPOSITE	Sec. 2.5.2.2 See Appendix A	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
composite_remove_member_vehicle	Sec. 2.8.1.3.10	

Table 2.8-34: composite_remove_inferior_composite CSU [8.8.1.3.6]

2.8.1.3.7 composite_add_inferior_vehicle CSU

This CSU adds a vehicle to a composite's inferior list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
vehicle	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
composite_add_member_vehicle	Sec. 2.8.1.3.9	

Table 2.8-35: composite_add_inferior_vehicle CSU [8.8.1.3.7]

2.8.1.3.8 composite_remove_inferior_vehicle CSU

This CSU removes a vehicle from a composite's inferior list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
vehicle	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
DEBUG_COMPOSITE	Sec. 2.5.2.2 See Appendix A	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
composite_remove_member_vehicle	Sec. 2.8.1.3.10	

Table 2.8-36: composite_remove_inferior_vehicle CSU [8.8.1.3.8]

2.8.1.3.9 composite_add_member_vehicle CSU

This CSU adds a vehicle to a composite's member list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
vehicle	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
composite_add_member_vehicle	Sec. 2.8.1.3.9	

Table 2.8-37: composite_add_member_vehicle CSU [8.8.1.3.9]

2.8.1.3.10 composite_remove_member_vehicle CSU

This CSU removes a vehicle from the composite's member list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
vehicle	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
DEBUG_COMPOSITE	Sec. 2.5.2.2 See Appendix A	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
composite_remove_member_vehicle	Sec. 2.8.1.3.10	

Table 2.8-38: composite_remove_member_vehicle CSU [8.8.1.3.10]

2.8.1.3.11 composite_start_ticking CSU

This CSU causes a composite to start ticking by making a call to start_ticking.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
start_ticking	Sec. 2.2.3.3	

Table 2.8-39: composite_start_ticking CSU [8.8.1.3.11]

2.8.1.3.12 composite_fill_in_echelon_data CSU

This CSU fills in an echelon data structure for transmission to the workstations.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
desc_ptr	pointer to VEHICLE_ ECHELON_DESCRIPTOR	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	

Table 2.8-40: composite_fill_in_echelon_data CSU [8.8.1.3.12]

2.8.1.3.13 composite_fill_in_position_data CSU

This CSU fills in a position data structure for transmission to the workstations.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
desc_ptr	pointer to VEHICLE_ POSITION_DESCRIPTOR	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
entity_fill_in_position_data	Sec. 2.9.2.1.5	

Table 2.8-41: composite_fill_in_position_data CSU [8.8.1.3.13]

2.8.1.3.14 composite_fill_in_appearance_data CSU

This CSU fills in an appearance data structure for transmission to the workstations.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
desc_ptr	pointer to VEHICLE_APPEARANCE_DESCRIPTOR	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
entity_fill_in_appearance_data	Sec. 2.9.2.1.6	

Table 2.8-42: composite_fill_in_appearance_data CSU [8.8.1.3.14]

2.8.1.3.15 composite_find_living_vehicle CSU

This CSU finds a living vehicle in a composite, if there are any.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
NULL	pointer to SAF_OBJECT	No composite vehicles.
composite->member_vehicles[i]	pointer to SAF_OBJECT	Living vehicle found.
composite->member_vehicles[0]	pointer to SAF_OBJECT	No living vehicle found.

Table 2.8-43: composite_find_living_vehicle CSU [8.8.1.3.15]

2.8.1.3.16 composite_distinguished_member CSU

This CSU finds the leading vehicle in a composite, and if one does not exist, it finds any living vehicle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
safobj->composite->leading_vehicle	pointer to SAF_OBJECT	Leading vehicle found.
composite_find_living_vehicle(...)	pointer to SAF_OBJECT	No leading vehicle, so look for a living vehicle.

Calls	
Function	**Where Described**
composite_find_living_ vehicle	Sec. 2.8.1.3.15

Table 2.8-44: composite_distinguished_member CSU [8.8.1.3.16]

2.8.1.3.17 composite_check_unit_strength CSU

This CSU determines a unit's percentile of living vehicles as a function of member vehicles and, if it is below 50% (either less than 50% or less than 25%), sends a message to the owner through a call to composite_send_unit_strength_message.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
composite_send_unit_ strength_message	Sec. 2.8.1.3.18	

Table 2.8-45: composite_check_unit_strength CSU [8.8.1.3.17]

2.8.1.3.18 composite_send_unit_strength_message CSU

This CSU sends a message from a top level unit to the owner that the unit strength is below a certain percentile.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
percent	int	Standard
Calls		
Function	**Where Described**	
sbx_printf	Sec. 2.4.3.2.8	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
OBJ_OWNER_PORT_ NUMBER	Sec. 2.9.1.1 See Appendix A	

Table 2.8-46: composite_send_unit_strength_message CSU [8.8.1.3.18]

2.8.1.3.19 composite_tick CSU

At the tick, this CSU checks if the previous leading vehicle is still the leading vehicle. It updates the previous leading vehicle to the present one if the previous one was not the same or was NULL. Note that the present leading vehicle may be NULL or may be making the previous leading vehicle NULL. If there are vehicles, the CSU does the commander tick. If there are living vehicles, it does the spotter and reporter ticks, as well as performs a check of unit strength. Next, if the commander is executing a mission, it does the navigator tick. Lastly, it checks for any packets in the receive queue, and digests them.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
composite_distinguished_ member	Sec. 2.8.1.3.16	
DEBUG_STATION	Sec. 2.5.2.2 See Appendix A	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
vec_copy	Sec. 2.6.2.59.1 Vehicle CSCI SDD	
OBJ_POSITION	Sec. 2.9.1.1 See Appendix A	
commander_tick	Sec. 2.8.1.1.9	
spotter_tick	Sec. 2.8.2.3.7	
reporter_tick	Sec. 2.8.2.3.7	
composite_check_unit_ strength	Sec. 2.8.1.3.17	
navigator_tick	Sec. 2.10.2.3.6	
buffer_simple_dequeue	Sec. 2.14.4.1.15	
OBJ_SIMPLE_RCVQ	Sec. 2.9.1.1 See Appendix A	
composite_pro_sim	Sec. 2.8.1.3.20	
buffer_deallocate	Sec. 2.14.4.2.15	

Table 2.8-47: composite_tick CSU [8.8.1.3.19]

2.8.1.3.20 composite_pro_sim CSU

This CSU examines the type of message received and, if it is an indirect fire type, calls composite_indirect_fire. If the message is of any other recognized type, it is ignored. If the message type is unrecognizable, it labels it as a bad packet type in an ERROR_OUT message.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
spdu	pointer to SimulationPDU	p_sim.h
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	Bad packet type.	

Calls	
Function	**Where Described**
composite_indirect_fire	Sec. 2.8.1.3.21
ERROR_OUT	Sec. 2.5.2.2 See Appendix A
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A

Table 2.8-48: composite_pro_sim CSU [8.8.1.3.20]

2.8.1.3.21 composite_indirect_fire CSU

This CSU computes the position for all impending bursts and calls cluster_shell for each one.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
spdu	pointer to SimulationPDU	p_sim.h
Calls		
Function	**Where Described**	
cluster_shell	Sec. 2.8.2.1.39	

Table 2.8-49: composite_indirect_fire CSU [8.8.1.3.21]

2.8.1.3.22 composite_get_composite_bumpers CSU

This CSU sets the composite marking field by taking all common elements of its inferiors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
composite_get_composite_bumpers	Sec. 2.8.1.3.22	

Table 2.8-50: composite_get_composite_bumpers CSU [8.8.1.3.22]

2.8.1.3.23 composite_set_recursive_ids CSU

This CSU assigns each member composite of a unit a number unique within that unit.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
top_id	unsigned char	Standard
rel_id	unsigned char	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
rel_id	unsigned char	Recursive id number.
Calls		
Function	**Where Described**	
composite_set_recursive_ids	Sec. 2.8.1.3.23	

Table 2.8-51: composite_set_recursive_ids CSU [8.8.1.3.23]

2.8.1.3.24 composite_set_sbx_ids CSU

This CSU assigns each member vehicle of a unit a number unique within that unit.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
sbx_uniq_id	unsigned char	Standard
Calls		
Function	**Where Described**	
composite_set_recursive_ids	Sec. 2.8.1.3.23	

Table 2.8-52: composite_set_sbx_ids CSU [8.8.1.3.24]

2.8.1.3.25 composite_assume_formation_internal CSU

This CSU obtains the desired vehicle positions for a formation, and assigns each vehicle its position.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
formation	pointer to char	Standard
Calls		
Function	**Where Described**	
get_station_info	Sec. 2.8.1.4.10	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
DEBUG_COMPOSITE	Sec. 2.5.2.2 See Appendix A	
saf_vehicle_set_leader	Sec. 2.6.1.1.5	
turret_set_scan_parms	Sec. 2.6.9.6.8	

Table 2.8-53: composite_assume_formation_internal CSU [8.8.1.3.25]

2.8.1.3.26 composite_assume_formation CSU

This CSU causes a composite to assume formation through calls to commander_note_leader_state, composite_assume_formation_internal, and commander_restore_leader_state.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
formation	pointer to char	Standard
Calls		
Function	**Where Described**	
commander_note_leader_state	Sec. 2.8.1.1.23	
composite_assume_formation_internal	Sec. 2.8.1.3.25	
commander_restore_leader_state	Sec. 2.8.1.1.24	

Table 2.8-54: composite_assume_formation CSU [8.8.1.3.26]

2.8.1.3.27 composite_reassign_current_formation CSU

This CSU reassigns the current composite formation through calls to composite_assume_formation_internal and commander_restore_leader_state.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
composite_assume_formation_internal	Sec. 2.8.1.3.25	
composite_restore_leader_state	Sec. 2.8.1.1.24	

Table 2.8-55: composite_reassign_current_formation CSU [8.8.1.3.27]

2.8.1.3.28 composite_teleport_to_station CSU

For each of the member vehicles of a composite, this CSU calls saf_vehicle_teleport_to_station.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1

Calls	
Function	**Where Described**
saf_vehicle_teleport_to_station	Sec. 2.6.1.1.6

Table 2.8-56: composite_teleport_to_station CSU [8.8.1.3.28]

2.8.1.3.29 composite_find_vehicle_for_stealth CSU

This CSU finds a vehicle for stealth attachment through a call to composite_find_living_vehicle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
composite_find_living_vehicle(...)	pointer to SAF_OBJECT	Living vehicle, if any.
Calls		
Function	**Where Described**	
composite_find_living_vehicle	Sec. 2.8.1.3.15	

Table 2.8-57: composite_find_vehicle_for stealth CSU [8.8.1.3.29]

2.8.1.3.30 unit_send_pae_data CSU

This CSU causes a PAE packet regarding all vehicles in a unit to go out.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
broadcast_pae_data	Sec. 2.4.3.2.74	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
unit_send_pae_data	Sec. 2.8.1.3.30	

Table 2.8-58: unit_send_pae_data CSU [8.8.1.3.30]

2.8.1.3.31 composite_set_targeting_parameters CSU

This CSU calls saf_vehicle_set_targeting_parameters for each of the composite member vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
firestatus	int	Standard
max_engagement_range	REAL	sim_types.h
marksmanship	REAL	sim_types.h
position_x	REAL	sim_types.h
position_y	REAL	sim_types.h
radius	REAL	sim_types.h
targets[ ]	unsigned short	Standard
Calls		
Function	**Where Described**	
saf_vehicle_set_targeting_ parameters	Sec. 2.6.1.1.12	

Table 2.8-59: composite_set_targeting_parameters CSU [8.8.1.3.31]

2.8.1.3.32 composite_fake_resupply CSU

This CSU calls saf_fake_resupply for each of the composite member vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
saf_vehicle_fake_resupply	Sec. 2.6.1.1.33	

Table 2.8-60: composite_fake_resupply CSU [8.8.1.3.32]

2.8.1.3.33 composite_generate_status_report CSU

This CSU generates a status report of the composite for the user at a workstation.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1

Calls	
Function	**Where Described**
commander_executing_order	Sec. 2.8.1.1.6

Table 2.8-63: composite_executing_order CSU [8.8.1.3.35]

2.8.1.3.36 composite_note_leader_state CSU

This CSU calls commander_note_leader_state.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
commander_note_leader_ state	Sec. 2.8.1.1.23	

Table 2.8-64: composite_note_leader_state CSU [8.8.1.3.36]

2.8.1.3.37 composite_inferior_changed_status CSU

This CSU calls commander_inferior_changed_status.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
commander_inferior_ changed_status	Sec. 2.8.1.1.22	

Table 2.8-65: composite_inferior_changed_status CSU [8.8.1.3.37]

2.8.1.3.38 composite_note_member_vehicle_has_died CSU

This CSU informs itself that a member vehicle has died, reducing the number of living vehicles, and then informs the vehicle's superior through a call to itself.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
dead_inferior	pointer to SAF_OBJECT	Sec. 2.9.1.1

Calls	
Function	**Where Described**
buffer_allocate	Sec. 2.14.4.2.12
clear_weapons_status	Sec. 2.6.9.8.11
is_dead	Sec. 2.13.3.1 See Appendix A
OBJ_APPEARANCE	Sec. 2.9.1.1 See Appendix A
OBJ_VEHICLECLASS	Sec. 2.9.1.1 See Appendix A
add_weapons_status	Sec. 2.6.9.8.12
OBJ_POSITION	Sec. 2.9.1.1 See Appendix A
tdb_giv_xy_get_utm	Sec. 2.21.7.24.3
commander_get_mission_status	Sec. 2.8.1.1.5
fill_sbx_opfor_header	Sec. 2.4.3.2.18
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A
sbx_connection_send_to_port	Sec. 2.4.3.2.19
buffer_deallocate	Sec. 2.14.4.2.15

Table 2.8-61: composite_generate_status_report CSU [8.8.1.3.33]

2.8.1.3.34 composite_remove_vehicles CSU

This CSU removes vehicles known to the spotter and/or reporter through calls to spotter_remove_vehicles and reporter_remove_vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
num	int	Standard
v_list[ ]	unsigned int	Standard

Calls	
Function	**Where Described**
spotter_remove_vehicles	Sec. 2.8.2.3.6
reporter_remove_vehicles	Sec. 2.8.2.1.3

Table 2.8-62: composite_remove_vehicles CSU [8.8.1.3.34]

2.8.1.3.35 composite_executing_order CSU

This CSU is an access point to commander_executing_order.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1

ReturnValues		
Return Value	**Type**	**Meaning**
commander_executing_order (...)	int	Normal end.

Calls	
Function	**Where Described**
composite_note_member_ vehicle_has_died	Sec. 2.8.1.3.38

Table 2.8-66: composite_note_member_vehicle_has_died CSU [8.8.1.3.38]

2.8.1.3.39 composite_rejoin_unit CSU

This CSU calls composite_cancel_overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
composite_cancel_overlay	Sec. 2.8.1.3.42	

Table 2.8-67: composite_rejoin_unit CSU [8.8.1.3.39]

2.8.1.3.40 composite_execute_overlay CSU

This CSU mission command is accomplished through calls to navigator_set_overlay and commander_execute_overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
navigator_set_overlay	Sec. 2.10.2.3.4	
commander_execute_overlay	Sec. 2.8.1.1.7	

Table 2.8-68: composite_execute_overlay CSU [8.8.1.3.40]

2.8.1.3.41 composite_readjust_overlay CSU

This CSU mission command is accomplished through a call to navigator_reset_overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
overlay	pointer to OVERLAY	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
navigator_reset_overlay	Sec. 2.10.2.3.5	

Table 2.8-69: composite_readjust_overlay CSU [8.8.1.3.41]

2.8.1.3.42 composite_cancel_overlay CSU

This CSU mission command is accomplished through a call to commander_cancel_overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
commander_cancel_overlay	Sec. 2.8.1.1.8	

Table 2.8-70: composite_cancel_overlay CSU [8.8.1.3.42]

2.8.1.3.43 composite_halt CSU

This CSU calls commander_halt.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
type	int	Standard
Calls		
Function	**Where Described**	
commander_halt	Sec. 2.8.1.1.11	

Table 2.8-71: composite_halt CSU [8.8.1.3.43]

2.8.1.3.44 composite_change_speed CSU

This CSU calls commander_change_speed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
speed	REAL	sim_types.h
Calls		
Function	**Where Described**	
commander_change_speed	Sec. 2.8.1.1.14	

Table 2.8-72: composite_change_speed CSU [8.8.1.3.44]

2.8.1.3.45 composite_follow_vehicle CSU

This CSU calls commander_follow_vehicle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
vid	unsigned int	Standard
xoff	int	Standard
yoff	int	Standard
Calls		
Function	**Where Described**	
commander_follow_vehicle	Sec. 2.8.1.1.17	

Table 2.8-73: composite_follow_vehicle CSU [8.8.1.3.45]

2.8.1.3.46 composite_simulator_in_command CSU

This CSU calls commander_simulator_in command.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
vid	unsigned int	Standard
Calls		
Function	**Where Described**	
commander_simulator_in_ command	Sec. 2.8.1.1.18	

Table 2.8-74: composite_simulator_in_command CSU [8.8.1.3.46]

2.8.1.3.47 composite_change_formation CSU

This CSU calls commander_change_formation.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
formation	pointer to char	Standard
Calls		
Function	**Where Described**	
commander_change_ formation	Sec. 2.8.1.1.16	

Table 2.8-75: composite_change_formation CSU [8.8.1.3.47]

2.8.1.3.48 composite_goto_point CSU

This CSU calls commander_goto_point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
x	REAL	sim_types.h
y	REAL	sim_types.h
backwardp	int	Standard
Calls		
Function	**Where Described**	
commander_goto_point	Sec. 2.8.1.1.19	

Table 2.8-76: composite_goto_point CSU [8.8.1.3.48]

2.8.1.3.49 composite_resume_mission CSU

This CSU calls commander_resume_mission.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
commander_resume_mission	Sec. 2.8.1.1.20	

Table 2.8-77: composite_resume_mission CSU [8.8.1.3.49]

2.8.1.3.50 composite_face_direction CSU

This CSU calls commander_face_direction.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
mathradians	REAL	sim_types.h
Calls		
Function	**Where Described**	
commander_face_direction	Sec. 2.8.1.1.21	

Table 2.8-78: composite_face_direction CSU [8.8.1.3.50]

2.8.1.3.51 composite_attack CSU

This CSU calls commander_attack.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
st_x	REAL	sim_types.h
st_y	REAL	sim_types.h
tar_x	REAL	sim_types.h
tar_y	REAL	sim_types.h
Calls		
Function	**Where Described**	
commander_attack	Sec. 2.8.1.1.27	

Table 2.8-79: composite_attack CSU [8.8.1.3.51]

2.8.1.3.52 composite_land CSU

This CSU calls commander_land.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
x	REAL	sim_types.h
y	REAL	sim_types.h
Calls		
Function	**Where Described**	
commander_land	Sec. 2.8.1.1.25	

Table 2.8-80: composite_land CSU [8.8.1.3.52]

2.8.1.3.53 composite_change_altitude CSU

This CSU calls commander_change_altitude.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
alt	REAL	sim_types.h
Calls		
Function	**Where Described**	
commander_change_altitude	Sec. 2.8.1.1.26	

Table 2.8-81: composite_change_altitude CSU [8.8.1.3.53]

2.8.1.4 formation.c CSC

/simnet/src/host/formation.c

This CSC takes the raw data in the formations parameter files and places vehicles into the proper positions. It decides who will follow whom in formation keeping, spacing, etc. Please note that whom a vehicle follows is not usually the vehicle from whom it takes orders.

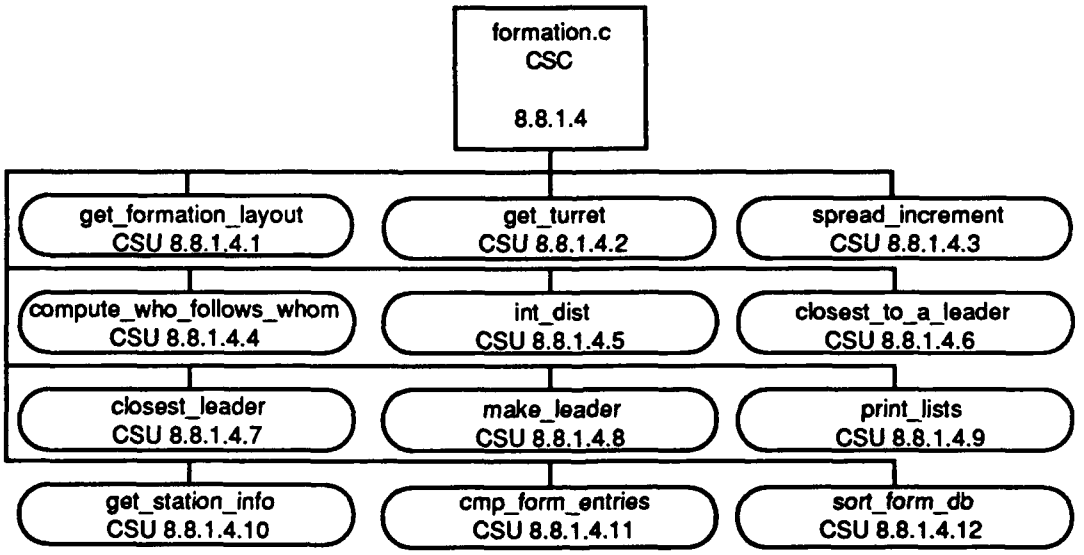


Figure 2.8-5: formation.c CSC Structure

In addition to the 12 CSUs detailed below, and shown in the above figure, the formation.c CSC contains a number of contant definitions and a type definition structure, to be found in the following tables.

Constant	Value
FORM_JOB	1
FORM_X	2
FORM_Y	3
SMALL_NUMBER	-0x40000000
BIG_NUMBER	0x40000000
TURRET_UNKNOWN	0
TURRET_FRONT	1
TURRET_LEFT	2
TURRET_RIGHT	3
SEARCHING /* used by make_leader */	0
SHIFTING /* used by make_leader */	1

Table 2.8-82: formation.c Constant Definitions

The following formation description typedef struct is tagged formation_desc.

Item	Type	Where Type Defined
x_pos	int	Standard
y_pos	int	Standard
xoff	int	Standard
yoff	int	Standard
leader /* index in job_list */	unsigned short	Standard
turret	unsigned short	Standard

Table 2.8-83: FORMATION_DESC Structure Definition

2.8.1.4.1 get_formation_layout CSU

This CSU derives a position for each vehicle, compensating for missing vehicles and extra vehicles. It then assigns turret orientations through recursive calls to get_turret.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
table	pointer to DATA_UNION	Sec. 2.1.1.5
formation	pointer to char	Standard
num_veh	unsigned short	Standard
job_list[ ]	pointer to char	Standard
fd_list[ ]	FORMATION_DESC	Sec. 2.8.1.4

Calls	
Function	**Where Described**
ft_table	Sec. 2.14.1.2.14
symbols_match	Sec. 2.1.1.5 See Appendix A
spread_increment	Sec. 2.8.1.4.3
compute_who_follows_whom	Sec. 2.8.1.4.4
get_turret	Sec. 2.8.1.4.2

Table 2.8-84: get_formation_layout CSU [8.8.1.4.1]

2.8.1.4.2 get_turret CSU

This CSU checks a vehicle for a turret direction. If the vehicle does not have one, an appropriate direction is assigned based on its leader's turret direction.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
for_whom	unsigned short	Standard
fd_list[ ]	FORMATION_DESC	Sec. 2.8.1.4
ReturnValues		
Return Value	**Type**	**Meaning**
fd_list[for_whom].turret	unsigned short	He already has turret direction.
fd_list[for_whom].turret= TURRET_FRONT	unsigned short	None of the othe rs.
fd_list[for_whom].turret= TURRET_LEFT	unsigned short	Leader is facing right.
fd_list[for_whom].turret= TURRET_RIGHT	unsigned short	Leader is facing left.
Calls		
Function	**Where Described**	
abs	Sec. 2.6.7.3 & Sec. 2.13.3.2 See Appendix A	
get_turret	Sec. 2.8.1.4.2	

Table 2.8-85: get_turret CSU [8.8.1.4.2]

2.8.1.4.3 spread_increment CSU

This CSU determines an appropriate vehicle spacing when units are missing.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
av	unsigned short	Standard
front	int	Standard
bv	unsigned short	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
front	int	Base no. of vehicles is 1 and actual no. of vehicles is 0.
(front / (bv - 1)) / av	int	Base no. of vehicles minus actual no. of vehicles equals 1.
previous_spread + this_spread	int	None of the above.

Calls	
Function	**Where Described**
spread_increment	Sec. 2.8.1.4.3

Table 2.8-86: spread_increment CSU [8.8.1.4.3]

2.8.1.4.4 compute_who_follows_whom CSU

This CSU determines which vehicle each vehicle should follow.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
num_veh	unsigned short	Standard
fd_list[ ]	FORMATION_DESC	Sec. 2.8.1.4
ymax	int	Standard
xmid	int	Standard

Calls	
Function	**Where Described**
abs	Sec. 2.6.7.3 & Sec. 2.13.3.2 See Appendix A
make_leader	Sec. 2.8.1.4.8
print_lists	Sec. 2.8.1.4.9
closest_to_a_leader	Sec. 2.8.1.4.6
DEBUG_COMPOSITE	Sec. 2.5.2.2 See Appendix A
closest_leader	Sec. 2.8.1.4.7

Table 2.8-87: compute_who_follows_whom CSU [8.8.1.4.4]

2.8.1.4.5 int_dist CSU

This CSU returns the sum of the squares of the input parameters, x and y.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x	int	Standard
y	int	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
x*x + y*y	int	Sum of squares x and y.

Table 2.8-88: int_dist CSU [8.8.1.4.5]

2.8.1.4.6 closest_to_a_leader CSU

This CSU returns the index of the vehicle closest to a specified vehicle already on the leader list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
leader_list[ ]	unsigned short	Standard
follower_list[ ]	unsigned short	Standard
num_leaders	unsigned short	Standard
num_followers	unsigned short	Standard
x_list[ ]	int	Standard
y_list[ ]	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
closest_so_far	unsigned short	Closest unit to leader.
Calls		
Function	**Where Described**	
int_dist	Sec. 2.8.1.4.5	

Table 2.8-89: closest_to_a_leader CSU [8.8.1.4.6]

2.8.1.4.7 closest_leader CSU

This CSU returns the index of the vehicle on the leader list closest to the specified vehicle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
to_whom	unsigned short	Standard
leader_list[ ]	unsigned short	Standard
num_leaders	unsigned short	Standard
x_list[ ]	int	Standard
y_list[ ]	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
closest_so_far	unsigned short	Closest follower
Calls		
Function	**Where Described**	
int_dist	Sec. 2.8.1.4.5	

Table 2.8-90: closest_leader CSU [8.8.1.4.7]

2.8.1.4.8 make_leader CSU

This CSU removes a vehicle from the follower list and adds it to the leader list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
who	unsigned short	Standard
leader_list[ ]	unsigned short	Standard
follower_list[ ]	unsigned short	Standard
num_leaders	pointer to unsigned short	Standard
num_followers	pointer to unsigned short	Standard

Table 2.8-91: make_leader CSU [8.8.1.4.8]

2.8.1.4.9 print_lists CSU

This CSU displays a list of leaders and followers.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
lead[ ]	unsigned short	Standard
foll[ ]	unsigned short	Standard
nlead	unsigned short	Standard
nfoll	unsigned short	Standard

Table 2.8-92: print_lists CSU [8.8.1.4.9]

2.8.1.4.10 get_station_info CSU

This CSU collects vehicle information, finds positions for these vehicles, and returns the positions.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
table	pointer to DATA_UNION	Sec. 2.1.1.5
formation	pointer to char	Standard
num_veh	unsigned short	Standard
veh_list[ ]	pointer to SAF_OBJECT	Sec. 2.9.1.1
id_list[ ]	unsigned int	Standard
xoff_list[ ]	int	Standard
yoff_list[ ]	int	Standard
x_pos_list[ ]	int	Standard
y_pos_list[ ]	int	Standard
turret_left_list[ ]	REAL	sim_types.h
turret_right_list[ ]	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
unit_leader	pointer to SAF_OBJECT	The unit lead vehicle.

Calls	
Function	**Where Described**
saf_vehicle_useless	Sec. 2.6.1.1.41
get_formation_layout	Sec. 2.8.1.4.1
deg_to_rad	sim_macros.h
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A

Table 2.8-93: get_station_info CSU [8.8.1.4.10]

2.8.1.4.11 cmp_form_entries CSU

This CSU, given two entries in the formation database, determines which is more front and center.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
fe0	pointer to pointer to DATA_UNION	Sec. 2.1.1.5
fe1	pointer to pointer to DATA_UNION	Sec. 2.1.1.5
ReturnValues		
Return Value	**Type**	**Meaning**
(by-ay)	int	Furthest forward comes first.
(ax-bx)	int	Furthest to the left comes first.

Table 2.8-94: cmp_form_entries CSU [8.8.1.4.11]

2.8.1.4.12 sort_form_db CSU

This CSU sorts the formation database such that the most front-and-center vehicle appears first in each group.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
table	pointer to DATA_UNION	Sec. 2.1.1.5
Calls		
Function	**Where Described**	
qsort	Sec. 2.	

Table 2.8-95: sort_form_db CSU [8.8.1.4.12]

2.8.2 Reports CSC

2.8.2.1 reporter.c CSC

/simnet/src/host/reporter.c

This code handles the clustering of groups of vehicles and sets of artillery impacts, reporting them to the user.

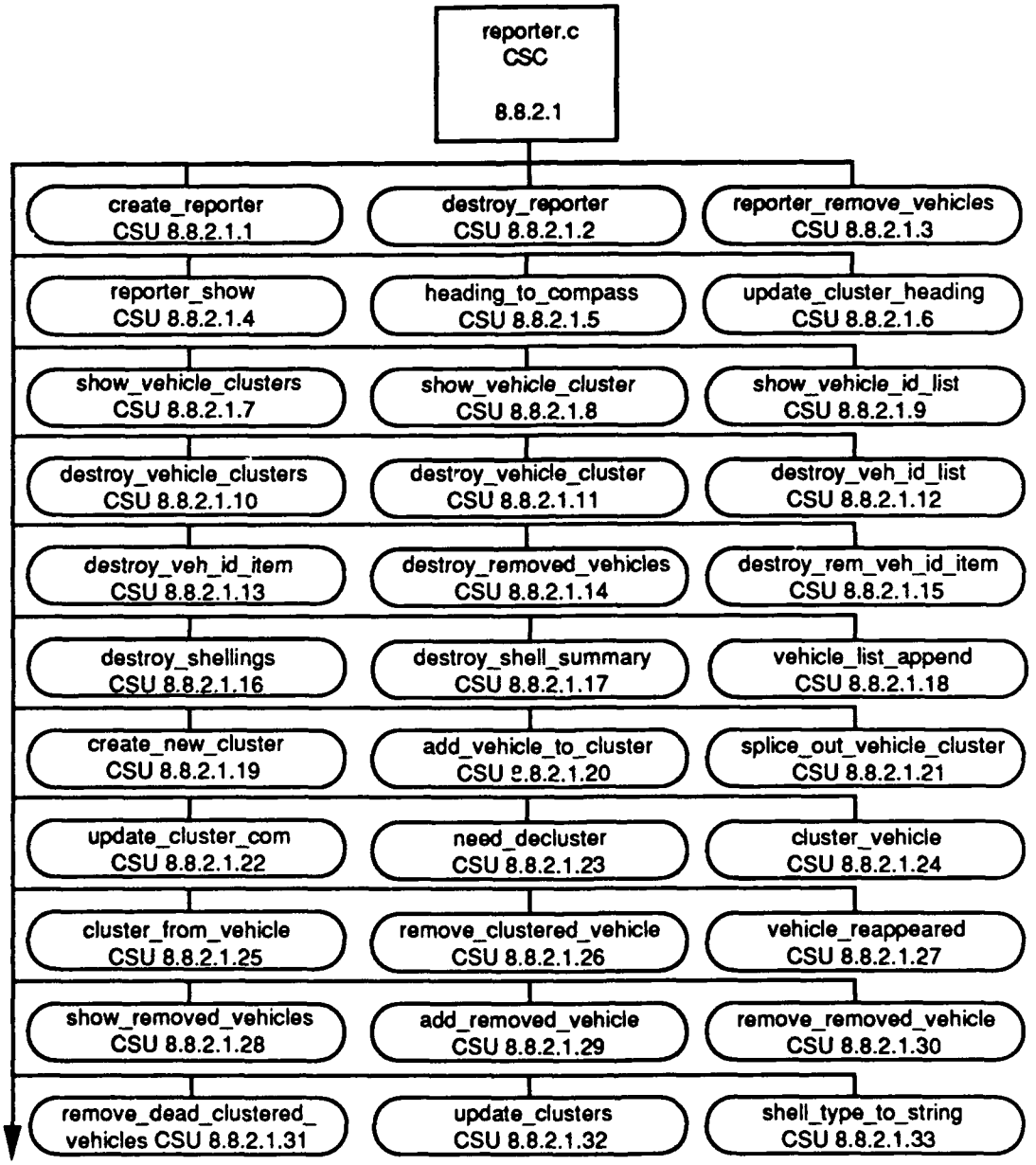


Figure 2.8-6: reporter.c CSC Structure Part 1 of 2

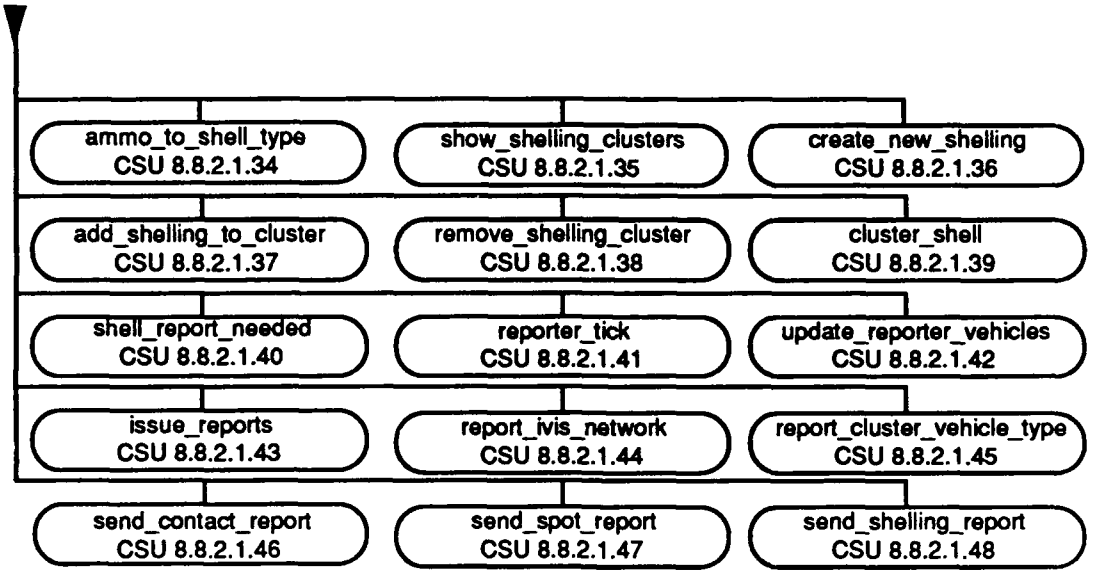


Figure 2.8-7: reporter.c CSC Structure Part 2 of 2

2.8.2.1.1 create_reporter CSU

This CSU allocates memory for a reporter's REPORTER_VARS structure and its current and old vehicle bitfield. It initializes the reporter's structure, and returns a pointer to that structure.

ReturnValues		
Return Value	**Type**	**Meaning**
reporter	pointer to REPORTER_VARS	Created reporter.

Calls	
Function	**Where Described**
allocate_reporter	Sec. 2.9.1.2 See Appendix A
allocate_bitfield	Sec. 2.14.3.4.1
clear_bitfield	Sec. 2.14.3.4.6

Table 2.8-96: create_reporter CSU [8.8.2.1.1]

2.8.2.1.2 destroy_reporter CSU

This CSU destroys the records of the reporter and deallocates its structure memory heap.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
reporter	pointer to REPORTER_VARS	Sec. 2.9.1.2 veh_storage.h

Calls	
Function	**Where Described**
destroy_vehicle_clusters	Sec. 2.8.2.1.10
destroy_removed_vehicles	Sec. 2.8.2.1.14
destroy_shellings	Sec. 2.8.2.1.16
deallocate_bitfield	Sec. 2.14.3.4.2
deallocate_reporter	Sec. 2.9.1.2 See Appendix A

Table 2.8-97: destroy_reporter CSU [8.8.2.1.2]

2.8.2.1.3 reporter_remove_vehicles CSU

This CSU removes the clustered vehicles passed as parameters and clears the reporter's current and old vehicle tables for these vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
num	int	Standard
v_list[ ]	unsigned int	Standard

Calls	
Function	**Where Described**
remove_clustered_vehicle	Sec. 2.8.2.1.26
clear_bit	Sec. 2.14.3.4.4

Table 2.8-98: reporter_remove_vehicles CSU [8.8.2.1.3]

2.8.2.1.4 reporter_show CSU

This CSU displays the reporter's current information concerning vehicles and shellings.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
reporter	pointer to REPORTER_VARS	Sec. 2.9.1.2 veh_storage.h
flags	int	Standard

Calls	
Function	**Where Described**
show_vehicle_clusters	Sec. 2.8.2.1.6
show_removed_vehicles	Sec. 2.8.2.1.28
show_shelling_clusters	Sec. 2.8.2.1.35

Table 2.8-99: reporter_show CSU [8.8.2.1.4]

2.8.2.1.5 heading_to_compass CSU

This CSU converts the angle passed into a compass-point-heading character string containing the direction if the angle passed is not for a stationary heading or if it is too large.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
angle	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
"stationary or heading in and terminated"	pointer to char	angle == STATIONARY_ HEADING
"heading West"	pointer to char	angle < (-7.0 / 8.0) * PI
"heading Southwest"	pointer to char	angle < (-5.0 / 8.0) * PI
"heading South"	pointer to char	angle < (-3.0 / 8.0) * PI
"heading Southeast"	pointer to char	angle < (-1.0 / 8.0) * PI
"heading East"	pointer to char	angle < (1.0 / 8.0) * PI
"heading Northeast"	pointer to char	angle < (3.0 / 8.0) * PI
"heading North"	pointer to char	angle < (5.0 / 8.0) * PI
"heading Northwest"	pointer to char	angle < (7.0 / 8.0) * PI
"heading West"	pointer to char	angle <= (8.0 / 8.0) * PI
"Unknown Direction"	pointer to char	Direction is unknown.

Table 2.8-100: heading_to_compass CSU [8.8.2.1.5]

2.8.2.1.6 update_cluster_heading CSU

This CSU updates a cluster heading either to the new heading or to its stationary heading if insignificant change has occurred.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
old_com	pointer to REAL	sim_types.h
Calls		
Function	**Where Described**	
vec_sub	Sec. 2.6.2.65.1 Vehicles CSCI SDD	
vec2_dot	Sec. 2.14.3.5.24	
s_atan2	Sec. 2.14.3.9 See Appendix A	

Table 2.8-101: update_cluster_heading CSU [8.8.2.1.6]

2.8.2.1.7 show_vehicle_clusters CSU

This CSU calls show_vehicle_cluster for each of the clusters.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
show_vehicle_cluster	Sec. 2.8.2.1.8	

Table 2.8-102: show_vehicle_clusters CSU [8.8.2.1.7]

2.8.2.1.8 show_vehicle_cluster CSU

This CSU displays a status report on the cluster passed including a list of the vehicle identifications within the cluster.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
heading_to_compass	Sec. 2.8.2.1.5	
show_vehicle_id_list	Sec. 2.8.2.1.9	

Table 2.8-103: show_vehicle_cluster CSU [8.8.2.1.8]

2.8.2.1.9 show_vehicle_id_list CSU

This CSU displays the vehicle identification list for the cluster.

Parameters		
vehicles	pointer to VEH_ID_LIST	Sec. 2.8.2.2

Table 2.8-104: show_vehicle_id_list CSU [8.8.2.1.9]

2.8.2.1.10 destroy_vehicle_clusters CSU

This CSU, through multiple calls to destroy_vehicle_cluster, destroys all the clusters passed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
clusters	pointer to VEH_CLUSTER	Sec. 2.8.2.2

Calls	
Function	**Where Described**
destroy_vehicle_cluster	Sec. 2.8.2.1.11

Table 2.8-105: destroy_vehicle_clusters CSU [8.8.2.1.10]

2.8.2.1.11 destroy_vehicle_cluster CSU

This CSU destroys a vehicle cluster by destroying the vehicle identification list, setting the cluster->vehicles element in the passed VEH_CLUSTER structure to NULL, and deallocating the memory heap for the cluster structure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2

Calls	
Function	**Where Described**
destroy_veh_id_list	Sec. 2.8.2.1.12
deallocate_veh_cluster	Sec. 2.8.2.2 See Appendix A

Table 2.8-106: destroy_vehicle_cluster CSU [8.8.2.1.11]

2.8.2.1.12 destroy_veh_id_list CSU

This CSU destroys a vehicle identification list through calls to destroy_vehicle_id_item for each vehicle in the VEH_ID_LIST.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vehicles	pointer to VEH_ID_LIST	Sec. 2.8.2.2

Calls	
Function	**Where Described**
destroy_veh_id_item	Sec. 2.8.1.2.13

Table 2.8-107: destroy_veh_id_list CSU [8.8.2.1.12]

2.8.2.1.13 destroy_veh_id_item CSU

This CSU destroys a vehicle identification item through a call to deallocate_veh_id_list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vehicle	pointer to VEH_ID_LIST	Sec. 2.8.2.2

Calls	
Function	**Where Described**
deallocate_veh_id_list	Sec. 2.8.2.2 See Appendix A

Table 2.8-108: destroy_veh_id_item CSU [8.8.2.1.13]

2.8.2.1.14 destroy_removed_vehicles CSU

This CSU destroys a removed vehicle identification list through calls to destroy_rem_veh_id_item for each vehicle in the REM_VEH_ID_LIST.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vehicle	pointer to REM_VEH_ID_LIST	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
destroy_rem_veh_id_item	Sec. 2.8.2.1.15	

Table 2.8-109: destroy_removed_vehicles CSU [8.8.2.1.14]

2.8.2.1.15 destroy_rem_veh_id_item CSU

This CSU destroys a removed vehicle identification item through a call to deallocate_rem_veh_id_list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vehicle	pointer to REM_VEH_ID_LIST	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
deallocate_rem_veh_id_list	Sec. 2.8.2.2 See Appendix A	

Table 2.8-110: destroy_rem_veh_id_item CSU [8.8.2.1.15]

2.8.2.1.16 destroy_shellings CSU

This CSU destroys each shell summary in the SHELL_SUMMARY structure, through calls to destroy_shell_summary.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
shell	pointer to SHELL_SUMMARY	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
destroy_shell_summary	Sec. 2.8.2.1.17	

Table 2.8-111: destroy_shellings CSU [8.8.2.1.16]

2.8.2.1.17 destroy_shell_summary CSU

This CSU destroys a shell summary structure through a call to deallocate_shell_summary.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
shell	pointer to SHELL_SUMMARY	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
deallocate_shell_summary	Sec. 2.8.2.2 See Appendix A	

Table 2.8-112: destroy_shell_summary CSU [8.8.2.1.17]

2.8.2.1.18 vehicle_list_append CSU

This CSU appends two vehicle lists and returns the results.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
l1	pointer to VEH_ID_LIST	Sec. 2.8.2.2
l2	pointer to VEH_ID_LIST	Sec. 2.8.2.2
ReturnValues		
Return Value	**Type**	**Meaning**
l1	pointer to VEH_ID_LIST	Only l1 exists or l2 appended.
l2	pointer to VEH_ID_LIST	Only l2 exists.

Table 2.8-113: vehicle_list_append CSU [8.8.2.1.18]

2.8.2.1.19 create_new_cluster CSU

This CSU creates a new vehicle cluster containing one vehicle, the vehicle passed as an argument to the function.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
veh_id	unsigned int	Standard
monitor_time	unsigned int	Standard
report_status		
ReturnValues		
Return Value	**Type**	**Meaning**
cluster	pointer to register VEH_CLUSTER	Created cluster.

Calls	
Function	**Where Described**
LOOKUP_POSITION	Sec. 2.9.1.1 See Appendix A
LOOKUP_VELOCITY	Sec. 2.9.1.1 See Appendix A
allocate_veh_cluster	Sec. 2.8.2.2 See Appendix A
vec_copy	Sec. 2.6.2.59.1 Vehicles CSCI SDD
vec2_dot	Sec. 2.14.3.5.24
s_atan2	Sec. 2.14.3.9 See Appendix A
allocate_veh_id_list	Sec. 2.8.2.2 See Appendix A

Table 2.8-114: create_new_cluster CSU [8.8.2.1.19]

2.8.2.1.20 add_vehicle_to_cluster CSU

This CSU adds a vehicle to an existing cluster. The center of mass of the cluster is then updated through a call to update_cluster_com.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
veh_id	unsigned int	Standard
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
report_status		

Errors	
Error Name	**Reason for Error**
"Attempt to add vehicle ... to empty cluster."	There are no vehicles in the cluster.

Calls	
Function	**Where Described**
allocate_veh_id_list	Sec. 2.8.2.2 See Appendix A
update_cluster_com	Sec. 2.8.2.1.22

Table 2.8-115: add_vehicle_to_cluster CSU [8.8.2.1.20]

2.8.2.1.21 splice_out_vehicle_cluster CSU

This CSU removes a vehicle cluster but does not deallocate the list of vehicles within it.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2

Calls	
Function	**Where Described**
deallocate_veh_cluster	Sec. 2.8.2.2 See Appendix A

Table 2.8-116: splice_out_vehicle_cluster CSU [8.8.2.1.21]

2.8.2.1.22 update_cluster_com CSU

This CSU recomputes the center of mass of a cluster.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
vec_init	Sec. 2.6.2.61.1 Vehicles CSCI SDD	
vec_add	Sec. 2.6.2.57.1 Vehicles CSCI SDD	
LOOKUP_POSITION	Sec. 2.9.1.1 See Appendix A	
vec_scale	Sec. 2.6.2.64.1 Vehicles CSCI SDD	

Table 2.8-117: update_cluster_com CSU [8.8.2.1.22]

2.8.2.1.23 need_decluster CSU

This CSU checks if a cluster needs to be broken up because its vehicles have moved too far from the cluster's center of mass.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
thresh2	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	A vehicle is too far away.
FALSE	int	No vehicles are too far away.
Calls		
Function	**Where Described**	
vec2_range_squared	Sec. 2.14.3.5.30	
LOOKUP_POSITION	Sec. 2.9.1.1 See Appendix A	

Table 2.8-118: need_decluster CSU [8.8.2.1.23]

2.8.2.1.24 cluster_vehicle CSU

This CSU adds a vehicle to the vehicle's cluster system. An existing cluster may be augmented, or a new cluster may be added.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
veh_id	unsigned int	Standard
report_status	int	Standard

Calls	
Function	**Where Described**
DEBUG_REPORTER	Sec. 2.5.2.2 See Appendix A
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A
create_new_cluster	Sec. 2.8.2.1.19
vec2_range_squared	Sec. 2.14.3.5.30
add_vehicle_to_cluster	Sec. 2.8.2.1.20

Table 2.8-119: cluster_vehicle CSU [8.8.2.1.24]

2.8.2.1.25 cluster_from_vehicle CSU

This CSU finds the cluster inwhich a vehicle exists. If the vehicle is not in any cluster, NULL is returned.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
veh_id	unsigned int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
NULL	pointer to VEH_CLUSTER	Vehicle not in cluster.
cluster	pointer to VEH_CLUSTER	Cluster containing vehicle.

Table 2.8-120: cluster_from_vehicle CSU [8.8.2.1.25]

2.8.2.1.26 remove_clustered_vehicle CSU

This CSU removes a vehicle from its cluster system. Depending on the passed argument, the removed vehicle may be recorded in the removed_vehicle_list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
veh_id	unsigned int	Standard
save_him	int	Standard
Calls		
Function	**Where Described**	
DEBUG_REPORTER	Sec. 2.5.2.2 See Appendix A	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
add_removed_vehicle	Sec. 2.8.2.1.29	
destroy_veh_id_item	Sec. 2.8.2.1.13	
splice_out_vehicle_cluster	Sec. 2.8.2.1.21	

Table 2.8-121: remove_clustered_vehicle CSU [8.8.2.1.26]

2.8.2.1.27 vehicle_reappeared CSU

This CSU searches through a removed_vehicle_list for the specified input vehicle, and removes it from the list if it is there.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
id	unsigned int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Vehicle removed from list.
FALSE	int	Vehicle not found in list.
Calls		
Function	**Where Described**	
remove_removed_vehicle	Sec. 2.8.2.1.30	
DEBUG_REPORTER	Sec. 2.5.2.2 See Appendix A	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	

Table 2.8-122: vehicle_reappeared CSU [8.8.2.1.27]

2.8.2.1.28 show_removed_vehicles CSU

This CSU displays all the recorded removed vehicles in the input removed_vehicle_list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vehicles	pointer to REM_VEH_ID_LIST	Sec. 2.8.2.2

Table 2.8-123: show_removed_vehicles CSU [8.8.2.1.28]

2.8.2.1.29 add_removed_vehicle CSU

This CSU adds a vehicle to the removed_vehicle_list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
id	unsigned int	Standard
Calls		
Function	**Where Described**	
allocate_rem_veh_id_list	Sec. 2.8.2.2 See Appendix A	

Table 2.8-124: add_removed_vehicle CSU [8.8.2.1.29]

2.8.2.1.30 remove_removed_vehicle CSU

This CSU removes a vehicle from the removed_vehicle_list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
removed	pointer to REM_VEH_ID_LIST	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
destroy_rem_veh_id_item	Sec. 2.8.2.1.15	

Table 2.8-125: remove_removed_vehicle CSU [8.8.2.1.30]

2.8.2.1.31 remove_dead_clustered_vehicles CSU

This CSU removes all dead vehicles from the vehicle cluster system. If a cluster has all of its vehicles removed, that cluster is deallocated.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
is_dead	Sec. 2.13.3.1 See Appendix A	
LOOKUP_APPEARANCE	Sec. 2.9.1.1 See Appendix A	
LOOKUP_VEHICLECLASS	Sec. 2.9.1.1 See Appendix A	
DEBUG_REPORTER	Sec. 2.5.2.2 See Appendix A	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
destroy_veh_id_item	Sec. 2.8.2.1.13	
deallocate_veh_cluster	Sec. 2.8.2.2 See Appendix A	

Table 2.8-126: remove_dead_clustered_vehicles CSU [8.8.2.1.31]

2.8.2.1.32 update_clusters CSU

This CSU removes the dead clustered vehicles, recalculates all cluster centers of mass and headings, declusters all clusters that are too spread out or too close to each other, and reclusters all of the declustered vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1

Calls	
Function	**Where Described**
remove_dead_clustered_ vehicles	Sec. 2.8.2.1.31
vec_copy	Sec. 2.6.2.59.1 Vehicles CSCI SDD
update_cluster_com	Sec. 2.8.2.1.22
update_cluster_heading	Sec. 2.8.2.1.6
need_decluster	Sec. 2.8.2.1.23
DEBUG_REPORTER	Sec. 2.5.2.2 See Appendix A
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A
vehicle_list_append	Sec. 2.8.2.1.18
splice_out_vehicle_cluster	Sec. 2.8.2.1.21
vec2_range_squared	Sec. 2.14.3.5.30
cluster_vehicle	Sec. 2.8.2.1.24
destroy_veh_id_list	Sec. 2.8.2.1.12

Table 2.8-127: update_clusters CSU [8.8.2.1.32]

2.8.2.1.33 shell_type_to_string CSU

This CSU returns a human-readable string indicating what shell type was passed as the argument.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
type	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
"Mortar"	pointer to char	Type is SHELL_TYPE_MORTAR.
"Artillery"	pointer to char	Type is SHELL_TYPE_ARTILLERY.
"Mine"	pointer to char	Type is SHELL_TYPE_MINE.
"unknown indirect fire"	pointer to char	Type is none of the above.

Table 2.8-128: shell_type_to_string CSU [8.8.2.1.33]

2.8.2.1.34 ammo_to_shell_type CSU

This CSU extracts the shell type from the ammo object type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
ammo	ObjectType	p_sim.h

ReturnValues		
Return Value	**Type**	**Meaning**
SHELL_TYPE_MINE	int	Ammo is mine.
SHELL_TYPE_ARTILLERY	int	Ammo is artillery.
SHELL_TYPE_MORTAR	int	Ammo is mortar.

Table 2.8-129: ammo_to_shell_type CSU [8.8.2.1.34]

2.8.2.1.35 show_shelling_clusters CSU

This CSU displays information about each shelling cluster (e.g., center of mass, shell type, and number of shells).

Parameters		
Parameters	**Type**	**Where Typedef Declared**
shell	pointer to SHELL_SUMMARY	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
shell_type_to_string	Sec. 2.8.2.1.33	

Table 2.8-130: show_shelling_clusters CSU [8.8.2.1.35]

2.8.2.1.36 create_new_shelling CSU

This CSU creates a new shelling cluster.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
pos	pointer to REAL	sim_types.h
type	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
shell	pointer to SHELL_SUMMARY	Created shelling cluster.
Calls		
Function	**Where Described**	
allocate_shell_summary	Sec. 2.8.2.2 See Appendix A	
vec_copy	Sec. 2.6.2.59.1 Vehicles CSCI SDD	

Table 2.8-131: create_new_shelling CSU [8.8.2.1.36]

2.8.2.1.37 add_shelling_to_cluster CSU

This CSU adds a shelling to an existing shelling cluster, then updates the cluster's center of mass.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
pos	pointer to REAL	sim_types.h
shell	pointer to SHELL_SUMMARY	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
vec_scale	Sec. 2.6.2.64.1 Vehicles CSCI SDD	
vec_add	Sec. 2.6.2.57.1 Vehicles CSCI SDD	

Table 2.8-132: add_shelling_to_cluster CSU [8.8.2.1.37]

2.8.2.1.38 remove_shelling_cluster CSU

This CSU removes a shelling cluster from a vehicle's shelling cluster system.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
reporter	pointer to REPORTER_VARS	Sec. 2.9.1.2 veh_storage.h
shell	pointer to SHELL_SUMMARY	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
deallocate_shell_summary	Sec. 2.8.2.2 See Appendix A	

Table 2.8-133: remove_shelling_cluster CSU [8.8.2.1.38]

2.8.2.1.39 cluster_shell CSU

This CSU adds a shelling to a vehicle's shelling cluster system if the shelling is close enough to be significant.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
pos	pointer to REAL	sim_types.h
ammo	ObjectType	p_sim.h

Calls	
Function	**Where Described**
ammo_to_shell_type	Sec. 2.8.2.1.34
vec2_range_squared	Sec. 2.14.3.5.30
OBJ_POSITION	Sec. 2.5.2.2 See Appendix A
DEBUG_REPORTER	Sec. 2.9.1.1 See Appendix A
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A
create_new_shelling	Sec. 2.8.2.1.36
add_shelling_to_cluster	Sec. 2.8.2.1.37

Table 2.8-134: cluster_shell CSU [8.8.2.1.39]

2.8.2.1.40 shell_report_needed CSU

This CSU checks if a shelling report is needed based on the presence of new shelling or the elapse of a specified duration of time since the last report.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
shell	pointer to SHELL_SUMMARY	Sec. 2.8.2.2

ReturnValues		
Return Value	**Type**	**Meaning**
(shell->time_of_rep == 0) || (g_last_millisecond_time - shell->time_of_rep > g_report_monitor_time) || (g_last_millisecond_time - shell-->time_of_last_shell > DEFAULT_SHELL_ LATENCY)	int	Normal end.

Table 2.8-135: shell_report_needed CSU [8.8.2.1.40]

2.8.2.1.41 reporter_tick CSU

This CSU is the primary interface to the reporter. Clusters and vehicle information are updated periodically, and reports are issued.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1

Calls	
Function	**Where Described**
update_clusters	Sec. 2.8.2.1.32
get_me_a_random_fraction	Sec. 2.14.3.7.1
update_reporter_vehicles	Sec. 2.8.4.1.42
issue_reports	Sec. 2.8.4.1.43

Table 2.8-136: reporter_tick CSU [8.8.2.1.41]

2.8.2.1.42 update_reporter_vehicles CSU

This CSU obtains new information about what vehicles are visible from the spotter. Changes in this information cause vehicles to be clustered or declustered.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
copy_bitfield	Sec. 2.14.3.4.5	
extract_bit	Sec. 2.14.3.9 See Appendix A	
is_dead	Sec. 2.13.3.1 See Appendix A	
LOOKUP_APPEARANCE	Sec. 2.9.1.1 See Appendix A	
LOOKUP_VEHICLECLASS	Sec. 2.9.1.1 See Appendix A	
vehicle_reappeared	Sec. 2.8.2.1.27	
cluster_vehicle	Sec. 2.8.2.1.24	
remove_clustered_vehicle	Sec. 2.8.2.1.26	

Table 2.8-137: update_reporter_vehicles CSU [8.8.2.1.42]

2.8.2.1.43 issue_reports CSU

This CSU causes contact, spot and shelling reports to be issued.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
send_contact_report	Sec. 2.8.2.1.46	
vec2_range_squared	Sec. 2.14.3.5.30	
send_spot_report	Sec. 2.8.2.1.47	
shell_report_needed	Sec. 2.8.2.1.40	
send_shelling_report	Sec. 2.8.2.1.48	
DEBUG_REPORTER	Sec. 2.9.1.1 See Appendix A	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
remove_shelling_cluster	Sec. 2.8.2.1.38	

Table 2.8-138: issue_reports CSU [8.8.2.1.43]

2.8.2.1.44 report_ivis_network CSU

This CSU computes an IVIS network based on an input text string.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
text	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
a1P1tNetwork	int	Input text: A1 or 11
a2P1tNetwork	int	Input text: A2 or 12
a3P1tNetwork	int	Input text: A3 or 13
aCoNetwork	int	Input text: An or 1n (n != 1, 2, or 3)
b1P1tNetwork	int	Input text: B1 or 21
b2P1tNetwork	int	Input text: B2 or 22
b3P1tNetwork	int	Input text: B3 or 23
bCoNetwork	int	Input text: Bn or 2n (n != 1, 2, or 3)
c1P1tNetwork	int	Input text: C1 or 31
c2P1tNetwork	int	Input text: C2 or 32
c3P1tNetwork	int	Input text: C3 or
cCoNetwork	int	Input text: Cn or 3 (n != 1, 2, or 3)
d1P1tNetwork	int	Input text: D1 or 41
d2P1tNetwork	int	Input text: D2 or 42
d3P1tNetwork	int	Input text: D3 or 43
dCoNetwork	int	Input text: Dn or 4n (n != 1, 2, or 3)
bnCommandNetwork	int	Input text is none of the above.

Table 2.8-139: report_ivis_network CSU [8.8.2.1.44]

2.8.2.1.45 report_cluster_vehicle_type CSU

This CSU returns the predominant vehicle type in a cluster.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
ReturnValues		
Return Value	**Type**	**Meaning**
type	int	Normal end.
Calls		
Function	**Where Described**	
LOOKUP_OBJECT_TYPE	Sec. 2.9.1.1 See Appendix A	

Table 2.8-140: report_cluster_vehicle_type CSU [8.8.2.1.45]

2.8.2.1.46 send_contact_report CSU

This CSU sends contact reports for a cluster, both throughout the IVIS network and to the Symbolics.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
buffer_allocate	Sec. 2.14.4.2.12	
tdb_giv_xy_get_utm	Sec. 2.21.7.24.3	
heading_to_compass	Sec. 2.8.2.1.5	
report_ivis_network	Sec. 2.8.2.1.44	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
report_cluster_vehicle_type	Sec. 2.8.2.1.45	
ivis_send_contact_report		
fill_sbx_opfor_header	Sec. 2.4.3.2.18	
sbx_connection_send_to_port	Sec. 2.4.3.2.19	
buffer_deallocate	Sec. 2.14.4.2.15	
vec_copy	Sec. 2.6.2.59.1 Vehicles CSCI SDD	

Table 2.8-141: send_contact_report CSU [8.8.2.1.46]

2.8.2.1.47 send_spot_report CSU

This CSU sends spot reports for a cluster, both throughout the IVIS network and to the Symbolics.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
Calls		
Function	**Where Described**	
buffer_allocate	Sec. 2.14.4.2.12	
tdb_giv_xy_get_utm	Sec. 2.21.7.24.3	
heading_to_compass	Sec. 2.8.2.1.5	
report_ivis_network	Sec. 2.8.2.1.44	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
report_cluster_vehicle_type	Sec. 2.8.2.1.45	
ivis_send_spot_report		
fill_sbx_opfor_header	Sec. 2.4.3.2.18	
sbx_connection_send_to_port	Sec. 2.4.3.2.19	
buffer_deallocate	Sec. 2.14.4.2.15	
vec_copy	Sec. 2.6.2.59.1 Vehicles CSCI SDD	

Table 2.8-142: send_spot_report CSU [8.8.2.1.47]

2.8.2.1.48 send_shelling_report CSU

This CSU sends shelling reports for a cluster, both throughout the IVIS network and to the Symbolics.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
shell	pointer to SHELL_SUMMARY	Sec. 2.8.2.2
ReturnValues		
Return Value	**Type**	**Meaning**
result /* TRUE */	int	Shells still impacting.
result /* FALSE */	int	Shells stopped impacting.
Calls		
Function	**Where Described**	
buffer_allocate	Sec. 2.14.4.2.12	
tdb_giv_xy_get_utm	Sec. 2.21.7.24.3	
report_ivis_network	Sec. 2.8.2.1.44	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
ivis_send_shell_report		
fill_sbx_opfor_header	Sec. 2.4.3.2.18	
sbx_connection_send_to_port	Sec. 2.4.3.2.19	
buffer_deallocate	Sec. 2.14.4.2.15	

Table 2.8-143: send_shelling_report CSU [8.8.2.1.48]

2.8.2.2 reporter.h CSU

/simnet/src/host/reporter.h

This CSU contains the structures, symbolic constants, and macro definitions (shown in Appendix A) used by the reporter code.

Constant	Value
SHELL_TYPE_MORTAR	0
SHELL_TYPE_ARTILLERY	1
SHELL_TYPE_MINE	2
DEFAULT_CLUSTER_DISTANCE	1000.0
DEFAULT_DECLUSTER_DISTANCE	1200.0
DEFAULT_SPOT_REP_RANGE_THRESHOLD	1000.0
DEFAULT_MAX_REAPPEAR_LATENCY	(3 * 60 * 1000)
DEFAULT_SHELL_RANGE_THRESHOLD	2000.0
DEFAULT_SHELL_DISTANCE	1000.0
DEFAULT_SHELL_LATENCY	(60 * 1000)
DEFAULT_REPORT_MONITOR_TIME	(5 * 60 * 1000)
REPORTER_VELOCITY_THRESHOLD2	(3.0 * 3.0)
REPORTER_MOVEMENT_THRESHOLD2	(400.0 * 400.0)
STATIONARY_HEADING	(100 * PI)
REPORTER_NEW_VEHICLE_INTERVAL	20000
REPORTER_NEW_VEHICLE_VARIANCE	10000
REPORTER_UPDATE_INTERVAL	10000
REPORTER_UPDATE_VARIANCE	1000
NO_REPORT_NEEDED	0
CONTACT_REPORT_NEEDED	1
SPOT_REPORT_NEEDED	2
NON_PROMOTABLE_SPOT_REPORT_NEEDED	3

Table 2.8-144: reporter.h Constant Definitions

The following typedef struct is tagged SHELL_SUMMARY.

Item	Type	Where Type Defined
type	int	Standard
num_shells	short	Standard
last_num_shells	short	Standard
time_of_rep	unsigned_int	Standard
time_of_last_shell	unsigned_int	Standard
com	VECTOR	sim_types.h
next	pointer to struct shell_summary	This struct
prev	pointer to struct shell_summary	This struct

Table 2.8-145: SHELL_SUMMARY Structure Definition

The following typedef struct is tagged VEH_ID_LIST.

Item	Type	Where Type Defined
id	unsigned_int	Standard
next	pointer to struct veh_id_list	This struct
prev	pointer to struct veh_id_list	This struct

Table 2.8-146: VEH_ID_LIST Structure Definition

The following typedef struct is tagged REM_VEH_ID_LIST.

Item	Type	Where Type Defined
id	unsigned_int	Standard
time_disappeared	unsigned_int	Standard
next	pointer to struct rem_veh_id_list	This struct
prev	pointer to struct rem_veh_id_list	This struct

Table 2.8-147: REM_VEH_ID_LIST Structure Definition

The following typedef struct is tagged VEH_CLUSTER.

Item	Type	Where Type Defined
report_status	unsigned_int	Standard
time_of_rep	unsigned_int	Standard
com	VECTOR	sim_types.h
last_com	VECTOR	sim_types.h
delta	VECTOR	sim_types.h
heading	REAL	sim_types.h
num_vehicles	short	Standard
vehicles	pointer to VEH_ID_LIST	Sec. 2.8.2.2
next	pointer to struct veh_cluster	This struct
prev	pointer to struct veh_cluster	This struct

Table 2.8-148: VEH_CLUSTER Structure Definition

2.8.2.3 spotter.c CSC

/simnet/src/host/spotter.c

This CSC detects enemy vehicles and artillery fire in the area, via a spotter, so they can be reported to the user.

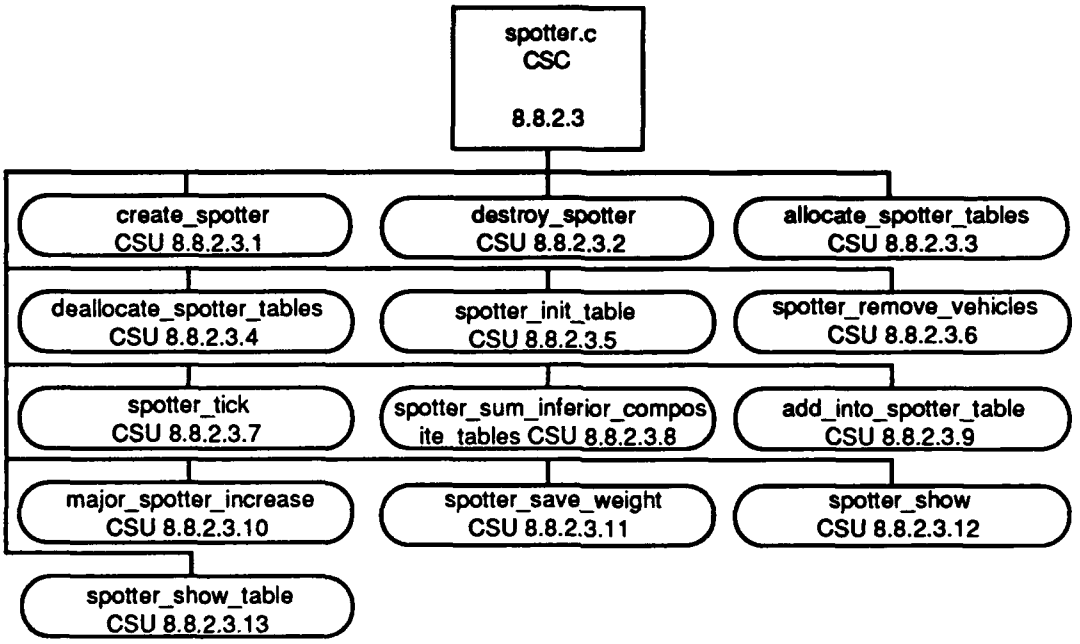


Figure 2.8-8: spotter.c CSC Structure

2.8.2.3.1 create_spotter CSU

This CSU allocates a block of memory for a spotter's SPOTTER_VARS structure and for that spotter's tables, sets spotter's time, sets the spotter's current index and generation count to zero, and returns a pointer to the spotter.

ReturnValues		
Return Value	**Type**	**Meaning**
s	pointer to SPOTTER_VARS	Created spotter.
Calls		
Function	**Where Described**	
allocate_spotter	Sec. 2.9.1.2 See Appendix A	
allocate_spotter_tables	Sec. 2.8.2.3.3	

Table 2.8-150: create_spotter CSU [8.8.2.3.1]

2.8.2.3.5 spotter_init_table CSU

This CSU initializes a spotter's vehicle table through a call to clear_bitfield.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
t	pointer to unsigned char	Standard
Calls		
Function	**Where Described**	
clear_bitfield	Sec. 2.14.3.4.6	

Table 2.8-154: spotter_init_table CSU [8.8.2.3.5]

2.8.2.3.6 spotter_remove_vehicles CSU

This CSU removes vehicles from both the spotter's current and old vehicle tables, based on the parameters passed, through calls to clear_bit.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
s	pointer to SPOTTER_VARS	Sec. 2.9.1.2 veh_storage.h
num	int	Standard
v_list[ ]	unsigned int	Standard
Calls		
Function	**Where Described**	
clear_bit	Sec. 2.14.3.4.4	

Table 2.8-155: spotter_remove_vehicles CSU [8.8.2.3.6]

2.8.2.3.7 spotter_tick CSU

This CSU is the primary interface to the spotter. At each tick, it checks whether detected vehicles from an inferior vehicles detection structure are actually visible. The visible vehicles are saved in a bitfield. If a vehicle is detected by one inferior vehicle and determined to be visible, it will not be checked for visibility by another inferior vehicle. Only one intervisibility operation is done at each tick.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
force	ForceID	basic.h

2.8.2.3.2 destroy_spotter CSU

This CSU destroys a spotter by deallocating the spotter table memory and spotter memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
s	pointer to SPOTTER_VARS	Sec. 2.9.1.2 veh_storage.h
Calls		
Function	**Where Described**	
deallocate_spotter_tables	Sec. 2.8.2.3.4	
deallocate_spotter	Sec. 2.9.1.2	

Table 2.8-151: destroy_spotter CSU [8.8.2.3.2]

2.8.2.3.3 allocate_spotter_tables CSU

This CSU obtains bitfield memory for a spotter's old and current vehicle tables and initializes the tables.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
spotter	pointer to SPOTTER_VARS	Sec. 2.9.1.2 veh_storage.h
Calls		
Function	**Where Described**	
allocate_bitfield	Sec. 2.14.3.4.1	
spotter_init_table	Sec. 2.8.2.3.5	

Table 2.8-152: allocate_spotter_tables CSU [8.8.2.3.3]

2.8.2.3.4 deallocate_spotter_tables CSU

This CSU deallocates the memory for the spotter's old and current vehicle tables.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
spotter	pointer to SPOTTER_VARS	Sec. 2.9.1.2 veh_storage.h
Calls		
Function	**Where Described**	
deallocate_bitfield	Sec. 2.14.3.4.2	

Table 2.8-153: deallocate_spotter_tables CSU [8.8.2.3.4]

Calls	
Function	**Where Described**
spotter_init_table	Sec. 2.8.2.3.5
spotter_sum_inferior_ composite_tables	Sec. 2.8.2.3.8
extract_bit	Sec. 2.14.3.9 See Appendix A
LOOKUP_VEHICLE	Sec. 2.9.3.2 See Appendix A
is_enemy	Sec. 2.13.3.1 See Appendix A
OBJ_FORCEID	Sec. 2.9.1.1 See Appendix A
OBJ_POSITION	Sec. 2.9.1.1 See Appendix A
is_dead	Sec. 2.13.3.1 See Appendix A
OBJ_APPEARANCE	Sec. 2.9.1.1 See Appendix A
OBJ_VEHICLECLASS	Sec. 2.9.1.1 See Appendix A
intervis_can_see_pt_to_pt	Sec. 2.6.5.2.1
set_bit	Sec. 2.14.3.4.3

Table 2.8-156: spotter_tick CSU [8.8.2.3.7]

2.8.2.3.8 spotter_sum_inferior_composite_tables CSU

This CSU adds spotter information from an inferior unit to that of a superior unit.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
composite	pointer to COMPOSITE_VARS	Sec. 2.9.1.2 veh_storage.h
t	pointer to unsigned char	Standard
Calls		
Function	**Where Described**	
add_into_spotter_table	Sec. 2.8.2.3.9	

Table 2.8-157: spotter_sum_inferior_composite_tables CSU [8.8.2.3.8]

2.8.2.3.9 add_into_spotter_table CSU

This CSU ORs together the information from the spotter tables.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
new_data	pointer to unsigned char	Standard
sum	pointer to unsigned char	Standard
Calls		
Function	**Where Described**	
or_bitfield	Sec. 2.14.3.4.8	

Table 2.8-158: add_into_spotter_table CSU [8.8.2.3.9]

At this point in the CSC, the spotter detection threshold (difference between new and old weights as a percentage) is declared for use within major_spotter_increase CSU.

Constant	Value
SPOTTER_DETECTION_THRESHOLD	10

Table 2.8-159: SPOTTER_DETECTION_THRESHOLD Constant Definition

2.8.2.3.10 major_spotter_increase CSU

This CSU checks whether the number of vehicles seen by the spotter has increased by at least 10% since the spotter table was last recalculated.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
FALSE	int	No increase in enemy weight, or no spotter.
TRUE	int	Enemy weight increased.
Calls		
Function	**Where Described**	
DEBUG_TARGETING	Sec. 2.5.2.2 See Appendix A	

Table 2.8-160: major_spotter_increase CSU [8.8.2.3.10]

2.8.2.3.11 spotter_save_weight CSU

This CSU checks if the SAF object is a composite with its own SPOTTER_VARS, a vehicle with a superior that has a SPOTTER_VARS, or a stranded unit with no spotter. In the former two cases, it saves the spotter's weight as the old weight.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1

Table 2.8-161: spotter_save_weight CSU [8.8.2.3.11]

2.8.2.3.12 spotter_show CSU

This CSU displays the spotter's current index and generation count along with the old vehicle table and the table currently being computed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
s	pointer to SPOTTER_VARS	Sec. 2.9.1.2 veh_storage.h
flags	int	Standard
Calls		
Function	**Where Described**	
spotter_show_table	Sec. 2.8.2.3.13	

Table 2.8-162: spotter_show CSU [8.8.2.3.12]

2.8.2.3.13 spotter_show_table CSU

This CSU displays a spotter's old or current vehicle table.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
t	pointer to unsigned char	Standard
Calls		
Function	**Where Described**	
extract_bit	Sec. 2.14.3.9 See Appendix A	

Table 2.8-163: spotter_show_table CSU [8.8.2.3.13]

2.8.2.4 spotter.h CSU

/simnet/src/host/spotter.h

This CSU contains the symbolic constants used by the spotter code.

Constant	Value
DETECTABLE /* Reults of detection */	0
NOT_DETECTABLE	1
NOT_POSSIBLY_VISIBLE	2
NOT_IN_RANGE	3
NOT_ENEMY	4
NOT_THERE	5
DETECTION_TIME_INTERVAL	1000
SPOTTER_TIME_INTERVAL	1000

Table 2.8-164: spotter.h Constant Definitions

2.8.3 Situation Assessment CSC

The Situation Assessment CSC consists of a single header CSU, tactical_state.h.

2.8.3.1 tactical_state.h CSU

/simnet/src/host/tactical_state.h

This CSU contains structures and symbolic constants used by the information reporting code. It is unused.

Constant	Value
ALL_QUIET	300000
ATTACK_REPORT_DEAD_TIME	60000
TOGETHER_DISTANCE2	25000000
MAX_VEH_TYPES	24

Table 2.8-165: tactical.h Constant Definitions

Item	Type	Where Type Defined
max_fuel	unsigned int	Standard
actual_fuel	unsigned int	Standard
nominal_vehicles	unsigned int	Standard
actual_vehicles	unsigned int	Standard
under_attack_p	unsigned int	Standard
vehicles[MAX_VEH_TYPES]	unsigned int	Standard
altitude	float	Standard

Table 2.8-166: SUB_STATE_REPORT Structure Definition

Item	Type	Where Type Defined
x	int	Standard
y	int	Standard
ammo	int	Standard
fuze	int	Standard
rounds	int	Standard

Table 2.8-167: SHELLING_DESC Structure Definition

2.9 SAF OBJECTS CSC

The SAF objects code, a collection of data structures and CSUs, forms the basis on which the executable objects in the SAF simulation are built. These objects are the local vehicles, remote vehicles, and units. The SAF object data structures allow for maintaining the position, SIMNET id, and appearance of all SAF data structures.

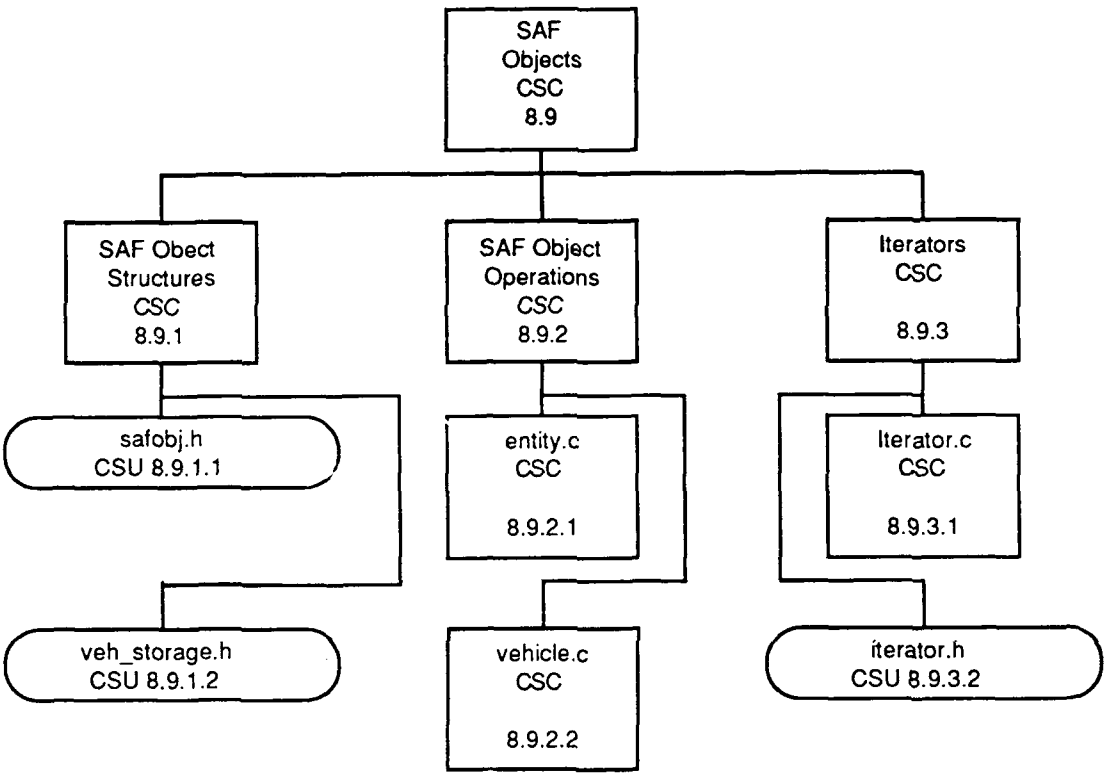


Figure 2.9-1: SAF Objects CSC Structure

2.9.1 SAF Object Structures CSC

The SAF Object Structures CSC consists of two header file CSUs, safobj.h and veh_storage.h. *These files contain the structures and macros for the SAF Object CSC.*

2.9.1.1 safobj.h CSU

/simnet/src/host/safobj.h

This CSU contains the SAF_OBJECT definition (typedef struct saf_object), which contains pointers to all of the other _VARS type structures (see Table 2.9-1). It also contains code to gather object information, and lookup macros which are contained in Appendix A.

Item	Type	Where Type Defined
saf_type	unsigned short int	-
tickable	pointer to TICKABLE_VARS	Sec. 2.9.1.2 veh_storage.h
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2 veh_storage.h
vehicle	pointer to VEHICLE_VARS	Sec. 2.9.1.2 veh_storage.h
remote	pointer to REMOTE_VARS	Sec. 2.9.1.2 veh_storage.h
damage	pointer to DAMAGE_VARS	Sec. 2.9.1.2 veh_storage.h
groundveh	pointer to GROUNDVEH_VARS	Sec. 2.9.1.2 veh_storage.h
turret	pointer to TURRET_VARS	Sec. 2.9.1.2 veh_storage.h
targeting	pointer to TARGETING_VARS	Sec. 2.9.1.2 veh_storage.h
weapon_systems	pointer to WEAPON_SYSTEMS_VARS	Sec. 2.9.1.2 veh_storage.h
driver	pointer to DRIVER_VARS	Sec. 2.9.1.2 veh_storage.h
logistics	pointer to LOGISTICS_VARS	Sec. 2.9.1.2 veh_storage.h
saf_vehicle	pointer to SAF_VEHICLE_VARS	Sec. 2.9.1.2 veh_storage.h
composite	pointer to COMPOSITE_VARS	Sec. 2.9.1.2 veh_storage.h
spotter	pointer to SPOTTER_VARS	Sec. 2.9.1.2 veh_storage.h
detection	pointer to DETECTION_VARS	Sec. 2.9.1.2 veh_storage.h
commander	pointer to COMMANDER_VARS	Sec. 2.9.1.2 veh_storage.h
navigator	pointer to NAVIGATOR_VARS	Sec. 2.9.1.2 veh_storage.h
collision	pointer to COLLISION_VARS	Sec. 2.9.1.2 veh_storage.h
reporter	pointer to REPORTER_VARS	Sec. 2.9.1.2 veh_storage.h
airveh	pointer to AIRVEH_VARS	Sec. 2.9.1.2 veh_storage.h
pilot	pointer to PILOT_VARS	Sec. 2.9.1.2 veh_storage.h

Table 2.9-1: SAF_OBJECT Structure

2.9.1.2 veh_storage.h CSU

/simnet/src/host/veh_storage.h

This CSU contains the structure definitions used by the entities modeled by this program. These structure definitions follow in tables along with two constant definitions. The CSU also contains the macros which allocate and deallocate each of the structures as needed. These are defined in Appendix A.

Item	Type	Where Type Defined
now	unsigned	Standard
time_since_new_tick	REAL	Sim_types.h
ep	int	Standard
periodic_msecs	unsigned int	Standard
tick_function	FUNC_PTR	

Table 2.9-2: TICKABLE_VARS Structure Definition

Item	Type	Where Type Defined
vehicleID	unsigned int	Standard
forceID	ForceID	basic.h
owner_port_number	int	Standard
sbx_uniq_id	unsigned char	Standard
top_unit_id	unsigned char	Standard
relative_id	unsigned char	Standard
tactics	unsigned char	Standard
icon	int	Standard
direction	VECTOR	sim_types.h
position	VECTOR	sim_types.h
velocity	VECTOR	sim_types.h
simple_rcvq	pointer to SIMPLE_QUEUE	Sec. 2.14.4.3

Table 2.9-3: ENTITY_VARS Structure Definition

Item	Type	Where Type Defined
vehicleClass	VehicleClass	
appearance	int	Standard
guises	VehicleGuises	basic.h
marking	VehicleMarking	basic.h
capabilities	VehicleCapabilities	basic.h
object_type	ObjectType	p_sim.h
network_event	short	Standard
turretAzimuth	Angle	basic.h
unElevation	Angle	basic.h
hull_to_world	T_MATRIX	sim-types.h
grid_entries	GRID_ENTRY_LIST	Sec. 2.9.3.2

Table 2.9-4: VEHICLE_VARS Structure Definition

Item	Type	Where Type Defined
last_appearance	pointer to SimulationPDU	p_sim.h
new_appearance_flag	unsigned short	Standard
last_update_time	unsigned int	Standard
normal_tick_rate	unsigned int	Standard
stealth	unsigned short	Standard
next_packet_to_sbx_time	unsigned int	Standard
controlling_port	int	Standard
being_watched : 1	unsigned int	Standard
on_road : 1	unsigned int	Standard
last_time_on_road	unsigned int	Standard
last_road_point	pointer to ROUTEPOINT	Sec. 2.10.2.5
last_road_index	int	Standard
last_watched_at	unsigned int	Standard

Table 2.9-5: REMOTE_VARS Structure Definition

Item	Type	Where Type Defined
df_damage_table	pointer to DATA_UNION	Sec. 2.1.1.5
if_damage_table	pointer to DATA_UNION	Sec. 2.1.1.5
collision_damage_prob	REAL	sim_types.h

Table 2.9-6: DAMAGE_VARS Structure Definition

Item	Type	Where Type Defined
hull_state	int	Standard
max_mps_fording	REAL	sim_types.h
max_mps_hardpack	REAL	sim_types.h
max_mps_softsoil	REAL	sim_types.h
max_mps_muck	REAL	sim_types.h
max_accel_fording	REAL	sim_types.h
max_accel_hardpack	REAL	sim_types.h
max_accel_softsoil	REAL	sim_types.h
max_accel_muck	REAL	sim_types.h
max_accel_braking	REAL	sim_types.h
skew_matrix	T_MATRIX	sim_types.h
last_soil_type	int	Standard

Table 2.9-7: GROUNDVEH_VARS Structure Definition

The following structure is tagged TURRET_VARS.

Item	Type	Where Type Defined
scan_state	int	Standard
turret_trashed	int	Standard
next_start_time	unsigned int	Standard
max_elevation	REAL	sim_types.h
min_elevation	REAL	sim_types.h
max_azimuth_rps	REAL	sim_types.h
max_elevate_rps	REAL	sim_types.h
scan_azimuth_rps	REAL	sim_types.h
scan_elevate_rps	REAL	sim_types.h
current_azimuth	REAL	sim_types.h
desired_azimuth	REAL	sim_types.h
azimuth_rps	REAL	sim_types.h
current_elevation	REAL	sim_types.h
desired_elevation	REAL	sim_types.h
elevation_rps	REAL	sim_types.h
scan_left_limit	REAL	sim_types.h
scan_right_limit	REAL	sim_types.h
hull_interest_dir	VECTOR	sim_types.h
hull_turret_offset	VECTOR	sim_types.h

Table 2.9-8: TURRET_VARS Structure Definition

The following structure is TARGETING_VARS.

Item	Type	Where Type Defined
firestatus	unsigned char	Standard
target	unsigned short	Standard
target_list[MAX_TARGETS]	unsigned short	Standard
position	VECTOR	sim_types.h
position_radius2	REAL	sim_types.h
position_valid	short	Standard
attack_role	unsigned char	Standard
view_type	unsigned char	Standard
max_vision_range2	REAL	sim_types.h
num_vision_grids	unsigned short	Standard
max_engagement_range2	REAL	sim_types.h
marksmanship	REAL	sim_types.h
next_tl_time	unsigned int	Standard
tl_recompute_time	unsigned int	Standard
tl_outdated	unsigned char	Standard
targeting_superior	unsigned short	Standard
chosen_weapon	unsigned char	Standard
selected_weapon	unsigned char	Standard
gunner_state	unsigned char	Standard
gunner_target	unsigned short	Standard
gunner_start_time	unsigned int	Standard
old_weight	int	Standard

Table 2.9-9: TARGETING_VARS Structure Definition

The following structure is tagged WEAPON_VARS.

Item	Type	Where Type Defined
state	int	Standard
start_time	unsigned int	Standard
attackerID	unsigned short	Standard
weapon_name	pointer to char	Standard
hitmodel	pointer to DATA_UNION	Sec. 2.1.1.5
round_type	ObjectType	p_sim.h
detonator	ObjectType	p_sim.h
jink_radius2	REAL	sim_types.h
min_range2	REAL	sim_types.h
max_range2	REAL	sim_types.h
muzzle_length	REAL	sim_types.h
muzzle_height	REAL	sim_types.h
muzzle_offset	REAL	sim_types.h
round_velocity	REAL	sim_types.h
mass	REAL	sim_types.h
directionality	REAL	sim_types.h
max_rounds	unsigned int	Standard
rounds_left	unsigned int	Standard
rounds_loaded	unsigned int	Standard
select_time	unsigned int	Standard
load_time	unsigned int	Standard
acquire_time	unsigned int	Standard
track_time	unsigned int	Standard
lose_sight_time	unsigned int	Standard
does_flyout	unsigned char	Standard
uses_turret	unsigned char	Standard
track_while_flying	unsigned char	Standard
radiate	unsigned char	Standard
radar_frequency	REAL	sim_types.h
radar_power	REAL	sim_types.h
antenna_gain	REAL	sim_types.h
pulse_energy	REAL	sim_types.h
last_radiate_time	unsigned int	Standard
burst_description	GUN_BURST_DESCRIPTION	
missile	pointer to MISSILE_VARS	Sec. 2.9.1.2

Table 2.9-10: WEAPON_VARS Structure Definition

The following structure is tagged WEAPON_PRIORITY_LIST.

Item	Type	Where Type Defined
count	int	Standard
list[MAX_WEAPONS]	unsigned char	Standard

Table 2.9-11: WEAPON_PRIORITY_LIST Structure Definition

The following structure is tagged WEAPON_SYSTEMS_VARS.

Item	Type	Where Type Defined
max_weapons	unsigned char	Standard
default_weapons	unsigned char	Standard
weapons[MAX_WEAPONS]	WEAPON_VARS	Sec. 2.9.1.2
tank_priority_list	WEAPON_PRIORITY_LIST	Sec. 2.9.1.2
air_priority_list	WEAPON_PRIORITY_LIST	Sec. 2.9.1.2
ground_priority_list	WEAPON_PRIORITY_LIST	Sec. 2.9.1.2

Table 2.9-12: WEAPON_SYSTEMS_VARS Structure Definition

Item	Type	Where Type Defined
immediate	int	Standard
imleadveh	pointer to struct saf_object	
imstationoffset	VECTOR	sim_types.h
imheading	VECTOR	sim_types.h
imdestpoint	VECTOR	sim_types.h
im_follow_on_road_point	pointer to ROUTEPOINT	Sec. 2.10.2.5
im_forp_dir	int	Standard
mission	int	Standard
leadveh	pointer to struct saf_object	Sec. 2.9.1.1
stationoffset	VECTOR	sim_types.h
follow_on_road_point	pointer to ROUTEPOINT	Sec. 2.10.2.5
forp_dir	int	Standard
direction	int	Standard
routedirection	int	Standard
maxerr2	REAL	sim_types.h
useimspeed	int	Standard
speed	REAL	sim_types.h
route	pointer to ROUTE	Sec. 2.10.2.5
routepoint	pointer to ROUTEPOINT	Sec. 2.10.2.5
debug	int	Standard
desired_speed	REAL	sim_types.h
desired_direction	VECTOR	sim_types.h

Table 2.9-13: DRIVER_VARS Structure Definition

Item	Type	Where Type Defined
range_remaining	REAL	sim_types.h
max_vehicle_range	REAL	sim_types.h
model_base_adjustment	REAL	sim_types.h
speed_mps	REAL	sim_types.h
max_speed_mps	REAL	sim_types.h
max_turns_rps	REAL	sim_types.h
vehicleStatus	int	Standard
superior	pointer to struct saf_object	Sec. 2.9.1.1
job	pointer to char	Standard
update_thresholds	pointer to DiscrepancyThresholds	libapp.h
last_update	pointer to SimulationPDU	p_sim.h
dimensions	VECTOR	sim_types.h
max_waypt_error	REAL	sim_types.h
max_waypt_error_squared	REAL	sim_types.h
desired_direction	VECTOR	sim_types.h
desired_speed	REAL	sim_types.h
last_position	VECTOR	sim_types.h
orderer	pointer to struct saf_object	
executing_mission	unsigned short	Standard
desired_velocity	VECTOR	sim_types.h
stealthy_mimic	unsigned int	Standard
max_times_between_apps	unsigned int	Standard

Table 2.9-14: SAF_VEHICLE_VARS Structure Definition

The following structure is tagged NAVIGATOR_VARS.

Item	Type	Where Type Defined
current_overlay	pointer to OVERLAY	Sec. 2.10.2.2
control_measures	pointer to CONTROL_MEASURE_LIST	Sec. 2.10.2.2
last_x	INT32	Sec. 2.12.1.18
last_y	INT32	Sec. 2.12.1.18

Table 2.9-15: NAVIGATOR_VARS Structure Definition

At this point in veh_storage.h, two constants are defined for use in the next structure.

Constant	Value
MAX_COMPOSITE_WIDTH	8
MAX_UNIT_SIZE	20

Table 2.9-16: veh_storage.h Constant Definitions

The following structure is tagged COMPOSITE_VARS.

Item	Type	Where Type Defined
superior	pointer to struct saf_object	Sec. 2.9.1.1
inferior_composites [MAX_COMPOSITE_WIDTH]	pointer to struct saf_object	Sec. 2.9.1.1
num_inferior_composites	unsigned short	Standard
inferior_vehicles [MAX_UNIT_SIZE]	pointer to struct saf_object	Sec. 2.9.1.1
num_inferior_vehicles	unsigned short	Standard
member_vehicles [MAX_UNIT_SIZE]	pointer to struct saf_object	Sec. 2.9.1.1
num_member_vehicles	unsigned short	Standard
echelon	pointer to char	Standard
echelon_type	pointer to char	Standard
echelon_as_int	unsigned short	Standard
echelon_type_as_int	unsigned short	Standard
formation_table	pointer to DATA_UNION	Sec. 2.1.1.5
leading_vehicle	pointer to struct saf_object	Sec. 2.9.1.1
prev_lead_veh	pointer to struct saf_object	Sec. 2.9.1.1
marking	VehicleMarking	basic.h
form_xoffset	int	Standard
formation	pointer to char	Standard
combat_instruction_set	pointer to CIS	
predicates	pointer to PREDICATES	
cis_table	pointer to DATA_UNION	Sec. 2.1.1.5
num_living_vehicles	unsigned short	Standard
below_fifty	unsigned char	Standard
below_twenty_five	unsigned char	Standard

Table 2.9-17: COMPOSITE_VARS Structure Definition

The following structure is tagged LOGISTICS_VARS.

Item	Type	Where Type Defined
status	unsigned char	Standard
logistics_state	unsigned char	Standard
full	unsigned char	Standard
canceled	unsigned char	Standard
current_loading	ObjectType	p_sim.h
current_load_qty	float	Standard
supplier_id	unsigned int	Standard
end_time	unsigned int	Standard
timeout_time	unsigned int	Standard
mun_offerQ	pointer to MunitionQuantity	basic.h

Table 2.9-18: LOGISTICS_VARS Structure Definition

The following structure is tagged SPOTTER_VARS.

Item	Type	Where Type Defined
last_spotter_time	unsigned int	Standard
current_index	unsigned short	Standard
generation_count	unsigned short	Standard
vehicle_table	pointer to unsigned char	Standard
old_vehicle_table	pointer to unsigned char	Standard
spotter_weight	int	Standard
old_spotter_weight	int	Standard

Table 2.9-19: SPOTTER_VARS Structure Definition

The following structure is tagged DETECTION_VARS.

Item	Type	Where Type Defined
last_detection_time	unsigned int	Standard
generation_count	unsigned int	Standard
detection_weight	int	Standard
old_detection_weight	int	Standard
detection_table[MAX_VEH]	char	Standard
detection_probs[MAX_VEH]	char	Standard

Table 2.9-20: DETECTION_VARS Structure Definition

The following structure is tagged COLLISION_VARS.

Item	Type	Where Type Defined
state	unsigned short	Standard
disabled	unsigned short	Standard
secs_until_collision	REAL	sim_types.h
vehicle_to_avoid	pointer to struct saf_object	Sec. 2.9.1.1
vehicle_to_avoids_velocity	VECTOR	sim_types.h
reaction_time	REAL	sim_types.h
vehicle_hit	int	Standard
vehicle_hits_position	VECTOR	sim_types.h
vehicle_hits_velocity	VECTOR	sim_types.h
ret_to_avoid	RECT	Sec. 2.14.3.9
last_x_checked	int	Standard
last_y_checked	int	Standard
avoidance_speed	REAL	sim_types.h

Table 2.9-21: COLLISION_VARS Structure Definition

Item	Type	Where Type Defined
type	int	Standard
crashing_state	int	Standard
landed	int	Standard
desired_velocity	VECTOR	sim_types.h
altitude_agl	REAL	sim_types.h
model_offset	REAL	sim_types.h
ground_level	REAL	sim_types.h
orientation	VECTOR	sim_types.h
desired_orientation	VECTOR	sim_types.h
hull_desired_velocity	VECTOR	sim_types.h
hull_current_velocity	VECTOR	sim_types.h
vehicle_mass	REAL	sim_types.h
engine_power	REAL	sim_types.h
power_factor	REAL	sim_types.h
max_pitch	REAL	sim_types.h
max_roll	REAL	sim_types.h
max_mps_forward	REAL	sim_types.h
min_mps_forward	REAL	sim_types.h
max_mps_landing	REAL	sim_types.h
max_mps_takeoff	REAL	sim_types.h
roll_damping	REAL	sim_types.h
pitch_damping	REAL	sim_types.h
dx2roll	REAL	sim_types.h
roll2dazimuth	REAL	sim_types.h
y_drag	REAL	sim_types.h
roll_ydrag	REAL	sim_types.h
xz_drag	REAL	sim_types.h
roll_gain	REAL	sim_types.h
power_gain	REAL	sim_types.h
pitch_gain	REAL	sim_types.h
lift	REAL	sim_types.h
pitch_inertia	REAL	sim_types.h
roll_inertia	REAL	sim_types.h
dz2pitch	REAL	sim_types.h
max_aircraft_power	REAL	sim_types.h
drag	VECTOR	sim_types.h
ss_drag	VECTOR	sim_types.h
force	VECTOR	sim_types.h
torque	VECTOR	sim_types.h
acceration	VECTOR	sim_types.h
angular_velocity	VECTOR	sim_types.h
angular_acceleration	VECTOR	sim_types.h

Table 2.9-23: AIRVEH_VARS Structure Definition

Item	Type	Where Type Defined
type	int	Standard
crashing_state	int	Standard
landed	int	Standard
desired_velocity	VECTOR	sim_types.h
altitude_agl	REAL	sim_types.h
model_offset	REAL	sim_types.h
ground_level	REAL	sim_types.h
orientation	VECTOR	sim_types.h
desired_orientation	VECTOR	sim_types.h
hull_desired_velocity	VECTOR	sim_types.h
hull_current_velocity	VECTOR	sim_types.h
vehicle_mass	REAL	sim_types.h
engine_power	REAL	sim_types.h
power_factor	REAL	sim_types.h
max_pitch	REAL	sim_types.h
max_roll	REAL	sim_types.h
max_mps_forward	REAL	sim_types.h
min_mps_forward	REAL	sim_types.h
max_mps_landing	REAL	sim_types.h
max_mps_takeoff	REAL	sim_types.h
roll_damping	REAL	sim_types.h
pitch_damping	REAL	sim_types.h
dx2roll	REAL	sim_types.h
roll2dazimuth	REAL	sim_types.h
y_drag	REAL	sim_types.h
roll_ydrag	REAL	sim_types.h
xz_drag	REAL	sim_types.h
roll_gain	REAL	sim_types.h
power_gain	REAL	sim_types.h
pitch_gain	REAL	sim_types.h
lift	REAL	sim_types.h
pitch_inertia	REAL	sim_types.h
roll_inertia	REAL	sim_types.h
dz2pitch	REAL	sim_types.h
max_aircraft_power	REAL	sim_types.h
drag	VECTOR	sim_types.h
ss_drag	VECTOR	sim_types.h
force	VECTOR	sim_types.h
torque	VECTOR	sim_types.h
acceration	VECTOR	sim_types.h
angular_velocity	VECTOR	sim_types.h
angular_acceleration	VECTOR	sim_types.h

Table 2.9-23: AIRVEH_VARS Structure Definition

Item	Type	Where Type Defined
type	int	Standard
im	AIR_SM /* state machines */	
mis	AIR_SM /* state machines */	
useimspeed	int /* override variables */	Standard
useimaltitude	int	Standard
useimcis	int	Standard
useimform	int	Standard
firetimeout	unsigned int /* fire at will */	Standard
runtimeout	unsigned int /* variables */	Standard
fireat_state	int	Standard
firespeed	REAL	sim_types.h
runspeed	REAL	sim_types.h
firealt	REAL	sim_types.h
runalt	REAL	sim_types.h
tfire	unsigned int	Standard
trun	unsigned int	Standard
tooclose2	REAL	sim_types.h
oldalt	REAL	sim_types.h
oldspeed	REAL	sim_types.h
target_point	VECTOR /* targeting */	sim_types.h
popup_point	VECTOR /* variables */	sim_types.h
setup_point	VECTOR	sim_types.h
targetid	unsigned short	Standard
dir_to_target	VECTOR	sim_types.h
target_range2	REAL	sim_types.h
valid_target	int	Standard
new_target	int	Standard
point_at_target_called	int	Standard
max_target_offset2	REAL	sim_types.h
max_cos_err2	REAL	sim_types.h
hoverattack_time	unsigned int	Standard
hoverattack_alt	REAL	sim_types.h
hoverattack_timeout	unsigned int	Standard
hoverattack_radius	REAL	sim_types.h
airport	VECTOR	sim_types.h
takeoff_altitude	REAL	sim_types.h
debug	int	Standard
desired_velocity	VECTOR	sim_types.h

Table 2.9-24: PILOT_VARS Structure Definition

The following structure is tagged COMMANDER_VARS.

Item	Type	Where Type Defined
state	unsigned short	Standard
has_mission : 1	unsigned int	Standard
mission_formation	pointer to char	Standard
override_formation	pointer to char	Standard
noted_leader	pointer to struct saf_object	Sec. 2.9.1.1
union		
{		
driver	DRIVER_VARS	Sec. 2.9.1.2
pilot	PILOT_VARS	Sec. 2.9.1.2
} leaders_brain		
overlay_name	pointer to char	Standard
route_name	pointer to char	Standard
mimic_vehicle	unsigned int	Standard

Table 2.9-25: COMMANDER_VARS Structure Definition

2.9.2 SAF Object Operations CSC

2.9.2.1 entity.c CSC

/simnet/src/host/entity.c

This CSC contains the code which creates and destroys entities. (An entity is a vehicle or a composite which is used to organize sets of vehicles. It is effectively any object which is not part of the terrain database, which is being handled by the Simhost program.) The CSC also contains any entity-specific CSUs.

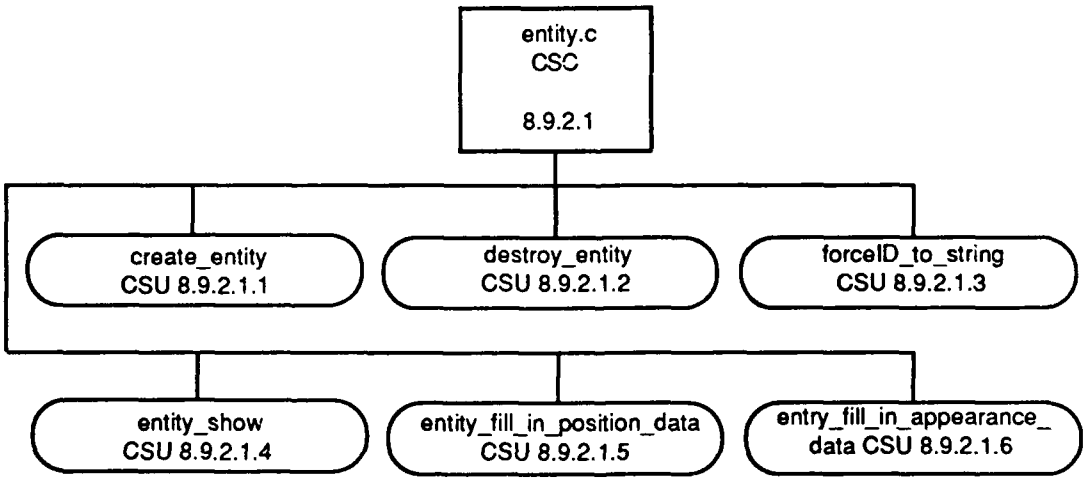


Figure 2.9-2: entity.c CSC Structure

2.9.2.1.1 create_entity CSU

This CSU creates an entity based on the characteristics passed in the parameters. An entity is any first-class object maintained by the phantom program. This includes opfor vehicles, missiles, composites and remote vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
id	unsigned int	Standard
forceID	ForceID	basic.h
heading	REAL	sim_types.h
position	pointer to REAL	sim_types.h
owner	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
icon	unsigned short	Standard
tactics	unsigned short	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
entity	pointer to ENTITY_VARS	Pointer to created entity
Calls		
Function	**Where Described**	
allocate_entity	Sec. 2.9.1.2	
vec_copy	Sec. 2.14.3.5.17	
tdb_get_gl	Sec. 2.14.1.2.2	
vec_init	Sec. 2.6.2.61.1 Vehicles CSCI SDD	
simple_queue_allocate	Sec. 2.14.2.1.7	

Table 2.9-26: create_entity CSU [8.9.2.1.1]

2.9.2.1.2 destroy_entity CSU

This CSU destroys an entity by flushing it from the queue, deallocating its queue and structure memory heaps.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
Calls		
Function	**Where Described**	
buffer_simple_flush	Sec. 2.14.4.1.16	
simple_queue_deallocate	Sec. 2.14.4.1.8	
deallocate_entity	Sec. 2.9.1.2 See Appendix A	

Table 2.9-27: destroy_entity CSU [8.9.2.1.2]

2.9.2.1.3 forceID_to_string CSU

This CSU provides a pointer to the character string for the provided force ID value.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
force	ForceID	basic.h
ReturnValues		
Return Value	**Type**	**Meaning**
"distinguished"	pointer to char	distinguishedForceID
"other"	pointer to char	otherForceID
"observer"	pointer to char	observerForceID
"target"	pointer to char	targetForceID
"unknown"	pointer to char	None of the above

Table 2.9-28: forceID_to_string CSU [8.9.2.1.3]

2.9.2.1.4 entity_show CSU

This CSU displays data on the entity passed as a parameter, including identification, location, speed and direction.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
flags	int	Standard
Calls		
Function	**Where Described**	
forceID_to_string	Sec. 2.9.2.1.3	
print_vector	Sec. 2.14.3.5.2	

Table 2.9-29: entity_show CSU [8.9.2.1.4]

2.9.2.1.5 entity_fill_in_position_data CSU

This CSU calculates and fills in the VEHICLE_POSITON_DESCRIPTOR position data from the position and direction information passed in via pointer to the ENTITY_VARS.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
desc_ptr	pointer to VEHICLE_ POSITION_DESCRIPTOR	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
s_atan2	Sec. 2.14.3.9 See Appendix A	

Table 2.9-30: entity_fill_in_position_data CSU [8.9.2.1.5]

2.9.2.1.6 entity_fill_in_appearance_data CSU

This CSU fills in the VEHICLE_APPEARANCE_DESCRIPTOR appearance data from the entity information passed in via the pointer to the ENTITY_VARS.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
desc_ptr	pointer to VEHICLE_APPEARANCE_DESCRIPTOR	Sec. 2.4.1.1

Table 2.9-31: entity_fill_in_appearance_data CSU [8.9.2.1.6]

2.9.2.2 vehicle.c CSC

/simnet/src/host/vehicle.c

This CSC handles creating, deleting, and filling in the information associated with all vehicles.

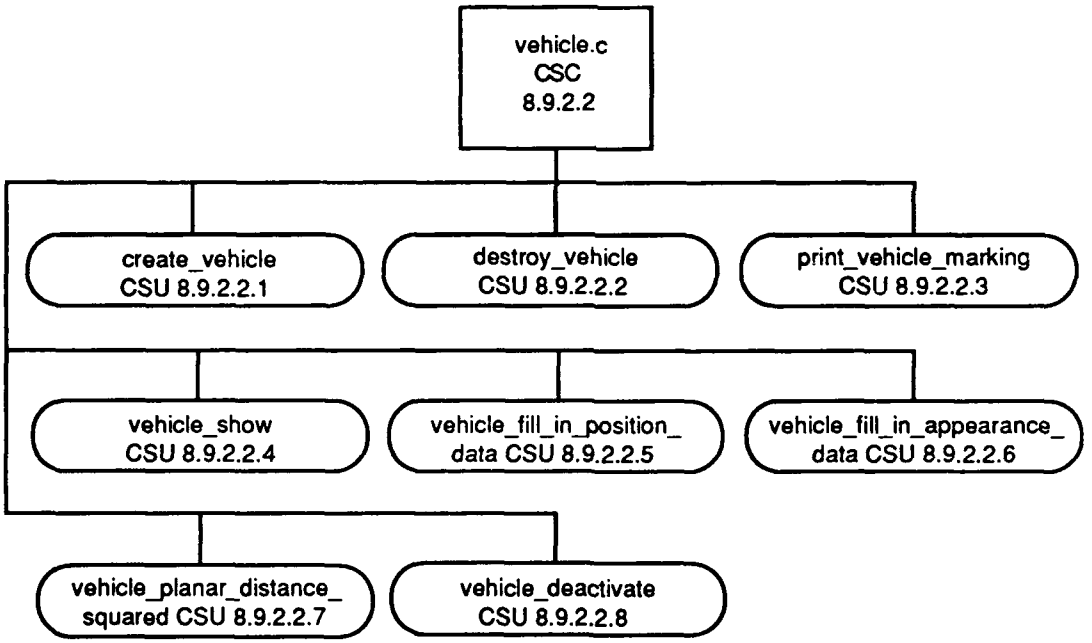


Figure 2.9-3: vehicle.c CSC Structure

2.9.2.2.1 create_vehicle CSU

This CSU creates a vehicle by obtaining memory for a VEHICLE_VARS structure, setting the class to vehicleClassSimple and initializing the other elements.

ReturnValues		
Return Value	**Type**	**Meaning**
vehicle	pointer to VEHICLE_VARS	Sec. 2.9.1.2
Calls		
Function	**Where Described**	
allocate_vehicle	Sec. 2.9.1.2	
mat_ident_init	Sec. 2.6.2.31.1 Vehicles CSCI SDD	

Table 2.9-32: create_vehicle CSU [8.9.2.2.1]

2.9.2.2.2 destroy_vehicle CSU

This CSU destroys the vehicle passed by calling deallocate_vehicle to release the VEHICLE_VARS structure's memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vehicle	pointer to VEHICLE_VARS	Sec. 2.9.1.2
Calls		
Function	**Where Described**	
deallocate_vehicle	Sec. 2.9.1.2	

Table 2.9-33: destroy_vehicle CSU [8.9.2.2.2]

2.9.2.2.3 print_vehicle_marking CSU

This CSU prints the marking of the vehicle passed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
marking	pointer to VehicleMarking	basic.h

Table 2.9-34: print_vehicle_marking CSU [8.9.2.2.3]

2.9.2.2.4 vehicle_show CSU

This CSU prints vehicle information including identification, type, appearance, conditions, and marking based on the passed ENTITY_VARS and VEHICLE_VARS structures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
vehicle	pointer to VEHICLE_VARS	Sec. 2.9.1.2
flags	int	Standard
Calls		
Function	**Where Described**	
simnet_id_string_from_saf_id		
print_vehicle_marking	Sec. 2.9.2.2.3	

Table 2.9-35: vehicle_show CSU [8.9.2.2.4]

2.9.2.2.5 vehicle_fill_in_position_data CSU

This CSU calculates and fills in the VEHICLE_POSITION_DESCRIPTOR structure's turret angle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vehicle	pointer to VEHICLE_VARS	Sec. 2.9.1.2
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
desc_ptr	pointer to VEHICLE_ POSITON_DESCRIPTOR	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
angle_clip	Sec. 2.14.3.5.7	
s_atan2	Sec. 2.14.3.9 See Appendix A	
simnet_angle_to_radians	libapp.h	

Table 2.9-36: vehicle_fill_in_position_data CSU [8.9.2.2.5]

2.9.2.2.6 vehicle_fill_in_appearance_data CSU

This CSU fills in the VEHICLE_APPEARANCE_DESCRIPTOR structure's appearance data.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vehicle	pointer to VEHICLE_VARS	Sec. 2.9.1.2
desc_ptr	pointer to VEHICLE_APPEARANCE_ DESCRIPTOR	Sec. 2.4.1.1

Table 2.9-37: vehicle_fill_in_appearance_data CSU [8.9.2.2.6]

2.9.2.2.7 vehicle_planar_distance_squared CSU

This CSU returns the square of the entity vehicle planar distance through a call to range_squared.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
location	pointer to REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
range_squared(location, entity->position)	REAL	Square of the vehicle planar distance
Calls		
Function	**Where Described**	
range_squared	Sec. 2.14.3.5.10	

Table 2.9-38: vehicle_planar_distance_squared CSU [8.9.2.2.7]

2.9.2.2.8 vehicle_deactivate CSU

This CSU deactivates a vehicle through a call to simnet_send_deactivate.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
Calls		
Function	**Where Described**	
simnet_send_deactivate		

Table 2.9-39: vehicle_deactivate CSU [8.9.2.2.8]

2.9.3 Iterators CSC

The Iterators CSC consists of a CSC, iterator.c, and a CSU, iterator.h.

2.9.3.1 iterator.c CSC

/simnet/src/host/iterator.c

This CSC maintains grid lists which help cut down on the amount of searching through vehicle tables done each time a vehicle tries to find out who it can see. It also maintains some basic information on vehicles to speed other things as well.

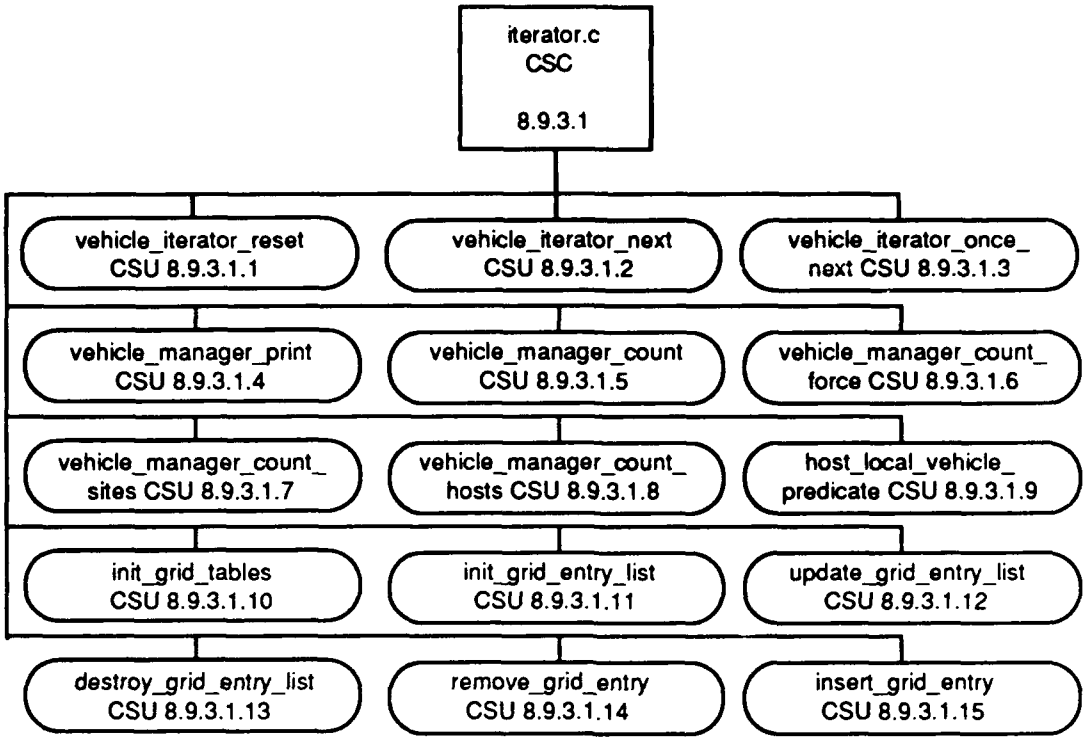


Figure 2.9-4: iterator.c CSC Structure

2.9.3.1.1 vehicle_iterator_reset CSU

This CSU resets the iterator passed by setting the next element (iterator->next) to zero and the match_type element to the type passed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
iterator	pointer to VEHICLE_ITERATOR	Sec. 2.9.3.2
type	unsigned short	Standard

Table 2.9-40: vehicle_iterator_reset CSU [8.9.3.1.1]

2.9.3.1.2 vehicle_iterator_next CSU

This CSU returns the next SAF object in the iterator list which is of the same vehicle type as that which was passed in. It stops searching if it has passed through the entire list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
iterator	pointer to VEHICLE_ITERATOR	Sec. 2.9.3.2

ReturnValues		
Return_Value	**Type**	**Meaning**
0	int	id found is the start id
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1

Calls	
Function	**Where Described**
LOOKUP_SAFOBJ	Sec. 2.9.1.1 See Appendix A
type_ok	Sec. 2.9.3.2 See Appendix A

Table 2.9-41: vehicle_iterator_next CSU [8.9.3.1.2]

2.9.3.1.3 vehicle_iterator_once_next CSU

This CSU returns the next SAF object in the iterator list which is of the same vehicle type as that which was passed in. It stops searching if it reaches an ID of MAX_VEH.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
iterator	pointer to VEHICLE_ITERATOR	Sec. 2.9.3.2

ReturnValues		
Return_Value	**Type**	**Meaning**
0	int	id found is the start id
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1

Calls	
Function	**Where Described**
LOOKUP_SAFOBJ	Sec. 2.9.1.1 See Appendix A
type_ok	Sec. 2.9.3.2 See Appendix A

Table 2.9-42: vehicle_iterator_once_next CSU [8.9.3.1.3]

2.9.3.1.4 vehicle_manager_print CSU

This CSU prints the id for all vehicles of the selected type, as well as the total number of such vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
type	int	Standard

Calls	
Function	**Where Described**
LOOKUP_SAFOBJ	Sec. 2.9.1.1 See Appendix A
type_ok	Sec. 2.9.3.2 See Appendix A

Table 2.9-43: vehicle_manager_print CSU [8.9.3.1.4]

2.9.3.1.5 vehicle_manager_count CSU

This CSU counts the number of vehicles of the type passed as the parameter, and returns the count.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
type	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
count	int	vehicle count
Calls		
Function	**Where Described**	
LOOKUP_SAFOBJ	Sec. 2.9.1.1 See Appendix A	
type_ok	Sec. 2.9.3.2 See Appendix A	

Table 2.9-44: vehicle_manager_count CSU [8.9.3.1.5]

2.9.3.1.6 vehicle_manager_count_force CSU

This CSU counts the number of force vehicles of the type passed as the parameter, and returns the count.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
force	ForceID	basic.h
ReturnValues		
Return Value	**Type**	**Meaning**
count	int	Number of force vehicles
Calls		
Function	**Where Described**	
LOOKUP_VEHICLE	Sec. 2.9.3.2 See Appendix A	
LOOKUP_FORCEID	Sec. 2.9.1.1 See Appendix A	

Table 2.9-45: vehicle_manager_count_force CSU [8.9.3.1.6]

2.9.3.1.7 vehicle_manager_count_sites CSU

This CSU updates the site table.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
site_table[ ]	short	Standard

Calls	
Function	**Where Described**
LOOKUP_VEHICLE	Sec. 2.9.3.2 See Appendix A
simnet_id_from_saf_id	

Table 2.9-46: vehicle_manager_count_sites CSU [8.9.3.1.7]

2.9.3.1.8 vehicle_manager_count_hosts CSU

This CSU updates the host table.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
site	int	Standard
host_table[ ]	short	Standard
Calls		
Function	**Where Described**	
LOOKUP_VEHICLE	Sec.2.9.3.2 See Appendix A	
simnet_id_from_saf_id		

Table 2.9-47: vehicle_manager_count_hosts CSU [8.9.3.1.8]

2.9.3.1.9 host_local_vehicle_predicate CSU

This CSU returns a value which indicates whether the Vehicle ID passed is a local vehicle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
id	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
safobj	pointer to SAF_OBJECT	
Calls		
Function	**Where Described**	
LOOKUP_SAFOBJ	Sec. 2.9.1.1 See Appendix A	
type_ok	Sec. 2.9.3.2 See Appendix A	

Table 2.9-48: host_local_vehicle_predicate CSU [8.9.3.1.9]

2.9.3.1.10 init_grid_tables CSU

This CSU calculates the grid table height, width and size for all grid tables, puts the data into three arrays for that grid, obtains the memory heap required for each grid table and clears the memory heap.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
terrain_x_size	int	Standard
terrain_y_size	int	Standard
Calls		
Function	**Where Described**	
TABLE_SIZE_FUNCTION	Sec. 2.9.3.2 See Appendix A	
heap_allocate	Sec. 2.14.2.1.1	

Table 2.9-49: init_grid_tables CSU [8.9.3.1.10]

2.9.3.1.11 init_grid_entry_list CSU

For each grid table in the grid entry list, this CSU updates the vehicle and the grid, and determines if the table applies to the function. If the table does apply to the function, insert_grid_entry is called, setting the previous and next entries in that table's GRID_ENTRY_LIST to NULL.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
gel	GRID_ENTRY_LIST	Sec. 2.9.3.2
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
TABLE_APPLIES_TO_FUNCTION	Sec. 2.9.3.2 See Appendix A	
insert_grid_entry	Sec. 2.9.3.1.15	

Table 2.9-50: init_grid_entry_list CSU [8.9.3.1.11]

2.9.3.1.12 update_grid_entry_list CSU

This CSU updates grid entries on the GRID_ENTRY_LIST passed based on x,y coordinates.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
gel	GRID_ENTRY_LIST	Sec. 2.9.3.2
x	int	Standard
y	int	Standard
Calls		
Function	**Where Described**	
coords_within_database	Sec. 2.13.3.1	
remove_grid_entry	Sec. 2.9.3.1.14	
insert_grid_entry	Sec. 2.9.3.1.15	

Table 2.9-51: update_grade_entry_list CSU [8.9.3.1.12]

2.9.3.1.13 destroy_grid_entry_list CSU

This CSU destroys the grid entry list passed, through calls to remove_grid_entry.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
gel	GRID_ENTRY_LIST	Sec. 2.9.3.2
Calls		
Function	**Where Described**	
remove_grid_entry()	Sec. 2.9.3.1.14	

Table 2.9-52: destroy_grid_entry_list CSU [8.9.3.1.13]

2.9.3.1.14 remove_grid_entry CSU

This CSU removes the grid entry passed from its grid entry list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
table	unsigned short	Standard
ge	pointer to GRID_ENTRY	Sec. 2.9.3.2

Table 2.9-53: remove_grid_entry CSU [8.9.3.1.14]

2.9.3.1.15 insert_grid_entry CSU

This CSU inserts the grid entry passed into its grid entry list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
table	unsigned short	Standard
grid	unsigned int	Standard
ge	pointer to GRID_ENTRY	Sec. 2.9.3.2

Table 2.9-54: insert_grid_entry CSU [8.9.3.1.15]

2.9.3.2 iterator.h CSU

/simnet/src/host/iterator.h

This CSU contains macros used in the iterator code as well as some binary templates used for comparisons. Constant and structure definitions within the header follow; macros are defined in Appendix A.

Constant	Value
LOCAL_VEHICLE	0x01
COMPOSITE	0x02
STEALTHY_LOCAL_VEHICLE	0x04
REMOTE_VEHICLE	0x08
MISSILE	0x10
INACTIVE_REMOTE_VEHICLE	0x20
VEHICLES	0x09
LOCALS	0x17
COMMANDABLES	0x07
ALL_OBJECTS	0xFF

Table 2.9-55: iterator.h Object Constant Definitions

Item	Type	Where Type Defined
next	unsigned short	Standard
match_type	unsigned short	Standard

Table 2.9-56: VEHICLE_ITERATOR Structure Definition

The following typedef struct is tagged GRID_ENTRY.

Item	Type	Where Type Defined
next	pointer to struct grid_entry	Sec. 2.9.3.2
prev	pointer to struct grid_entry	Sec. 2.9.3.2
vehicle	pointer to struct saf_object	Sec. 2.9.3.2
applies_to : 1	unsigned int	Standard
grid	int	Standard

Table 2.9-57: GRID_ENTRY Structure Definition

Constant	Value
NUM_GRID_TABLES	3
TABLE_200	0
TABLE_DIST_3500	1
TABLE_OTHER_3500	2

Table 2.9-58: Iterator.h Table Constant Definitions

Constant	Value
GRID_OFF_TDB	0xFFFFFFFF

Table 2.9-59: GRID_OFF_TDB Constant Definition

In addition, iterator.h also contains the following type definition:

typedef GRID_ENTRY GRID_ENTRY_LIST[NUM_GRID_TABLES]

and, among others, the following extern definition:

extern GRID_ENTRY ** grid_tables[ ]

They are required for the final two macros, DECLARE_FOR_VEHICLES_DO_VARS and FOR_VEHICLES_WITHIN_N_GRID_DO, which are defined in Appendix A.

2.10 OPORDERS CSC

The OPORDERS CSC [8.10] contains the data structures for the representation of graphical operations orders that are received from the workstation and the Combat Instruction Sets (CIS), which represent the behaviors a unit can simulate. The graphical operations orders are composed of routes and control measures to which CISs can be associated. This CSC also contains CSUs for determining when the unit reaches a control measure, and also for following routes.

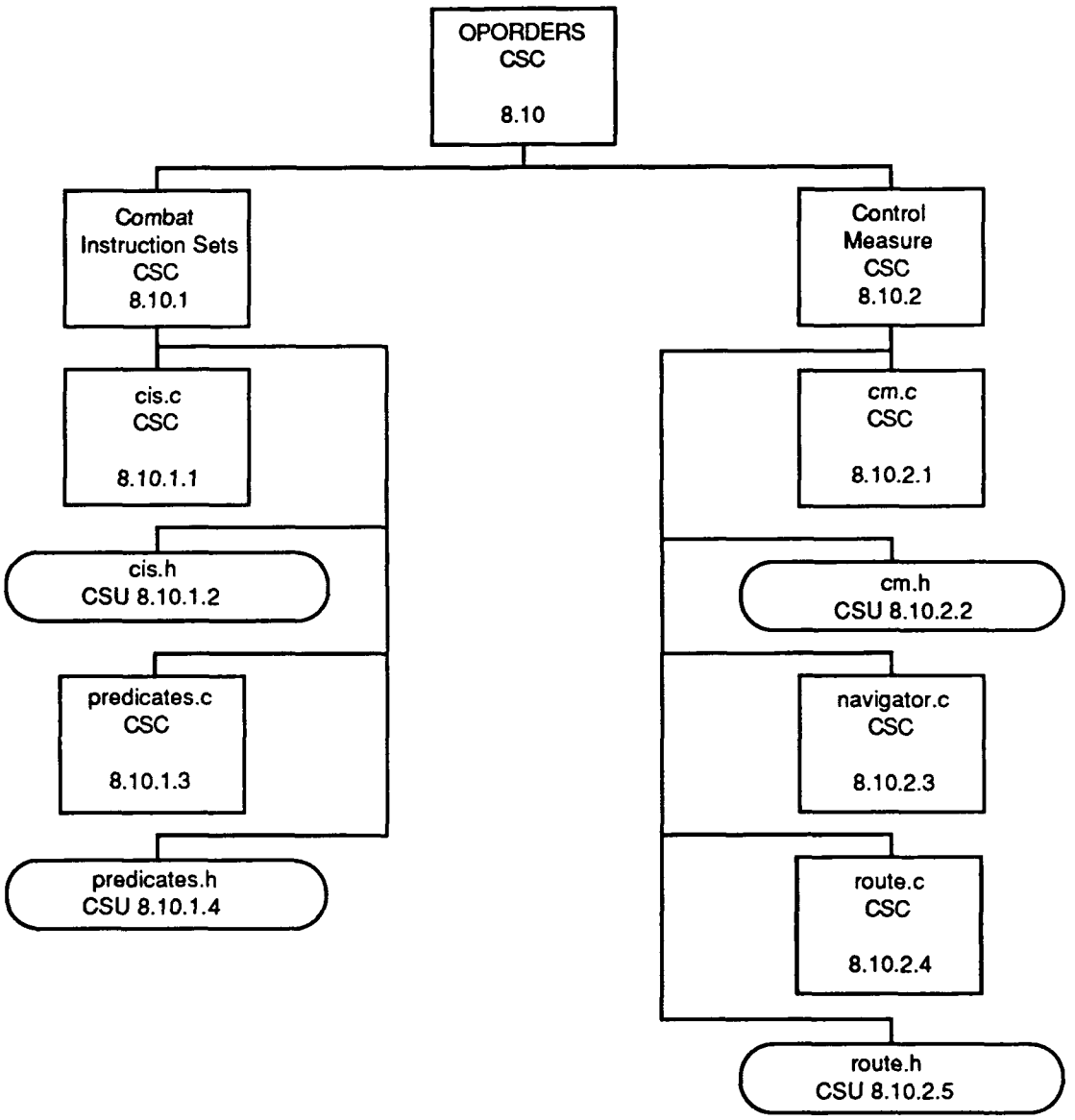


Figure 2.10-1: OPORDERS CSC Structure

2.10.1 Combat Instruction Sets CSC

The Combat Instruction Sets CSC [8.10.1] consists of four files: two CSCs, cis.c [8.10.1.1] and predicates.c [8.10.1.3], and their header file CSUs, cis.h [8.10.1.2] and predicates.h [8.10.1.4]. The cis.c file handles the reaction choices for a unit, while predicates.c tests whether a combat instruction set is needed to control the actions of an involved unit.

2.10.1.1 cis.c CSC

/simnet/src/host/cis.c

The combat instruction sets are the reaction choices available for a unit reacting to a control measure. They are loaded from a data file (see data files) and handled in the cis.c CSC.

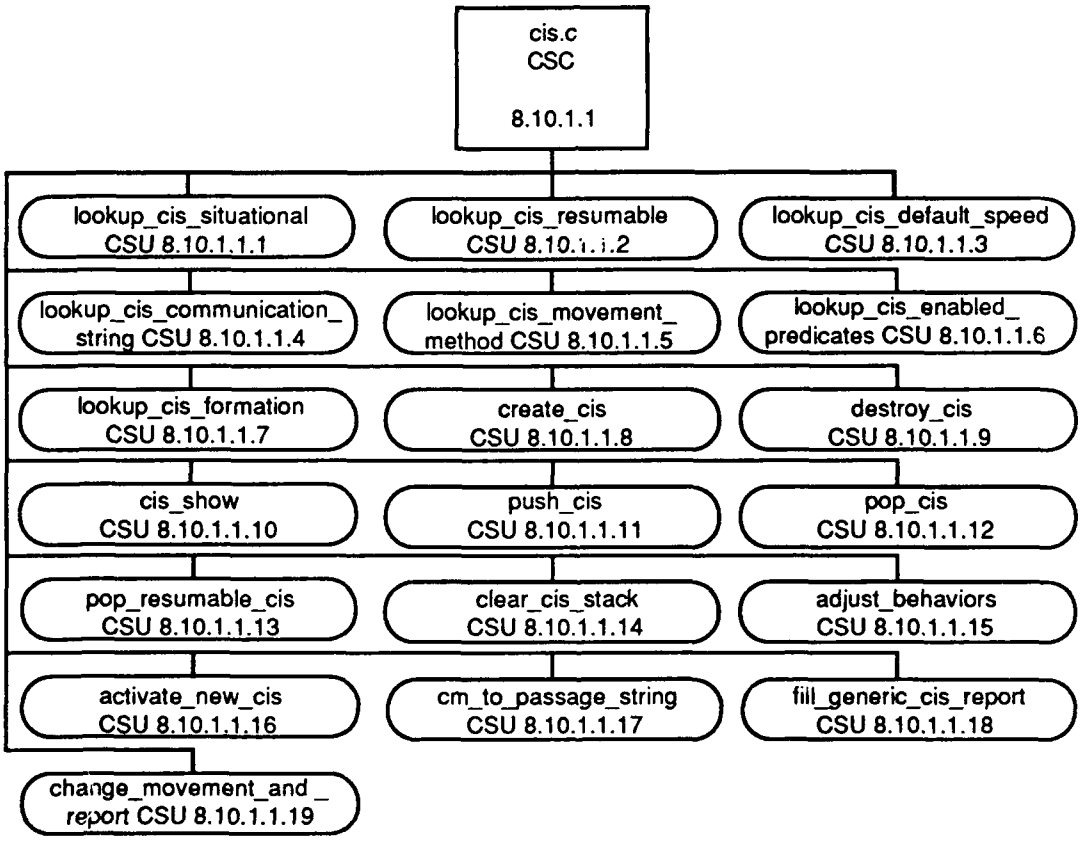


Figure 2.10-2: cis.c CSC Structure

2.10.1.1.1 lookup_cis_situational CSU

This CIS determines if a particular CIS is a situational CIS, and returns TRUE if it is.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cis_name	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Found
FALSE	int	Not found
Calls		
Function	**Where Described**	
ft_table	Sec. 2.14.1.2.14	
ft_int	Sec. 2.14.1.2.11	

Table 2.10-1: lookup_cis_situational CSU [8.10.1.1.1]

2.10.1.1.2 lookup_cis_resumable CSU

This CIS determines if a particular CIS is resumable, and returns TRUE if it is.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cis_name	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	CIS resumable.
FALSE	int	CIS not resumable.
Calls		
Function	**Where Described**	
ft_table	Sec. 2.14.1.2.14	
ft_int	Sec. 2.14.1.2.11	

Table 2.10-2: lookup_cis_resumable CSU [8.10.1.1.2]

2.10.1.1.3 lookup_cis_default_speed CSU

This CSU returns the default speed of a particular CIS.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cis_name	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
kph_to_mps(...)	REAL	Speed determined.
Calls		
Function	**Where Described**	
ft_table	Sec. 2.14.1.2.14	
kph_to_mps	Sec. 2.13.3.1	
ft_int	Sec. 2.14.1.2.11	

Table 2.10-3: lookup_cis_default_speed CSU [8.10.1.1.3]

2.10.1.1.4 lookup_cis_communication_string CSU

This CSU returns the string that should be sent to the workstation when a particular CIS is executed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cis_name	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
ft_symbol(...)	pointer to char	Normal end.
NULL	pointer to char	No CIS entry.
Calls		
Function	**Where Described**	
ft_table	Sec. 2.14.1.2.14	
ft_symbol	Sec. 2.14.1.2.13	

Table 2.10-4: lookup_cis_communication_string CSU [8.10.1.1.4]

2.10.1.1.5 lookup_cis_movement_method CSU

This CSU returns the appropriate movement method for a particular CIS.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cis_name	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
ft_symbol(...)	pointer to char	Normal end.
NULL	pointer to char	No CIS entry.
Calls		
Function	**Where Described**	
ft_table	Sec. 2.14.1.2.14	
ft_symbol	Sec. 2.14.1.2.13	

Table 2.10-5: lookup_cis_movement_method CSU [8.10.1.1.5]

2.10.1.1.6 lookup_cis_enabled_predicates CSU

This CSU returns the enabled predicates associated with a particular CIS.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cis_name	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
ft_symbol(...)	pointer to char	Normal end.
NULL	pointer to char	No CIS entry.
Calls		
Function	**Where Described**	
ft_table	Sec. 2.14.1.2.14	

Table 2.10-6: lookup_cis_enabled_predicates CSU [8.10.1.1.6]

2.10.1.1.7 lookup_cis_formation CSU

This CSU returns the appropriate formation associated with a particular CIS.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cis_name	pointer to char	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
ft_symbol(...)	pointer to char	Normal end.
UNKNOWN_SYM	pointer to char	No CIS entry.
Calls		
Function	**Where Described**	
ft_table	Sec. 2.14.1.2.14	
ft_symbol	Sec. 2.14.1.2.13	

Table 2.10-7: lookup_cis_formation CSU [8.10.1.1.7]

2.10.1.1.8 create_cis CSU

This CSU allocates memory for a CIS structure, sets the name to the one passed, sets stacked_cis to NULL, and returns a pointer to the set.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
name	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
set	pointer to CIS	Created CIS.
Calls		
Function	**Where Described**	
allocate_cis	Sec. 2.10.1.2 See Appendix A	

Table 2.10-8: create_cis CSU [8.10.1.1.8]

2.10.1.1.9 destroy_cis CSU

This CSU deallocates the CIS memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
set	pointer to CIS	Sec. 2.10.1.2
Calls		
Function	**Where Described**	
deallocate_cis	Sec. 2.10.1.2 See Appendix A	

Table 2.10-9: destroy_cis CSU [8.10.1.1.9]

2.10.1.1.10 cis_show CSU

This CSU displays all of the CISs on the stack, beginning with the current one.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
set	pointer to CIS	Sec. 2.10.1.2

Table 2.10-10: cis_show CSU [8.10.1.1.10]

2.10.1.1.11 push_cis CSU

This CSU pushes a new CIS onto the stack of a composite SAF object.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
set	pointer to CIS	Sec. 2.10.1.2
speed	REAL	sim_types.h
report	int	Standard
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
adjust_behaviors	Sec. 2.10.1.1.15	

Table 2.10-11: push_cis CSU [8.10.1.1.11]

2.10.1.1.12 pop_cis CSU

This CSU pops the CIS off the top of the CIS stack of a composite SAF object.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Errors		
Error Name	**Reason for Error**	
Error. ... cannot pop empty cis stack	Unable to set old_set equal to composite->combat_ instruction_set.	
Calls		
Function	**Where Described**	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
adjust_behaviors	Sec. 2.10.1.1.15	
destroy_cis	Sec. 2.10.1.1.9	

Table 2.10-12: pop_cis CSU [8.10.1.1.12]

2.10.1.1.13 pop_resumable_cis CSU

This CSU pops CISs off the top of the CIS stack until a CIS that is not resumable is found, or until the stack is empty.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
lookup_cis_resumable	Sec. 2.10.1.1.2	
destroy_cis	Sec. 2.10.1.1.9	
adjust_behaviors	Sec. 2.10.1.1.15	

Table 2.10-13: pop_resumable_cis CSU [8.10.1.1.13]

2.10.1.1.14 clear_cis_stack CSU

This CSU, for each of the CISs on the stack, calls destroy_cis to deallocate the set's memory. It then sets the composite's combat instruction set element to NULL.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
destroy_cis	Sec. 2.10.1.1.9	

Table 2.10-14: clear_cis_stack CSU [8.10.1.1.14]

2.10.1.1.15 adjust_behaviors CSU

This CSU makes the changes to a SAF composite object based on the behaviors associated with the CIS input.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
set	pointer to CIS	Sec. 2.10.1.2
speed	REAL	sim_types.h
report	int	Standard
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2

Calls	
Function	**Where Described**
commander_set_mission_ speed	Sec. 2.8.1.1.12
lookup_cis_information	Sec. 2.10.1.1.7
symbols_match	Sec. 2.1.1.5 See Appendix A
commander_set_mission_ formation	Sec. 2.8.1.1.15
change_movement_and_ report	Sec. 2.10.1.1.19
lookup_cis_movement_ method	Sec. 2.10.1.1.5
lookup_cis_enabled_ predicates	Sec. 2.10.1.1.6
reset_predicates	Sec. 2.10.1.3.10
enable_predicates	Sec. 2.10.1.3.12

Table 2.10-15: adjust_behaviors CSU [8.10.1.1.15]

2.10.1.1.16 activate_new_cis CSU

This CSU starts the execution of a CIS.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cis_name	pointer to char	Standard
new_speed	REAL	sim_types.h
report	int	Standard
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2

Calls	
Function	**Where Described**
buffer_allocate	Sec. 2.14.4.2.12
fill_generic_cis_report	Sec. 2.10.1.1.18
sbx_connection_send_to_ port	Sec. 2.4.3.2.19
buffer_deallocate	Sec. 2.14.4.2.15
create_cis	Sec. 2.10.1.1.8
lookup_cis_resumable	Sec. 2.10.1.1.2
destroy_cis	Sec. 2.10.1.1.9
push_cis	Sec. 2.10.1.1.11

Table 2.10-16: activate_new_cis CSU [8.10.1.1.16]

2.10.1.1.17 cm_to_passage_string CSU

This CSU returns, via a switch for the CM_TYPE_ token passed, a character string defining passage.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2
ReturnValues		
Return Value	**Type**	**Meaning**
"Passed"	pointer to char	CM_TYPE_POINT
"Crossed"	pointer to char	CM_TYPE_LINE
"Entered"	pointer to char	CM_TYPE_AREA or CM_TYPE_ZONE
"At"	pointer to char	Default, none of the above

Table 2.10-17: cm_to_passage_string CSU [8.10.1.1.17]

2.10.1.1.18 fill_generic_cis_report CSU

This CSU fills the appropriate information from a CIS into a CIS report, which is sent to the workstation.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
msg	pointer to GENERIC_MESSAGE_MSG	Sec. 2.4.1.1
ptr	pointer to char	Standard
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2
set	pointer to CIS	Sec. 2.10.1.2
speed_change	int	Standard
formation_change	int	Standard
Calls		
Function	**Where Described**	
fill_sbx_opfor_header	Sec. 2.4.3.2.18	
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A	
OBJ_POSITION	Sec. 2.9.1.1 See Appendix A	
tdb_giv_xy_get_utm	Sec. 2.21.7.24.3	
cm_to_passage_string	Sec. 2.10.1.1.17	
lookup_cis_communication_string	Sec. 2.10.1.1.4	

Table 2.10-18: fill_generic_cis_report CSU [8.10.1.1.18]

2.10.1.1.19 change_movement_and_report CSU

This CSU changes the movement method of a SAF composite object and sends a message to the workstation about the change.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
method	pointer to char	Standard
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2
report	int	Standard
set	pointer to CIS	Sec. 2.10.1.2
speed_change	int	Standard
formation_change	int	Standard

Calls	
Function	**Where Described**
buffer_allocate	Sec. 2.14.4.2.12
fill_generic_cis_report	Sec. 2.10.1.1.18
symbols_match	Sec. 2.1.1.5 See Appendix A
commander_set_mission_direction	Sec. 2.8.1.1.12
cm_get_center_of_mass	Sec. 2.10.2.3.15
tdb_giv_xy_get_utm	Sec. 2.21.7.24.3
commander_goto_point	Sec. 2.8.1.1.19
commander_halt	Sec. 2.8.1.1.11
intervis_get_high_ground	Sec. 2.6.5.2.5
OBJ_POSITION	Sec. 2.9.1.1 See Appendix A
lookup_cis_situational	Sec. 2.10.1.1.1
sbx_connection_send_to_port	Sec. 2.4.3.2.19
buffer_deallocate	Sec. 2.14.4.2.15

Table 2.10-19: change_movement_and_report CSU [8.10.1.1.19]

2.10.1.2 cis.h CSU

/simnet/src/host/cis.h

This file contains the macro definitions (defined in Appendix A), structure definition (CIS), and symbolic constants needed by the combat instruction set code.

The following typedef struct is tagged cis.

Item	Type	Where Type Defined
stacked_cis	pointer to struct cis	This struct
name	pointer to char	Standard

Table 2.10-20: CIS Structure Definition

Constant	Value
ENTERING_CIS	0
POPPING_CIS	1

Table 2.10-21: cis.h Constant Definitions

2.10.1.3 predicates.c CSC

/simnet/src/host/predicates.c

This CSC contains code to test whether certain combat instruction sets are needed to control the actions of the involved units.

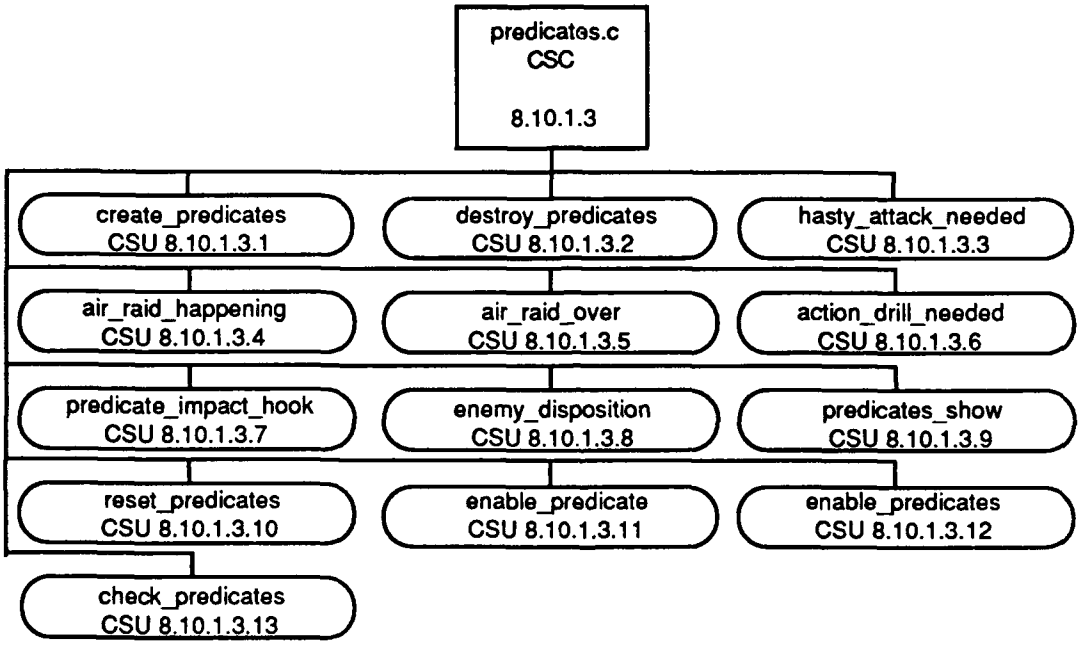


Figure 2.10-3: predictes.c CSC Structure

2.10.1.3.1 create_predicates CSU

This CSU allocates memory for a PREDICATES stucture and then calls CSU reset_predicates to initialize elements to FALSE.

ReturnValues		
Return Value	**Type**	**Meaning**
result	pointer to PREDICATES	Created predicates.
Calls		
Function	**Where Described**	
allocate_predicates	Sec. 2.10.1.4 See Appendix A	
reset_predicates	Sec. 2.10.1.3.10	

Table 2.10-22: create_predicates CSU [8.10.1.3.1]

2.10.1.3.2 destroy_predicates CSU

This CSU deallocates the PREDICATES structure memory through the CSU deallocate_predicate.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
p	pointer to PREDICATES	Sec. 2.10.1.4
Calls		
Function	**Where Described**	
deallocate_predicates	Sec. 2.10.1.4 See Appendix A	

Table 2.10-23: destroy_predicates CSU [8.10.1.3.2]

2.10.1.3.3 hasty_attack_needed CSU

This CSU checks if a there has been a round hit, and if the time since that hit is less than the HASTY_ATTACK_ROUND_TIME_THRESHOLD time. The CSU then calls cluster_from_vehicle to get the cluster. If the number of cluster vehicles is greater than or equal to the HASTY_ATTACK_VEHICLE_THRESHOLD and the position is within the HASTY_ATTACK_RANGE2_THRESHOLD, and the enemy disposition is one of the recognized cases, the predicates attacking force is set to the cluster and TRUE is returned; otherwise, the predicates attacking force is set to NULL and FALSE is returned.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Hasty attack needed.
FALSE	int	Hasty attack not needed.
Calls		
Function	**Where Described**	
OBJ_POSITION	Sec. 2.9.1.1 See Appendix A	
cluster_from_vehicle	Sec. 2.8.2.1.25	
vec2_range_squared	Sec. 2.14.3.5.30	
enemy_disposition	Sec. 2.10.1.3.8	

Table 2.10-24: hasty_attack_needed CSU [8.10.1.3.3]

2.10.1.3.4 air_raid_happening CSU

This CSU determines if an air hit was previous experienced, and if the time since the experience is less than the AIR_RAID_INTERVAL time. If so, it returns TRUE indictating that an air raid is happening; otherwise, it returns FALSE.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Air raid happening.
FALSE	int	Air raid not happening.

Table 2.10-25: air_raid_happening CSU [8.10.1.3.4]

2.10.1.3.5 air_raid_over CSU

This CSU determines if the time since an air hit was experienced is greater than the AIR_RAID_INTERVAL time. If so, it returns TRUE indicating that the air raid is over; otherwise, it returns FALSE.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Air raid is over.
FALSE	int	Air raid is not over.

Table 2.10-26: air_raid_over CSU [8.10.1.3.5]

2.10.1.3.6 action_drill_needed CSU

This CSU determines whether a SAF platoon should initiate an action drill by determining if the enemy is closing in or flanking.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Action drill needed.
FALSE	int	Action drill not needed.

Calls	
Function	**Where Described**
OBJ_POSITION	Sec. 2.9.1.1 See Appendix A
cluster_from_vehicle	Sec. 2.8.2.1.25
vec2_range_squared	Sec. 2.14.3.5.30
enemy_disposition	Sec. 2.10.1.3.8

Table 2.10-27: action_drill_needed CSU [8.10.1.3.6]

2.10.1.3.7 predicate_impact_hook CSU

This CSU communicates under-attack information to a SAF vehicle's superior.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
attackerID	unsigned int	Standard
Calls		
Function	**Where Described**	
LOOKUP_SAFOBJ	Sec. 2.9.1.1 See Appendix A	
IS_AIRCRAFT		
OBJ_OBJECT_TYPE	Sec. 2.9.1.1 See Appendix A	

Table 2.10-28: predicate_impact_hook CSU [8.10.1.3.7]

2.10.1.3.8 enemy_disposition CSU

This CSU calculates whether enemy vehicles are closing in or flanking on a SAF object.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
cluster	pointer to VEH_CLUSTER	Sec. 2.8.2.2
ReturnValues		
Return Value	**Type**	**Meaning**
ENEMY_AT_FLANK_OR_ REAR_STATIONARY	int	Enemy is at flank or rear and is not moving.
ENEMY_AT_FLANK_OR_ REAR_RETREATING	int	Enemy is at flank or rear and is retreating.
ENEMY_AT_FLANK_OR_ REAR_CLOSING_IN	int	Enemy is at flank or rear and is approaching.
ENEMY_AT_FRONT_ STATIONARY	int	Enemy is at front and not moving.
ENEMY_AT_FRONT_ CLOSING_IN	int	Enemy is at front and approaching.
ENEMY_AT_FRONT_ RETREATING	int	Enemy is at front and is retreating.
ENEMY_AT_UNKNOWN	int	There is no SAF vehicle.

Calls	
Function	**Where Described**
composite_distinguished_ member	Sec. 2.8.1.3.16
OBJ_POSITION	Sec. 2.9.1.1 See Appendix A
vec_copy	Sec. 2.6.2.53.1 Vehicles CSCI SDD
OBJ_DIRECTION	Sec. 2.9.1.1 See Appendix A
vec2_norm	Sec. 2.14.3.5.31
vec_sub	Sec. 2.6.2.65.1 Vehicles CSCI SDD
vec2_world2veh	Sec. 2.6.3.2.38
abs	Sec. 2.6.7.3 & Sec. 2.13.3.2 See Appendix A
vec2_dot	Sec. 2.14.3.5.24

Table 2.10-29: enemy_disposition CSU [8.10.1.3.8]

2.10.1.3.9 predicates_show CSU

This CSU displays predicates requirements and the last air hit, last round hit, and last round attacker.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
predicates	pointer to PREDICATES	Sec. 2.10.1.4

Table 2.10-30: predicates_show CSU [8.10.1.3.9]

2.10.1.3.10 reset_predicates CSU

This CSU resets the values in an enabled predicates to FALSE.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
predicates	pointer to PREDICATES	Sec. 2.10.1.4

Table 2.10-31: reset_predicates CSU [8.10.1.3.10]

2.10.1.3.11 enable_predicate CSU

This CSU, depending on the predicate passed, will reset the appropriate predicates variables and set the predicates event to TRUE; however, if the predicate is unknown, a message will be displayed that the SAF object (giving vehicle id number) is trying to enable an unkown predicate.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
predicate	pointer to char	Standard

Calls	
Function	**Where Described**
symbols_match	Sec. 2.1.1.5 See Appendix A
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A

Table 2.10-32: enable_predicate CSU [8.10.1.3.11]

2.10.1.3.12 enable_predicates CSU

This CSU calls enable_predicate for each of a number of predicates.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
predicates	pointer to DATA_UNION	Sec. 2.1.1.5
Calls		
Function	**Where Described**	
enable_predicate	Sec. 2.10.1.3.11	

Table 2.10-33: enable_predicates CSU [8.10.1.3.12]

2.10.1.3.13 check_predicates CSU

This CSU checks the predicates, and either creates, sets the speed for, and pushes a CIS onto the stack, or removes one.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
hasty_attack_needed	Sec. 2.10.1.3.3	
push_cis	Sec. 2.10.1.1.11	
create_cis	Sec. 2.10.1.1.8	
lookup_cis_default_speed	Sec. 2.10.1.1.3	
air_raid_happening	Sec. 2.10.1.3.4	
air_raid_over	Sec. 2.10.1.3.5	
pop_cis	Sec. 2.10.1.1.12	
action_drill_needed	Sec. 2.10.1.3.6	

Table 2.10-34: check_predicates CSU [8.10.1.3.13]

2.10.1.4 predicates.h CSU

/simnet/src/host/predicates.h

This CSU contains the structure definition, macro definitions (defined in Appendix A) and symbolic constants used by the predicate code. These constants specify threshold requirements used when performing the tests in predicates.c.

The following typedef struct is tagged predicates. The first four items are predicate flags and the last four items are data needed to support the predicates.

Item	Type	Where Type Defined
hasty_attack_needed : 1	unsigned int	Standard
air_raid_happening : 1	unsigned int	Standard
air_raid_over : 1	unsigned int	Standard
action_drill_needed : 1	unsigned int	Standard
last_air_hit	unsigned int	Standard
last_round_hit	unsigned int	Standard
last_round_attacker	unsigned int	Standard
attacking_force	pointer to VEH_CLUSTER	Sec. 2.8.2.2

Table 2.10-35: PREDICATES Structure Definition

Constant	Value
MOVING_IN_THRESHOLD2	(51.0 * 51.0)
AIR_RAID_INTERVAL	(3 * 60 * 1000)
HASTY_ATTACK_ROUND_TIME_THRESHOLD	(3 * 60 * 1000)
HASTY_ATTACK_VEHICLE_THRESHOLD	3
HASTY_ATTACK_RANGE2_THRESHOLD	(2000.0 * 2000.0)
ACTION_DRILL_ROUND_TIME_THRESHOLD	(3 * 60 * 1000)
ACTION_DRILL_VEHICLE_THRESHOLD	2
ACTION_DRILL_RANGE2_THRESHOLD	(2000.0 * 2000.0)
ENEMY_AT_FLANK_OR_REAR_STATIONARY	0
ENEMY_AT_FLANK_OR_REAR_RETREATING	1
ENEMY_AT_FLANK_OR_REAR_CLOSING_IN	2
ENEMY_AT_FRONT_STATIONARY	3
ENEMY_AT_FRONT_RETREATING	4
ENEMY_AT_FRONT_CLOSING_IN	5
ENEMY_AT_UNKNOWN	6

Table 2.10-36: predicates.h Constant Definitions

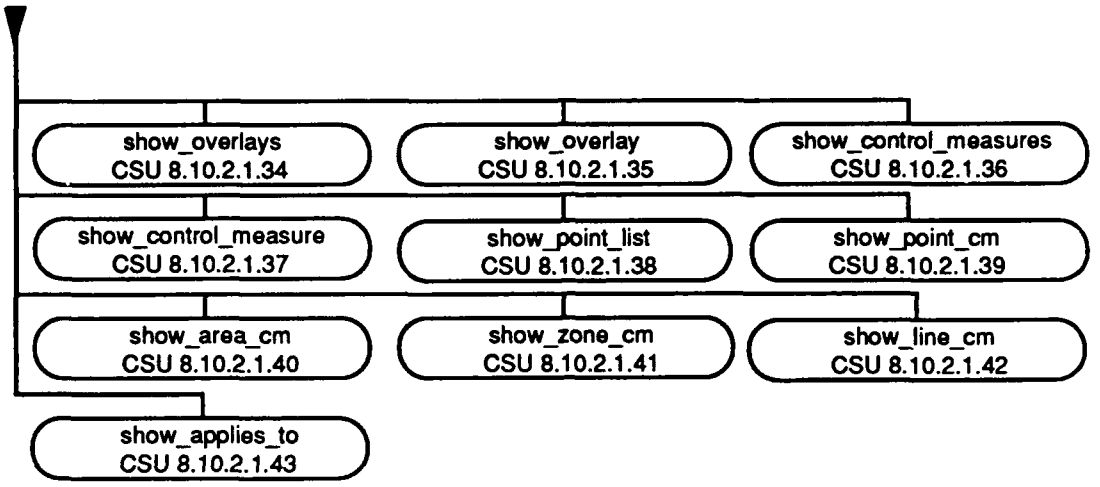


Figure 2.10-5: cm.c CSC Structure Part 2 of 2

2.10.2.1.1 copy_float_point_list_to_real_point_list CSU

This CSU copies a floating point list of points into a real point list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
float_point_list	pointer to FLOAT_CM_POINT_LIST	Sec. 2.10.2.2
real_point_list	pointer to CM_POINT_LIST	Sec. 2.10.2.2

Table 2.10-37: copy_float_point_list_to_real_point_list CSU [8.10.2.1.1]

2.10.2.1.2 get_overlay CSU

This CSU obtains an overlay from the SBX, creating it if necessary by allocating memory, initializing values, making the associated list item, splicing it into the front of the existing list, and returning the new overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
overlay_id	int	standard
overlay_name	pointer to char	standard
sbx_connection	pointer to SBX_CONNECTION_vars	Sec. 2.4.3.3
ReturnValues		
Return Value	**Type**	**Meaning**
overlay	pointer to OVERLAY	Overlay fetched or created

2.10.2 Control Measure CSC

2.10.2.1 cm.c CSC

/simnet/src/host/cm.c

This CSC takes the control measure information passed from the workstation and converts it into a form usable by the simulation host for the composites and vehicles it is simulating.

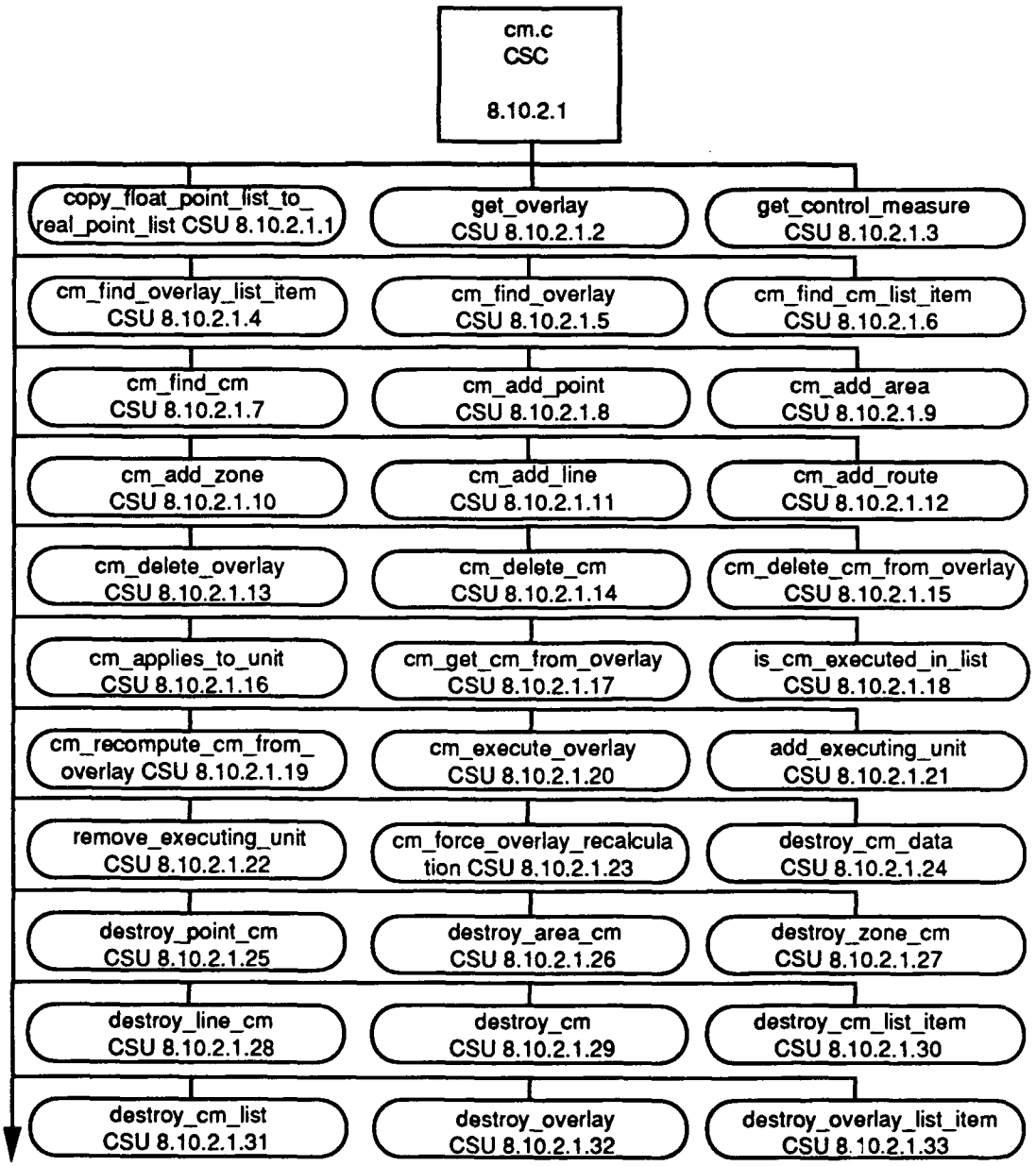


Figure 2.10-4: cm.c CSC Structure Part 1 of 2

Calls	
Function	**Where Described**
DEBUG_CM	Sec. 2.5.2.2
allocate_overlay	Sec. 2.10.2.2
allocate_overlay_list_item	Sec. 2.10.2.2

Table 2.10-38: get_overlay CSU [8.10.2.1.2]

2.10.2.1.3 get_control_measure CSU

This CSU obtains a control measure from an overlay. If the control measure does not exist, the CSU creates it and its, data based upon type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
id	int	Standard
name	pointer to char	Standard
type	int	Standard
overlay	pointer to OVERLAY	Sec. 2.10.2.2
num_units	short	Standard
applies_to[ ]	short	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
NULL	int	Not created
cm	int	The new control measure

Calls	
Function	**Where Described**
DEBUG_CM	Sec. 2.5.2.2
allocate_control_measure	Sec. 2.10.2.2
allocate_point_cm	Sec. 2.10.2.2
allocate_area_cm	Sec. 2..10.2.
allocate_zone_cm	Sec. 2.10.2.2
allocate_line_cm	Sec. 2.10.2.2
allocate_control_measure_list_item	Sec. 2.10.2.2

Table 2.10-39: get_control_measure CSU [8.10.2.1.3]

2.10.2.1.4 cm_find_overlay_list_item CSU

This CSU finds an overlay list item (id) if it is in the overlay lists.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
overlay_id	int	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
overlay_list	pointer to OVERLAY_LIST	Pointer to list
NULL	pointer to OVERLAY_LIST	Not found

Table 2.10-40: cm_find_overlay_list_item CSU [8.10.2.1.4]

2.10.2.1.5 cm_find_overlay CSU

This CSU finds an overlay through a call to cm_find_overlay_last_item returning a pointer to the overlay if one exists.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
overlay_id	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
result->overlay	pointer to OVERLAY	Overlay found
NULL	pointer to OVERLAY	Overlay not there
Calls		
Function	**Where Described**	
cm_find_overlay_list_item	Sec. 2.10.2.1.6	

Table 2.10-41: cm_find_overlay CSU [8.10.2.1.5]

2.10.2.1.6 cm_find_cm_list_item CSU

This CSU finds a control measure list if it exists.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm_id	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
cm_list	pointer to COUNTER_MEASURE_LIST	List found
NULL	pointer to COUNTER_MEASURE_LIST	Not there

Table 2.10-42: cm_find_cm_list_item CSU [8.10.2.1.6]

2.10.2.1.7 cm_find_cm CSU

This CSU finds a control measure, returning a pointer to it, if one exists.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm_id	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
result->cm	pointer to CONTROL_MEASURE_LIST	This is the control measure
NULL	pointer to CONTROL_MEASURE_LIST	Not there
Calls		
Function	**Where Described**	
cm_find_cm_list_item	Sec. 2.10.2.1.6	

Table 2.10-43: cm_find_cm CSU [8.10.2.1.7]

2.10.2.1.8 cm_add_point CSU

This CSU adds a point control measure from the SBX connection to an overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx_connection	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
msg_ptr	pointer to POINT_MSG	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
get_overlay	Sec. 2.10.2.1.2	
get_control_measure	Sec. 2.10.2.1.3	
DEBUG_CM	Sec. 2.10.2.2	
get_symbol	Sec. 2.1.1.3.2	
cm_force_overlay_recalculation	Sec. 2.10.2.1.23	

Table 2.10-44: cm_add_point CSU [8.10.2.1.8]

2.10.2.1.9 cm_add_area CSU

This CSU adds an area control measure from the SBX connection to an overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx_connection	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
msg_ptr	pointer to AREA_MSG	2.4.1.1
Calls		
Function	**Where Described**	
get_overlay	Sec. 2.10.2.1.2	
get_control_measure	Sec. 2.10.2.1.3	
DEBUG_CM	Sec. 2.10.2.2	
copy_float_point_list_to_ real_point_list	Sec. 2.10.2.1.1	
get_symbol	Sec. 2.1.1.3.2	
cm_force_overlay_ recalculation	Sec. 2.10.2.1.23	

Table 2.10-45: cm_add_area CSU [8.10.2.1.9]

2.10.2.1.10 cm_add_zone CSU

This CSU adds a zone control measure from the SBX connection to an overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx_connection	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
msg_ptr	pointer to ZONE_MSG	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
get_overlay	Sec. 2.10.2.1.2	
get_control_measure	Sec. 2.10.2.1.3	
DEBUG_CM	Sec. 2.10.2.2	
copy_float_point_list_to_ real_point_list	Sec. 2.10.2.1.1	
get_symbol	Sec. 2.1.1.3.2	
cm_force_overlay_ recalculation	Sec. 2.10.2.1.23	

Table 2.10-46: cm_add_zone CSU [8.10.2.1.10]

2.10.2.1.11 cm_add_line CSU

This CSU adds a line control measure from the SBX connection to an overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx_connection	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
msg_ptr	pointer to LINE_MSG	2.4.1.1
Calls		
Function	**Where Described**	
get_overlay	Sec. 2.10.2.1.2	
get_control_measure	Sec. 2.10.2.1.3	
DEBUG_CM	Sec. 2.10.2.2	
copy_float_point_list_to_real_point_list	Sec. 2.10.2.1.1	
get_symbol	Sec. 2.1.1.3.2	
cm_force_overlay_recalculation	Sec. 2.10.2.1.23	

Table 2.10-47: cm_add_line CSU [8.10.2.1.11]

2.10.2.1.12 cm_add_route CSU

This CSU adds a route control measure from the SBX connection to an overlay. If the route is already in use, the vehicles using the route are updated to use the new route.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx_connection	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
msg_ptr	pointer to ROUTE_MSG	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
get_overlay	Sec. 2.10.2.1.2	
get_control_measure	Sec. 2.10.2.1.3	
DEBUG_CM	Sec. 2.10.2.2	
generate_route_from_route_msg	Sec. 2.10.2.4.7	
FOR_VEHICLES_DO	Sec. 2.9.3.2	
p_follower_comp_reset_route	Sec. 2.6.4.3.4	
re_assign_routepoints	Sec. 2.10.2.4.13	
vec2_copy	Sec. 2.14.3.5.29	
OBJ_POSITION	Sec. 2.9.1.1	
pilot_start_gotopoint	Sec. 2.6.4.2.26	
destroy_route	Sec. 2.10.2.4.4	

Table 2.10-48: cm_add_route CSU [8.10.2.1.12]

2.10.2.1.13 cm_delete_overlay CSU

This CSU deletes control measures from the SBX connection.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx_connection	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
overlay_id	int	Standard
Calls		
Function	**Where Described**	
cm_find_overlay_list_item	Sec. 2.10.2.1.4	
DEBUG_CM	Sec. 2.10.2.2	
cm_delete_cm_from_overlay	Sec. 2.10.2.1.15	
destroy_overlay	Sec. 2.10.2.1.32	
destroy_overlay_list_item	Sec. 2.10.2.1.33	

Table 2.10-49: cm_delete_overlay CSU [8.10.2.1.13]

2.10.2.1.14 cm_delete_cm CSU

This CSU deletes a control measure from an overlay, and updates any vehicles using that overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx_connection	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
msg_ptr	pointer to DELETE_CM_MSG	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
cm_find_overlay	Sec. 2.10.2.1.5	
cm_delete_cm_from_overlay	Sec. 2.10.2.1.15	
cm_force_overlay_ recalculation	Sec. 2.10.2.1.23	

Table 2.10-50: cm_delete_cm CSU [8.10.2.1.14]

2.10.2.1.15 cm_delete_cm_from_overlay CSU

This CSU deletes a control measure from an overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm_id	int	Standard

Calls	
Function	**Where Described**
cm_find_cm_list_item	Sec. 2.10.2.1.6
DEBUG_CM	Sec. 2.10.2.2
fragment_route	Sec. 2.10.2.4.12
destroy_cm_data	Sec. 2.10.2.1.24
destroy_cm	Sec. 2.10.2.1.29
destroy_cm_list_item	Sec. 2.10.2.1.30

Table 2.10-51: cm_delete_cm_from_overlay CSU [8.10.2.1.15]

2.10.2.1.16 cm_applies_to_units CSU

This CSU determines which units are using a particular overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
num_units	unsigned short	Standard
applies_to[ ]	unsigned short	Standard
id	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	using overlay
FALSE	int	not using overlay

Table 2.10-52: cm_applies_to_units CSU [8.10.2.1.16]

2.10.2.1.17 cm_get_cm_from_overlay CSU

This CSU returns all control measures currently in use by a SAF unit.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
id	unsigned int	Standard
overlay	pointer to OVERLAY	Sec. 2.10.2.2
ReturnValues		
Return Value	**Type**	**Meaning**
result	pointer to CONTROL_MEASURE_LIST	control measures in use by SAF unit
Calls		
Function	**Where Described**	
cm_applies_to_unit	Sec. 2.10.2.1.16	

Table 2.10-53: cm_get_cm_from_overlay CSU [8.10.2.1.17]

2.10.2.1.18 is_cm_executed_in_list CSU

This CSU returns a list of control measures which have been executed by a particular SAF unit.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
id	int	Standard
list	pointer to CONTROL_MEASURE_LIST	Sec. 2.10.2.2
ReturnValues		
Return Value	**Type**	**Meaning**
list->executed	int	CM executed
FALSE	int	CM executed not found

Table 2.10-54: is_cm_executed_in_list CSU [8.10.2.1.18]

2.10.2.1.19 cm_recompute_cm_from_overlay CSU

This CSU remakes the control measure list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
id	unsigned int	Standard
overlay	pointer to OVERLAY	Sec. 2.10.2.2
old_cm	pointer to CONTROL_MEASURE_LIST	Sec. 2.10.2.2
ReturnValues		
Return Value	**Type**	**Meaning**
result	pointer to CONTROL_MEASURE_LIST	control measure list
Calls		
Function	**Where Described**	
cm_applies_to_unit	Sec. 2.10.2.1.16	
allocate_control_measure_list_item	Sec. 2.10.2.2	
is_cm_executed_in_list	Sec. 2.10.2.1.18	

Table 2.10-55: cm_recompute_cm_from_overlay CSU [8.10.2.1.19]

2.10.2.1.20 cm_execute_overlay CSU

This CSU executes an overlay from the SBX connection.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
sbx_connection	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
msg_ptr	pointer to EXECUTE_OVERLAY_MSG	Sec. 2.4.1.1
Calls		
Function	**Where Described**	
LOOKUP_VEHICLE	Sec. 2.9.1.1	
DEBUG_CM	Sec. 2.5.2.2	
OBJ_OWNER_PORT_ NUMBER	Sec. 2.9.1.1	
cm_find_overlay	Sec. 2.10.2.1.5	
allocate_control_measure	Sec. 2.10.2.2	
generate_inplace_route	Sec. 2.10.2.4.8	
cm_find_cm	Sec. 2.10.2.1.7	
remove_executing_unit	Sec. 2.10.2.1.22	
add_executing_unit	Sec. 2.10.2.1.21	
clear_cis_stack	Sec. 2.10.1.1.14	
activate_new_cis	Sec. 2.10.1.1.16	
get_symbol	Sec. 2.1.1.3.2	
execute_overlay	Sec. 2.14.1.1.8	

Table 2.10-56: cm_execute_overlay CSU [8.10.2.1.20]

2.10.2.1.21 add_executing_unit CSU

This CSU adds a unit to an overlay when that unit starts executing the overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
unitid	unsigned int	Standard
overlay	pointer to OVERLAY	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
allocate_veh_id_list	Sec. 2.8.2.2	

Table 2.10-57: add_executing_unit CSU [8.10.2.1.21]

2.10.2.1.22 remove_executing_unit CSU

This CSU removes a SAF unit from an overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
unitid	unsigned int	Standard
overlays	pointer to OVERLAY_LIST	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
deallocate_veh_id_list	Sec. 2.8.2.2	

Table 2.10-58: remove_executing_unit CSU [8.10.2.1.22]

2.10.2.1.23 cm_force_overlay_recalculation CSU

This CSU cycles through all of the units using a particular overlay and forces them to update their information about that overlay.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
overlay	pointer to OVERLAY	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
readjust_overlay	Sec. 2.14.1.1.9	
LOOKUP_SAFOBJ	Sec. 2.9.1.1	

Table 2.10-59: cm_force_overlay_recalculation CSU [8.10.2.1.23]

2.10.2.1.24 destroy_cm_data CSU

This CSU destroys a control measure type, based on type passed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
type	int	Standard
control_measure_union	pointer to CONTROL_ MEASURE_UNION	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
destroy_point_cm	Sec. 2.10.2.1.25	
destroy_area_cm	Sec. 2.10.2.1.26	
destroy_zone_cm	Sec. 2.10.2.1.27	
destroy_line_cm	Sec. 2.10.2.1.28	
destroy_route	Sec. 2.10.2.4.4	

Table 2.10-60: destroy_cm_data CSU [8.10.2.1.24]

2.10.2.1.25 destroy_point_cm CSU

This CSU calls deallocate_point_cm in order to destroy a point control measure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
point	pointer to POINT_CM	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
deallocate_point_cm	Sec. 2.10.2.2	

Table 2.10-61: destroy_point_cm CSU [8.10.2.1.25]

2.10.2.1.26 destroy_area_cm CSU

This CSU calls deallocate_area_cm in order to destroy an area control measure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
area	pointer to AREA_CM	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
deallocate_area_cm	Sec. 2.	

Table 2.10-62: destroy_area_cm CSU [8.10.2.1.26]

2.10.2.1.27 destroy_zone_cm CSU

This CSU calls deallocate_zone_cm in order to destroy a zone control measure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
zone	pointer to ZONE_CM	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
deallocate_zone_cm	Sec. 2.10.2.2	

Table 2.10-63: destroy_zone_cm CSU [8.10.2.1.27]

2.10.2.1.28 destroy_line_cm CSU

This CSU calls deallocate_line_cm in order to destroy a line control measure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
line	pointer to LINE_CM	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
deallocate_line_cm	Sec. 2.10.2.2	

Table 2.10-64: destroy_line_cm CSU [8.10.2.1.28]

2.10.2.1.29 destroy_cm CSU

This CSU calls deallocate_control_measure in order to deallocate a control measure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
deallocate_control_measure	Sec. 2.10.2.2	

Table 2.10-65: destroy_cm CSU [8.10.2.1.29]

2.10.2.1.30 destroy_cm_list_item CSU

This CSU calls deallocate_measure_list_item in order to remove a control measure item from the list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cml	pointer to CONTROL_MEASURE_LIST	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
deallocate_control_measure_list_item	Sec. 2.10.2.2	

Table 2.10-66: destroy_cm_list_item CSU [8.10.2.1.30]

2.10.2.1.31 destroy_cm_list CSU

This CSU frees up a control measure list by setting the pointers to next and previous lists to NULL and calling destroy_cm_list_item CSU to deallocate the control measure list item's memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cml	pointer to CONTROL_MEASURE_LIST	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
destroy_cm_list_item	Sec. 2.10.2.1.30	

Table 2.10-67: destroy_cm_list CSU [8.10.2.1.31]

2.10.2.1.32 destroy_overlay CSU

This CSU calls destroy_veh_id_list to remove the vehicle id list, and deallocate_overlay to give up the overlay memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
overlay	pointer to OVERLAY	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
destroy_veh_id_list	Sec. 2.8.2.1.12	
deallocate_overlay	Sec. 2.10.2.2	

Table 2.10-68: destroy_overlay CSU [8.10.2.1.32]

2.10.2.1.33 destroy_overlay_list_item CSU

This CSU calls deallocate_overlay_list_item to remove an overlay item from the list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
overlay	pointer to OVERLAY	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
deallocate_overlay_list_item	Sec. 2.10.2.2	

Table 2.10-69: destroy_overlay_list_item CSU [8.10.2.1.33]

2.10.2.1.34 show_overlays CSU

This CSU displays the overlays in the overlay list through a call to show_overlay CSU.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
overlay_list	pointer to OVERLAY_LIST	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
show_overlay	Sec. 2.10.2.1.35	

Table 2.10-70: show_overlays CSU [8.10.2.1.34]

2.10.2.1.35 show_overlay CSU

This CSU displays an overlay through calls to show_vehicle_id_list and show_control_measures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
overlay	pointer to OVERLAY	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
show_vehicle_id_list	Sec. 2.8.2.1.9	
show_control_measures	Sec. 2.10.2.1.36	

Table 2.10-71: show_overlay CSU [8.10.2.1.35]

2.10.2.1.36 show_control_measures CSU

This CSU displays a list of control measures through calls to show_control_measures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cm_list	pointer to CONTROL_MEASURE_LIST	Sec. 2.10.2.2
show_flag	int	Standard
Calls		
Function	**Where Described**	
show_control_measure	Sec. 2.10.2.1.37	

Table 2.10-72: show_control_measures CSU [8.10.2.1.36]

2.10.2.1.37 show_control_measure CSU

This CSU displays a control measure whose type is a function of the parameter passed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2

Calls	
Function	**Where Described**
show_point_cm	Sec. 2.10.2.1.39
show_area_cm	Sec. 2.10.2.1.40
show_zone_cm	Sec. 2.10.2.1.41
show_line_cm	Sec. 2.10.2.1.42
show_route	Sec. 2.10.2.4.14
show_applies_to	Sec. 2.10.2.1.43

Table 2.10-73: show_control_measure CSU [8.10.2.1.37]

2.10.2.1.38 show_point_list CSU

This CSU displays a list of points.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
prefix	pointer to char	Standard
points	pointer to CM_POINT_LIST	Sec. 2.10.2.2

Table 2.10-74: show_point_list CSU [8.10.2.1.38]

2.10.2.1.39 show_point_cm CSU

This CSU displays a control measure point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
point_cm	pointer to POINT_CM	Sec. 2.10.2.2

Table 2.10-75: show_point_cm CSU [8.10.2.1.39]

2.10.2.1.40 show_area_cm CSU

This CSU displays an area type combat instruction set report.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
area_cm	pointer to AREA_CM	Sec. 2.10.2.2

Calls	
Function	**Where Described**
show_point_list	Sec. 2.10.2.1.38

Table 2.10-76: show_area_cm CSU [8.10.2.1.40]

2.10.2.1.41 show_zone_cm CSU

This CSU displays a zone type combat instruction set report.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
zone_cm	pointer to ZONE_CM	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
show_point_list	Sec. 2.10.2.1.38	

Table 2.10-77: show_zone_cm CSU [8.10.2.1.41]

2.10.2.1.42 show_line_cm CSU

This CSU displays a line type combat instruction set report.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
line_cm	pointer to LINE_CM	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
show_point_list	Sec. 2.10.2.1.38	

Table 2.10-78: show_line_cm CSU [8.10.2.1.42]

2.10.2.1.43 show_applies_to CSU

This CSU displays a list of all "applies to" units ids.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
num	unsigned short	Standard
who[ ]	unsigned short	Standard

Table 2.10-79: show_applies_to CSU [8.10.2.1.43]

2.10.2.2 cm.h CSU

/simnet/src/host/cm.h

This CSU contains all of the definitions and symbolic constants needed by the control measure code. Macros which are used for allocating and deallocating control measures are described in Appendix A. The costants are defined in the following table.

Constant	Value
CM_TYPE_POINT	0
CM_TYPE_AREA	1
CM_TYPE_ZONE	2
CM_TYPE_LINE	3
CM_TYPE_ROUTE	4
CM_TYPE_ANY	5

Table 2.10-80: cm.h Constant Definitions

The following tables describe the internal data structures that represent the different types of control measures.

The following typedef struct is tagged point_cm.

Item	Type	Where Type Defined
x	REAL	sim_types.h
y	REAL	sim_types.h
route_id	int	Standard
speed	REAL	sim_types.h
cis	pointer to char	Standard
report	short	Standard

Table 2.10-81: POINT_CM Structure Definition

The following typedef struct is tagged area_cm.

Item	Type	Where Type Defined
points	CM_POINT_LIST	Sec. 2.4.1.1
type	pointer to char	Standard
speed	REAL	sim_types.h
cis	pointer to char	Standard
report	short	Standard

Table 2.10-82: AREA_CM Structure Definition

The following typedef struct is tagged zone_cm.

Item	Type	Where Type Defined
points	CM_POINT_LIST	Sec. 2.4.1.1
type	pointer to char	Standard
speed	REAL	sim_types.h
cis	pointer to char	Standard
report	short	Standard

Table 2.10-83: ZONE_CM Structure Definition

The following typedef struct is tagged line_cm.

Item	Type	Where Type Defined
points	CM_POINT_LIST	Sec. 2.4.1.1
speed	REAL	sim_types.h
cis	pointer to char	Standard
report	short	Standard

Table 2.10-84: LINE_CM Structure Definition

This union allows the passing around of one of these specific control measures without having to know which specific type it is.

Item	Type	Where Type Defined
point	pointer to POINT_CM	This include file
area	pointer to AREA_CM	This include file
zone	pointer to ZONE_CM	This include file
line	pointer to LINE_CM	This include file
route	pointer to ROUTE	Sec. 2.10.2.5 route.h

Table 2.10-85: CONTROL_MEASURE_UNION Union Definition

The following typedef struct is tagged control_measure.

Item	Type	Where Type Defined
id	int	Standard
name[CM_NAME_SIZE]	char	Standard
overlay_id	int	Standard
access_count	int	Standard
num_units	unsigned short	Standard
applies_to[NUM_APPLICABLE_UNITS]	unsigned short	Standard
cm_type	int	Standard
cm_data	CONTROL_MEASURE_UNION	This include file

Table 2.10-86: CONTROL_MEASURE Structure Definition

The following typedef struct is tagged control_measure_list. An control_measure_list is a double-linked list. The first two items are data for this control measure. The last two items point to the data for the next and previous overlays.

Item	Type	Where Type Defined
cm	pointer to CONTROL_MEASURE	This include file
executed	unsigned char	Standard
next	pointer to struct control_measure_list	This typedef struct
prev	pointer to struct control_measure_list	This typedef struct

Table 2.10-87: CONTROL_MEASURE_LIST Structure Definition

The following typedef struct is tagged overlay. An overlay is an identified list of control measures.

Item	Type	Where Type Defined
id	int	Standard
name[CM_NAME_SIZE]	int	Standard
access_count	int	Standard
control_measures	pointer to CONTROL_MEASURE_LIST	This include file
units	pointer to VEH_ID_LIST	

Table 2.10-88: OVERLAY Structure Definition

The following typedef struct is tagged overlay_list. An overlay_list is a double-linked list with each item comtaining an overlay id and a control_measure list. The three items point to the data for the present, previous, and next overlay in that order.

Item	Type	Where Type Defined
overlay	pointer to OVERLAY	This include file
prev	pointer to struct overlay_list	This typedef struct
next	pointer to struct overlay_list	This typedef struct

Table 2.10-89: OVERLAY_LIST Structure Definition

2.10.2.3 navigator.c CSC

/simnet/src/host/navigator.c

The navigator code handles moving the vehicles of the composite over the route. It also take care of notifying the Combat instruction set code when a control measure has been triggered by passing a point, leaving an area, etc.

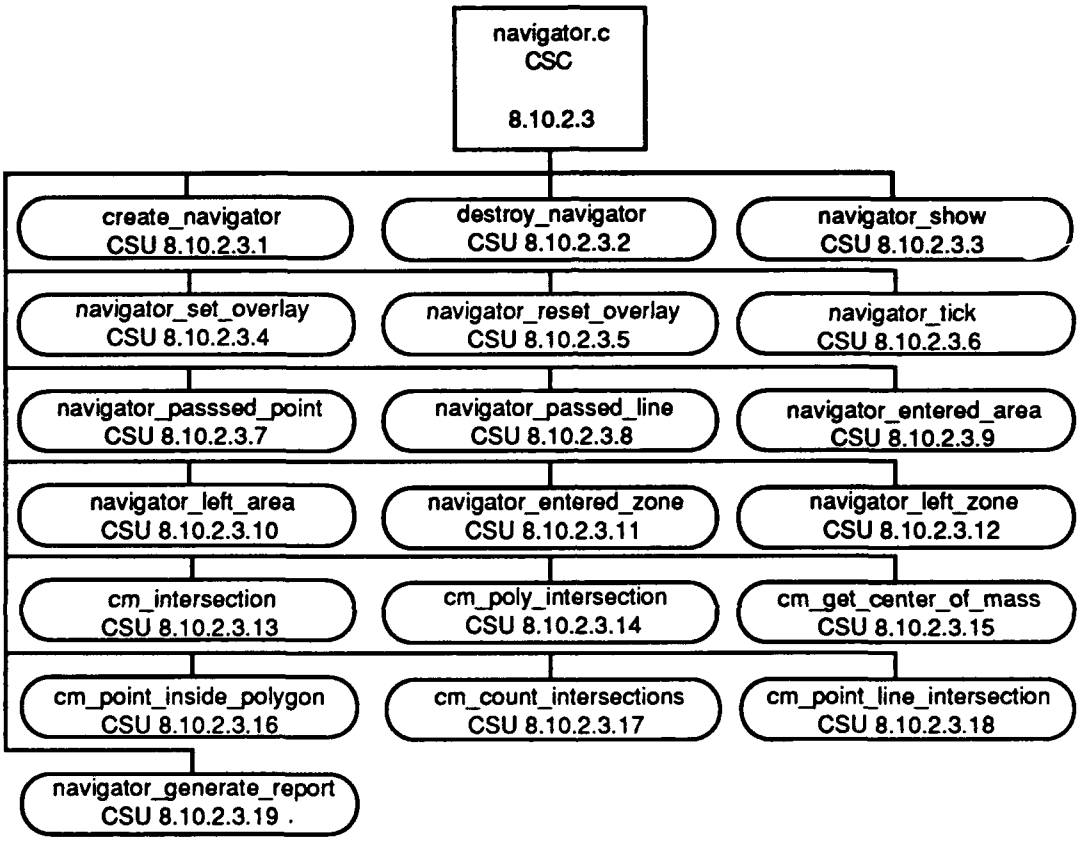


Figure 2.10-6: navigator.c CSC Structure

2.10.2.3.1 create_navigator CSU

This CSU creates a navigator structure.

ReturnValues		
Return Value	**Type**	**Meaning**
navigator	pointer to NAVIGATOR_VARS	pointer to the navigator structure
Calls		
Function	**Where Described**	
allocate_navigator	Sec. 2.9.1.2	

Table 2.10-90: create_navigator CSU [8.10.2.3.1]

2.10.2.3.2 destroy_navigator CSU

This CSU uses deallocate_navigator to destroy the navigator.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
navigator	pointer to NAVIGATOR_VARS	Sec. 2.9.1.2
Calls		
Function	**Where Described**	
deallocate_navigator	Sec. 2.9.1.2	

Table 2.10-91: destroy_navigator CSU [8.10.2.3.2]

2.10.2.3.3 navigator_show CSU

This CSU displays the contents of a navigator structure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
navigator	pointer to NAVIGATOR_VARS	Sec. 2.9.1.2
flags	int	Standard
Calls		
Function	**Where Described**	
show_control_measures	Sec. 2.10.2.1.36	

Table 2.10-92: navigator_show CSU [8.10.2.3.3]

2.10.2.3.4 navigator_set_overlay CSU

This CSU sets the overlay slot of the navigator structure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
navigator	pointer to NAVIGATOR_VARS	Sec. 2.9.1.2
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
overlay	pointer to OVERLAY	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
destroy_cm_list	Sec. 2.10.2.1.31	
cm_get_cm_from_overlay	Sec. 2.10.2.1.17	

Table 2.10-93: navigator_set_overlay CSU [8.10.2.3.4]

2.10.2.3.5 navigator_reset_overlay CSU

This CSU sets the control measure slot of the navigator structure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
navigator	pointer to NAVIGATOR_VARS	Sec. 2.9.1.2
entity	pointer to ENTITY_VARS	Sec. 2.9.1.2
overlay	pointer to OVERLAY	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
cm_recompute_cm_from_overlay	Sec. 2.10.2.1.19	
destroy_cm_list	Sec. 2.10.2.1.30	

Table 2.10-94: navigator_reset_overlay CSU [8.10.2.3.5]

2.10.2.3.6 navigator_tick CSU

This CSU checks for the interaction of a SAF unit with a control measure and calls the appropriate routine to change the unit's behavior based on the type of control measure passed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Calls		
Function	**Where Described**	
cm_point_line_intersection	Sec. 2.10.2.3.18	
navigator_passed_point	Sec. 2.10.2.3.7	
cm_intersection	Sec. 2.10.2.3.13	
navigator_passed_line	Sec. 2.10.2.3.8	
cm_poly_intersection	Sec. 2.10.2.3.14	
cm_point_inside_polygon	Sec. 2.10.2.3.16	
navigator_entered_area	Sec. 2.10.2.3.9	
navigator_left_area	Sec. 2.10.2.3.10	
navigator_entered_zone	Sec. 2.10.2.3.11	
navigator_left_zone	Sec. 2.10.2.3.12	

Table 2.10-95: navigator_tick CSU [8.10.2.3.6]

2.10.2.3.7 navigator_passed_point CSU

This CSU activates a new CIS attached to a point control measure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2

Calls	
Function	**Where Described**
DEBUG_NAVIGATOR	Sec. 2.5.2.2
activate_new_cis	Sec. 2.10.1.1.16

Table 2.10-96: navigator_passed_point CSU [8.10.2.3.7]

2.10.2.3.8 navigator_passed_line CSU

This CSU activates a new CIS attached to a line control measure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2

Calls	
Function	**Where Described**
DEBUG_NAVIGATOR	Sec. 2.5.2.2
activate_new_cis	Sec. 2.10.1.1.16

Table 2.10-97: navigator_passed_line CSU [8.10.2.3.8]

2.10.2.3.9 navigator_entered_area CSU

This CSU activates a new CIS when an area control measure is entered.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2

Calls	
Function	**Where Described**
DEBUG_NAVIGATOR	Sec. 2.5.2.2
activate_new_cis	Sec. 2.10.1.1.16

Table 2.10-98: navigator_entered_area CSU [8.10.2.3.9]

2.10.2.3.10 navigator_left_area CSU

This CSU detects when an area control measure is exited.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2

Calls	
Function	**Where Described**
DEBUG_NAVIGATOR	Sec. 2.5.2.2

Table 2.10-99: navigator_left_area CSU [8.10.2.3.10]

2.10.2.3.11 navigator_entered_zone CSU

This CSU activates a new CIS when a zone control measure is entered.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2

Calls	
Function	**Where Described**
DEBUG_NAVIGATOR	Sec. 2.5.2.2
activate_new_cis	Sec. 2.10.1.1.16

Table 2.10-100: navigator_entered_zone CSU [8.10.2.3.11]

2.10.2.3.14 cm_poly_intersection CSU

This CSU calculates whether a line segment interacts with a polygonal control measure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
point_list	pointer to CM_POINT_LIST	Sec. 2.4.1.1
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	intersected
FALSE	int	not intersected
Calls		
Function	**Where Described**	
line_intersection	Sec. 2.12.1.14.8	

Table 2.10-103: cm_poly_intersection CSU [8.10.2.3.14]

2.10.2.3.15 cm_get_center_of_mass CSU

This CSU calculates the center of mass of a control measure by averaging the points in a point list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
point_list	pointer to CM_POINT_LIST	Sec. 2.4.1.1
result	pointer to REAL	sim_types.h
Calls		
Function	**Where Described**	
vec2_init	Sec. 2.14.3.5.20	
vec2_scale	Sec. 2.14.3.5.28	

Table 2.10-104: cm_get_center_of_mass CSU [8.10.2.3.15]

2.10.2.3.16 cm_point_inside_polygon CSU

This CSU calculates whether a point is inside a polygonal control measure by projecting rays towards infinity and counting the number of intersections with the polygon.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
point_list	pointer to CM_POINT_LIST	Sec. 2.4.1.1
x	INT32	Sec. 2.12.1.18
y	INT32	Sec. 2.12.1.18

2.10.2.3.12 navigator_left_zone CSU

This CSU detects when a zone control measure is exited.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
overlay	pointer to OVERLAY	Sec. 2.10.2.2
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
DEBUG_NAVIGATOR	Sec. 2.5.2.2	

Table 2.10-101: navigator_left_zone CSU [8.10.2.3.12]

2.10.2.3.13 cm_intersection CSU

This CSU calculates whether a line segment intersects a control measure point list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
point_list	pointer to CM_POINT_LIST	Sec. 2.4.1.1
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	intersects
FALSE	int	does not intersect
Calls		
Function	**Where Described**	
line_intersection	Sec. 2.12.1.14.8	

Table 2.10-102: cm_intersection CSU [8.10.2.3.13]

ReturnValues		
Return Value	**Type**	**Meaning**
(ODD_P(count))	int	the number of intersections
TRUE	int	point is inside polygon
Calls		
Function	**Where Described**	
ODD_P	Sec. 2.12.1.18	
cm_count_intersections	Sec. 2.10.2.3.17	

Table 2.10-105: cm_point_inside_polygon CSU [8.10.2.3.16]

2.10.2.3.17 cm_count_intersections CSU

This CSU counts the number of times a line segment intersects a polygon. If the segment passes through a vertex of the polygon, -1 is returned which signifies a non-conclusive result.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
point_list	pointer to CM_POINT_LIST	Sec. 2.4.1.1
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
-1	int	non conclusive result
sum	int	number of intersections
Calls		
Function	**Where Described**	
line_intersection	Sec. 2.12.1.14.8	

Table 2.10-106: cm_count_intersections CSU [8.10.2.3.17]

2.10.2.3.18 cm_point_line_intersection CSU

This CSU determines whether a line segment passes within 200 meters of a point, using a parallel distance calculation.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	int	Standard
y1	int	Standard
x2	int	Standard
y2	int	Standard
p1	int	Standard
p2	int	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	passes within 200 meters of a point
FALSE	int	does not pass within 200 meters of a point
Calls		
Function	**Where Described**	
distance	Sec. 2.12.1.11.1	

Table 2.10-107: cm_point_line_intersection CSU [8.10.2.3.18]

2.10.2.3.19 navigator_generate_report CSU

This CSU generates the string to be included in the message back to the workstation.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
action	pointer to char	Standard
cm	pointer to CONTROL_MEASURE	Sec. 2.10.2.2
Calls		
Function	**Where Described**	
sbx_printf	Sec. 2.4.3.2.8	
OBJ_VEHICLEID	Sec. 2.9.1.1	

Table 2.10-108: navigator_generate_report CSU [8.10.2.3.19]

2.10.2.4 route.c CSC

/simnet/src/host/route.c

This CSC contains all the code for creating and destroying routes, handling special cases such as bridges, and modifying existing routes.

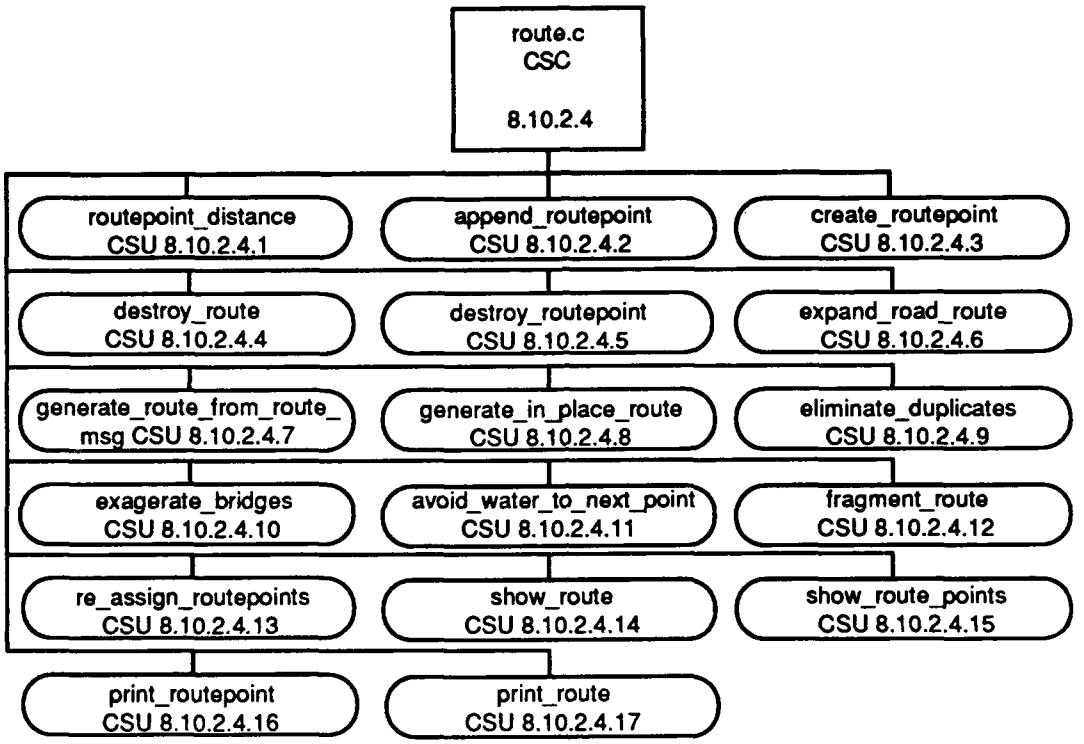


Figure 2.10-7: route.c CSC Structure

2.10.2.4.1 routepoint_distance CSU

This CSU calculates the distance between two routepoints.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
rpoint1	pointer to ROUTEPOINT	Sec. 2.10.2.5
rpoint2	pointer to ROUTEPOINT	Sec. 2.10.2.5
ReturnValues		
Return Value	**Type**	**Meaning**
(sqrt(...))	REAL	Distance between points
Calls		
Function	**Where Described**	
vec_sub	Sec. 2.6.2.65.1 Vehicles CSCI SDD	

Table 2.10-109: routepoint_distance CSU [8.10.2.4.1]

2.10.2.4.2 append_routepoint CSU

This CSU appends a routepoint onto the end of a route.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to ROUTE	Sec. 2.10.2.5
routepoint	pointer to ROUTEPOINT	Sec. 2.10.2.5
Calls		
Function	**Where Described**	
routepoint_distance	Sec. 2.10.2.4.1	

Table 2.10-110: append_routepoint CSU [8.10.2.4.2]

2.10.2.4.3 create_routepoint CSU

This CSU creates a new routepoint.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
id	int	Standard
x	int	Standard
y	int	Standard
road	int	Standard
bridge	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
routepoint	pointer to ROUTEPOINT	Pointer to routepoint
Calls		
Function	**Where Described**	
allocate_routepoint	Sec. 2.10.2.5	

Table 2.10-111: create_routepoint CSU [8.10.2.4.3]

2.10.2.4.4 destroy_route CSU

This CSU destroys a route by calling destroy_routepoint to remove all routpoints, and deallocate_route to remove the route memory heap.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to ROUTE	Sec. 2.10.2.5

Calls	
Function	**Where Described**
destroy_routepoint	Sec. 2.10.2.4.5
deallocate_route	Sec. 2.10.2.5

Table 2.10-112: destroy_route CSU [8.10.2.4.4]

2.10.2.4.5 destroy_routepoint CSU

This CSU destroys a routpoint using deallocate_routepoint.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to ROUTE	Sec. 2.10.2.5
Calls		
Function	**Where Described**	
deallocate_routepoint	Sec. 2.10.2.5	

Table 2.10-113: destroy_routepoint CSU [8.10.2.4.5]

2.10.2.4.6 expand_road_route CSU

This CSU takes the road segment indices and directions that come over from the workstation and expands them into a set of routepoints that specify each point in the road segments. The initial and final points are used to specify when a partial road segment is to be used.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
segdir	pointer to SBX_ROUTE_PT	Sec. 2.4.1.1
initial	pointer to SBX_ROUTE_PT	Sec. 2.4.1.1
final	pointer to SBX_ROUTE_PT	Sec. 2.4.1.1
route	pointer to ROUTE	Sec. 2.10.2.5
Calls		
Function	**Where Described**	
create_routepoint	Sec. 2.10.2.4.3	
append_routepoint	Sec. 2.10.2.4.2	

Table 2.10-114: expand_road_route CSU [8.10.2.4.6]

2.10.2.4.7 generate_route_from_route_msg CSU

This CSU takes the route message from the workstation and creates a route. Road points are specified with a negative id value. Road points consist of road index/direction pairs, where route points consist of x,y locations.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route_msg	pointer to ROUTE_MSG	Sec. 2.4.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
NULL	pointer to ROUTE	No route message
route	pointer to ROUTE	Pointer to the route
Calls		
Function	**Where Described**	
allocate_route	Sec. 2.10.2.4.2	
EVEN_P	Sec. 2.12.1.18	
ODD_P	Sec. 2.12.1.18	
expand_road_route	Sec. 2.10.2.4.6	
create_routepoint	Sec. 2.10.2.4.3	
append_routepoint	Sec. 2.10.2.4.2	
eliminate_duplicates	Sec. 2.10.2.4.9	
exagerate_bridges	Sec. 2.10.2.4.10	

Table 2.10-115: generate_route_from_route_msg CSU [8.10.2.4.7]

2.10.2.4.8 generate_in_place_route CSU

This CSU creates a route with a single point at the SAF object's current location.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
ReturnValues		
Return Value	**Type**	**Meaning**
route	pointer to ROUTE	Pointer to the route
Calls		
Function	**Where Described**	
OBJ_POSITION	Sec. 2.9.1.1	
allocate_route	Sec. 2.10.2.4.2	
create_routepoint	Sec. 2.10.2.4.3	
append_routepoint	Sec. 2.10.2.4.2	

Table 2.10-116: generate_in_place_route CSU [8.10.2.4.8]

2.10.2.4.9 eliminate_duplicates CSU

This CSU eliminates consecutive route points that have the same locations.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to ROUTE	Sec. 2.10.2.5

Table 2.10-117: eliminate_duplicates CSU [8.10.2.4.9]

2.10.2.4.10 exagerate_bridges CSU

This CSU marks route points that are near bridge points on a route.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to ROUTE	Sec. 2.10.2.5
Calls		
Function	**Where Described**	
vec_sub	Sec. 2.6.2.65.1 Vehicles CSCI SDD	
vec2_dot	Sec. 2.14.3.5.24	

Table 2.10-118: exagerate_bridges CSU [8.10.2.4.10]

2.10.2.4.11 avoid_water_to_next_point CSU

This CSU attempts to modify routes that pass through water by searching for bridges, fording points, or river ends.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route_point	pointer to ROUTEPOINT	Sec. 2.10.2.5
offset	int	Standard
Calls		
Function	**Where Described**	
find_route	Sec. 2.12.1.8.1	
last_item	Sec. 2.12.1.7.10	
create_routepoint	Sec. 2.10.2.4.3	
routepoint_distance	Sec. 2.10.2.4.1	
free_list	Sec. 2.12.1.7.4	

Table 2.10-119: avoid_water_to_next_point CSU [8.10.2.4.11]

2.10.2.4.12 fragment_route CSU

This CSU goes down the route and breaks all the links.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to ROUTE	Sec. 2.10.2.5

Table 2.10-120: fragment_route CSU [8.10.2.4.12]

2.10.2.4.13 re_assign_routepoints CSU

This CSU replaces a route point on a route with a new route point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
old_point	pointer to ROUTEPOINT	Sec. 2.10.2.5
new_point	pointer to ROUTEPOINT	Sec. 2.10.2.5
Calls		
Function	**Where Described**	
vec2_sub	Sec. 2.14.3.5.23	
vec2_init	Sec. 2.14.3.5.20	
vec2_dot	Sec. 2.14.3.5.24	

Table 2.10-121: re_assign_routepoints CSU [8.10.2.4.13]

2.10.2.4.14 show_route CSU

This CSU displays all the route points for a route through call(s) to show_route_points.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to ROUTE	Sec. 2.10.2.5
Calls		
Function	**Where Described**	
show_route_points	Sec. 2.10.2.4.15	

Table 2.10-122: show_route CSU [8.10.2.4.14]

2.10.2.4.15 show_route_points CSU

This CSU displays the route's points through calls to print_routepoint.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
pt	pointer to ROUTEPOINT	Sec. 2.10.2.5

Calls	
Function	**Where Described**
print_routepoint	Sec. 2.10.2.4.16

Table 2.10-123: show_route_points CSU [8.10.2.4.15]

2.10.2.4.16 print_routepoint CSU

This CSU displays a route point passed, either bridge or road.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
pt	pointer to ROUTEPOINT	Sec. 2.10.2.5

Table 2.10-124: print_routepoint CSU [8.10.2.4.16]

2.10.2.4.17 print_route CSU

The CSU displays the route passed by getting the points from the route and calling print_routepoint.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to ROUTE	Sec. 2.10.2.5
Calls		
Function	**Where Described**	
print_routepoint	Sec. 2.10.2.4.16	

Table 2.10-125: print_route CSU [8.10.2.4.17]

2.10.2.5 route.h CSU

/simnet/src/host/route.h

This CSU contains the structures and macros (defined in Appendix A) needed by the route following code.

The following typedef struct is tagged routepoint.

Item	Type	Where Type Defined
prev	pointer to struct routepoint	This typedef struct
id	int	Standard
point	VECTOR	sim_types.h
next	pointer to struct routepoint	This typedef struct
road : 1	int	Standard
bridge :1	int	Standard
prev_dist	REAL	sim_types.h
next_dist	REAL	sim_types.h

Table 2.10-126: ROUTEPOINT Structure Definition

The following typedef struct is tagged route and consists of the route id along with pointers to the start and ending routepoints.

Item	Type	Where Type Defined
id	int	Standard
start	pointer to ROUTEPOINT	Previous typedef struct
end	pointer to ROUTEPOINT	Previous typedef struct

Table 2.10-127: ROUTE Structure Definition

2.11 CREATE CSC

This CSC contains the CSUs for creating units and vehicles. These CSUs are usually called from the SAF command interface but can also be called from the parser interface. They interprete the unit organization data from the parameter files to determine the hierarchy which needs to be created.

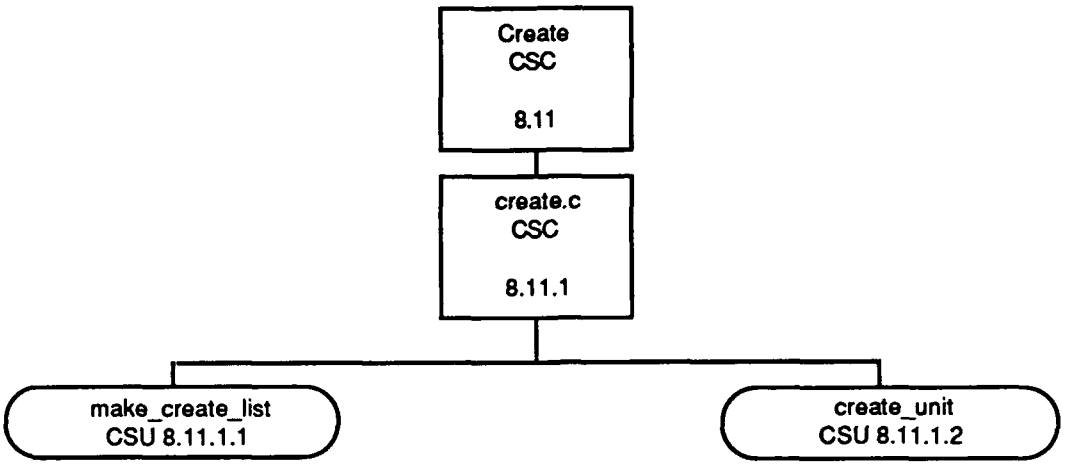


Figure 2.11-1: Create CSC Structure

2.11.1 create.c CSC

/simnet/src/host/create.c

This file creates the composites requested for the simulated vehicles, then has the vehicles created. In addition to the two CSUs, the file contains a structure, tagged creation_item, shown in the following table.

Item	Type	Where Type Defined
saf_type	unsigned short	Standard
*job	pointer to char	Standard
*echelon	pointer to char	Standard
*echelon_type	pointer to char	Standard
superior	unsigned short	Standard
rank	unsigned short	Standard
bumper	int	Standard

Table 2.11-1: CREATION_ITEM Structure Definition

2.11.1.1 make_create_list CSU

This CSU generates a list of creation information used in the creation of a SAF object.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*table	pointer to DATA_UNION	Sec. 2.1.1.5
*echelon	pointer to char	Standard
create_list[ ]	CREATION_ITEM	Sec. 2.11.1 This CSC
vnum	unsigned short	Standard
superior	unsigned short	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
vnum	unsigned short	Recursion.
Calls		
Function	**Where Described**	
find_tag	Sec. 2.1.1.4.3	
make_create_list	Sec. 2.11.1.1	

Table 2.11-2: make_create_list CSU [8.11.1.1]

2.11.1.2 create_unit CSU

This CSU creates a SAF object.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
forceID	ForceID	basic.h
cD	unsigned char	Standard
cO	unsigned char	Standard
tactics	unsigned char	Standard
*echelon	pointer to char	Standard
*echelon_type	pointer to char	Standard
*formation	pointer to char	Standard
heading	REAL	sim_types.h
*position	pointer to REAL	sim_types.h
*owner	pointer to SBX_CONNECTION_VARS	Sec. 2.4.3.3
sbx_uniq_id	unsigned char	Standard
battalion	unsigned short	Standard
company	unsigned char	Standard
plattoon	unsigned char	Standard
percent_ammo	REAL	sim_types.h
percent_fuel	REAL	sim_types.h

ReturnValues		
Return Value	**Type**	**Meaning**
0	unsigned int	No unit created.
OBJ_VEHICLEID(...)	unsigned int	Created unit.

Calls	
Function	**Where Described**
RANGE_CLIP	Sec. 2.14.3.9 See Appendix A
find_tag	Sec. 2.1.1.4.3
make_create_list	Sec. 2.11.1.1
create_saf_vehicle	Sec. 2.6.1.1.1
saf_vehicle_start_ticking	Sec. 2.6.1.1.3
IS_AIRCRAFT	
ft_int	Sec. 2.14.1.2.11
get_symbol	Sec. 2.1.1.3.2
create_composite	Sec. 2.8.1.3.1
composite_start_ticking	Sec. 2.8.1.3.11
saf_vehicle_set_superior	Sec. 2.6.1.1.4
composite_set_superior	Sec. 2.8.1.3.4
composite_set_sbx_ids	Sec. 2.8.1.3.24
composite_get_composite_bumpers	Sec. 2.8.1.3.22
composite_assume_formation	Sec. 2.8.1.3.26
composite_teleport_to_station	Sec. 2.8.1.3.28
unit_send_pae_data	Sec. 2.8.1.3.30
OBJ_VEHICLEID	Sec. 2.9.1.1 See Appendix A

Table 2.11-3: create_unit CSU [8.11.1.2]

2.12 TERRAIN CSC

This CSC contains the data structures for representing the terrain which the vehicles traverse.

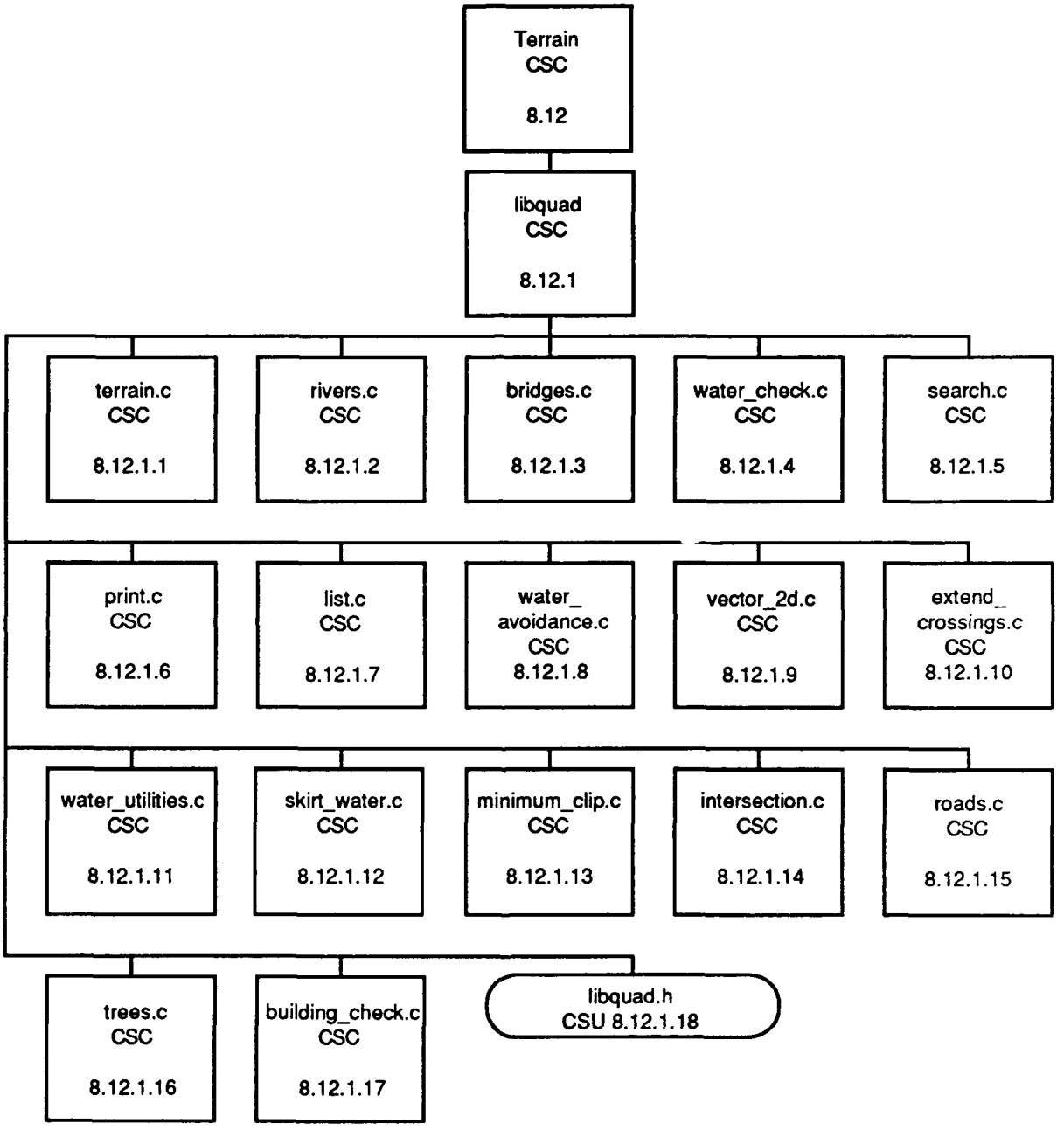


Figure 2.12-1: Terrain CSC Structure

There is a quadtree representation and a polygonal representation of the terrain. The quadtree represents terrain objects at a high level, and organizes them into a quadtree data structure. The polygonal data structure represents the terrain as a carpet of polygons. The quadtree database is used to reason about the terrain for road following, water avoidance, and obstacle avoidance. The polygonal representation is used to determine the support

polygon of the ground vehicles. It is used to determine the maximal accelerations of the vehicles as well as their orientations. It is also used to determine if two vehicles have a line of sight between them. This CSC contains many CSCs to access the terrain representations and to obtain data from them.

2.12.1 libquad CSC

/simnet/libsrc/libquad

This library handles terrain-handling functions, such as water skirting, road following, and bridge crossing, for vehicles.

2.12.1.1 terrain.c CSC

/simnet/libsrc/libquad/terrain.c

This CSC contains CSUs dealing with the terrain database.

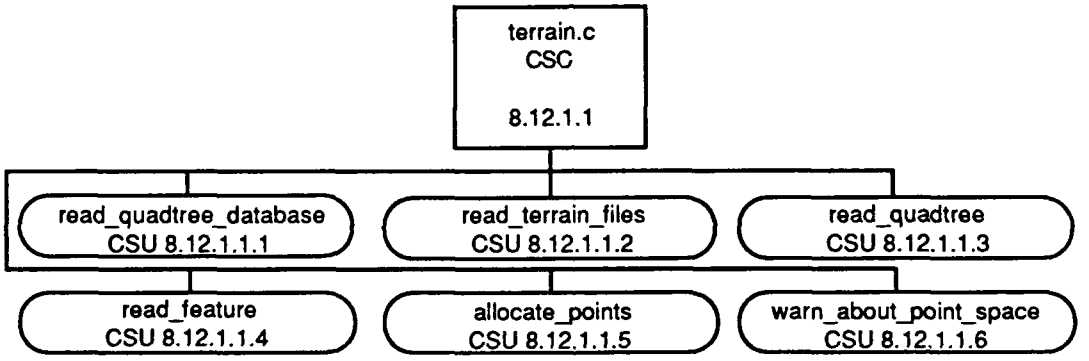


Figure 2.12-2: terrain.c CSC Structure

2.12.1.1.1 read_quadtree_database CSU

This CSU reads the whole database.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Calls		
Function	**Where Described**	
read_terrain_files	Sec. 2.12.1.1.2	
read_quadtree	Sec. 2.12.1.1.3	

Table 2.12-1: read_quadtree_database CSU [8.12.1.1.1]

2.12.1.1.2 read_terrain_files CSU

This CSU reads all feature files.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Calls		
Function	**Where Described**	
create_rivers	Sec. 2.12.1.2.1	
create_river_ints	Sec. 2.12.1.2.3	
create_roads	Sec. 2.12.1.15.1	
create_road_ints	Sec. 2.12.1.15.3	
create_bridges	Sec. 2.12.1.3.1	
create_lakes	Sec. 2.12.1.3.3	
create_trees	Sec. 2.12.1.16.1	
create_tree_canopies	Sec. 2.12.1.16.3	
create_buildings	Sec. 2.12.1.16.5	
create_contours	Sec. 2.12.1.16.7	
warn_about_point_space	Sec. 2.12.1.1.6	

Table 2.12-2: read_terrain_files CSU [8.12.1.1.2]

2.12.1.1.3 read_quadtree CSU

This CSU reads the quadtree and feature structures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
read_feature	Sec. 2.12.1.1.4	

Table 2.12-3: read_quadtree CSU [8.12.1.1.3]

2.12.1.1.4 read_feature CSU

This CSU reads a list of feature indices for a feature structure in a quad node.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
fd	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
feat	pointer to int	Number of indices.

Calls	
Function	**Where Described**
heap_allocate	Sec. 2.14.2.1.1

Table 2.12-4: read_feature CSU [8.12.1.1.4]

2.12.1.1.5 allocate_points CSU

This CSU allocates an array of points.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
n	unsigned short	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
ret	pointer to INT32	Allocated space.
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	

Table 2.12-5: allocate_points CSU [8.12.1.1.5]

2.12.1.1.6 warn_about_point_space CSU

This CSU warns the user when the point space is being increased.

2.12.1.2 rivers.c CSC

/simnet/libsrc/libquad/rivers.c

This CSC contains CSUs that deal with river creation (segments and intersections), and that print information on the above conditions if DEBUG is defined.

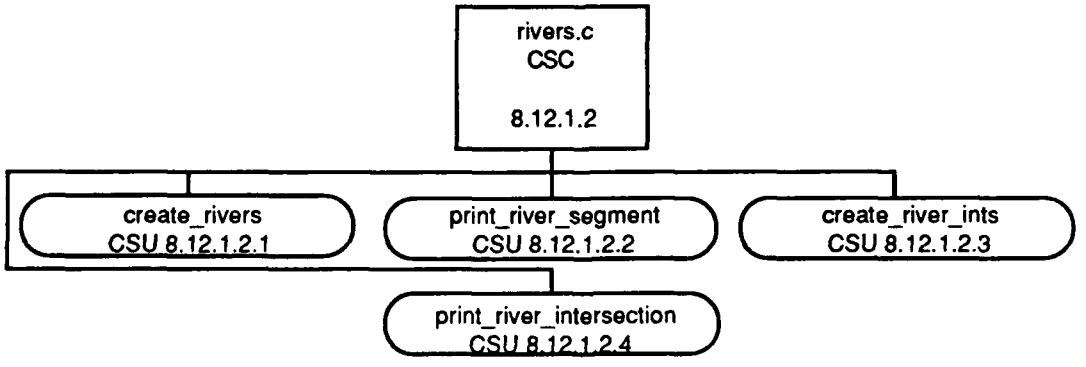


Figure 2.12-3: rivers.c CSC Structure

2.12.1.2.1 create_rivers CSU

This CSU reads river segment structures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
allocate_points	Sec. 2.12.1.1.5	
print_river_segment	Sec. 2.12.1.2.2	

Table 2.12-6: create_rivers CSU [8.12.1.2.1]

2.12.1.2.2 print_river_segment CSU

This CSU prints river segment information.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
segment_index	int	Standard

Table 2.12-7: print_river_segment CSU [8.12.1.2.2]

2.12.1.2.3 create_river_ints CSU

This CSU creates river intersections.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
print_river_intersection	Sec. 2.12.1.2.4	

Table 2.12-8: create_river_ints CSU [8.12.1.2.3]

2.12.1.2.4 print_river_intersection CSU

This CSU prints information about river intersections.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
int_index	int	Standard

Table 2.12-9: print_river_intersection CSU [8.12.1.2.4]

2.12.1.3 bridges.c CSC

/simnet/libsrc/libquad/bridges.c

This CSC contains CSUs that deal with bridge and lake creation, and that print information about the above conditions if DEBUG is defined.

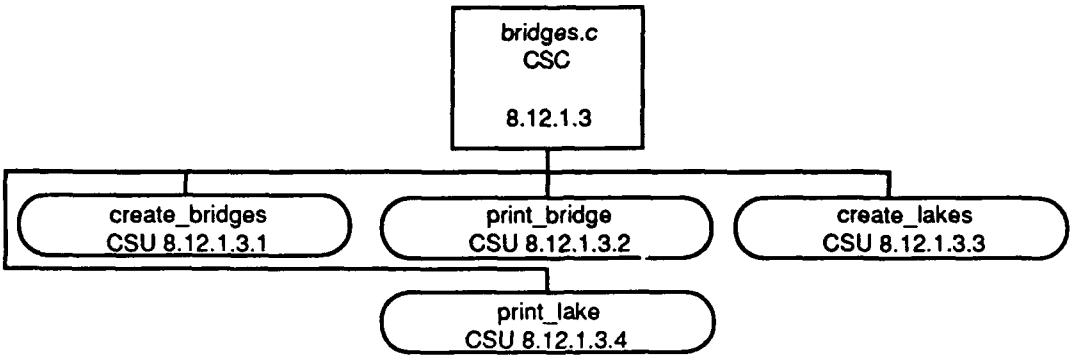


Figure 2.12-4: bridges.c CSC Structure

2.12.1.3.1 create_bridges CSU

This CSU reads bridge structures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
allocate_points	Sec. 2.12.1.1.5	
print_bridge	Sec. 2.12.1.3.2	

Table 2.12-10: create_bridges CSU [8.12.1.3.1]

2.12.1.3.2 print_bridge CSU

This CSU prints information about a bridge.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
bridge_index	int	Standard

Table 2.12-11: print_bridge CSU [8.12.1.3.2]

2.12.1.3.3 create_lakes CSU

This CSU reads lake structures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
allocate_points	Sec. 2.12.1.1.5	
print_lake	Sec. 2.12.1.3.4	

Table 2.12-12: create_lakes CSU [8.12.1.3.3]

2.12.1.3.4 print_lake CSU

This CSU prints information about a lake.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
lake_index	int	Standard

Table 2.12-13: print_lake CSU [8.12.1.3.4]

2.12.1.4 water_check.c CSC

/simnet/libsrc/libquad/water_check.c

This CSC consists of twelve CSUs which check for route intersections with lakes and rivers.

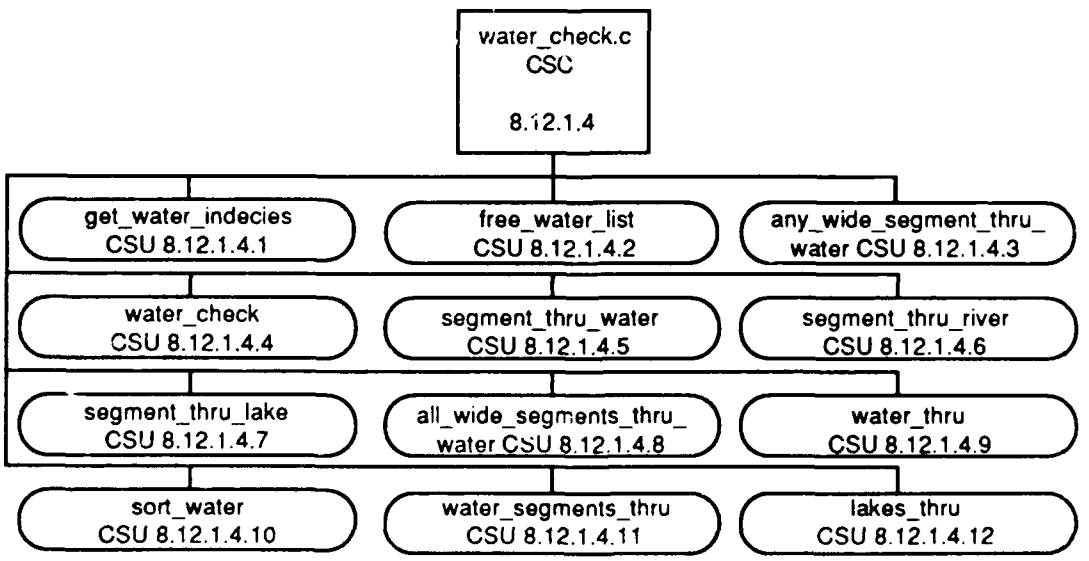


Figure 2.12-5: water_check.c CSC Structure

2.12.1.4.1 get_water_indecies CSU

This CSU returns all of the river and lake indices that fall within a bounding box of a line segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
water_index_list	pointer to LIST	Water indices.
Calls		
Function	**Where Described**	
get_quad_nodes	Sec. 2.12.1.5.2	
MIN	See Appendix A	
push	Sec. 2.12.1.7.3	
free_list	Sec. 2.12.1.7.4	
remove_duplicates	Sec. 2.12.1.7.5	

Table 2.12-14: get_water_indecies CSU [8.12.1.4.1]

2.12.1.4.2 free_water_list CSU

This CSU frees a list of river and lake indices.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
water_list	pointer to LIST	Sec. 2.12.1.18
Calls		
Function	**Where Described**	
free_list	Sec. 2.12.1.7.4	
FIRST	See Appendix A	
SECOND	See Appendix A	

Table 2.12-15: free_water_list CSU [8.12.1.4.2]

2.12.1.4.3 any_wide_segment_thru_water CSU

This CSU checks if a line segment intersects a river or lake, or whether lines parallel to the line segment intersect a river or lake. It returns the first water intersection found.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
offset	INT32	Sec. 2.12.1.18
water_index_list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
water	pointer to WATER_INT	Water intersection found.
NULL_PTR	pointer to WATER_INT	No water intersection found.
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
segment_thru_water	Sec. 2.12.1.4.5	
VEC_SUB_2D	See Appendix A	
vec_normalize_2d	Sec. 2.12.1.9.2	
vec_rotate_2d	Sec. 2.12.1.9.3	
VEC_SCALE_2D	See Appendix A	
VEC_ADD_2D	See Appendix A	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.12-16: any_wide_segment_thru_water CSU [8.12.1.4.3]

2.12.1.4.4 water_check CSU

This CSU checks whether a line segment intersects a river.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
return_val = TRUE	int	Line segment intersects a river.
return_val = FALSE	int	Line segment does not intersect a river.

Calls	
Function	**Where Described**
get_water_indecies	Sec. 2.12.1.4.1
segment_thru_river	Sec. 2.12.1.4.6
FIRST	See Appendix A
segment_thru_lake	Sec. 2.12.1.4.7
SECOND	See Appendix A
free_water_list	Sec. 2.12.1.4.2

Table 2.12-17: water_check CSU [8.12.1.4.4]

2.12.1.4.5 segment_thru_water CSU

This CSU checks if a line segment intersects a river or lake.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
xy1	pointer to VECTOR_2D	Sec. 2.12.1.18
xy2	pointer to VECTOR_2D	Sec. 2.12.1.18
water_index_list	pointer to LIST	Sec. 2.12.1.18
water	pointer to WATER_INT	Sec. 2.12.1.18

Calls	
Function	**Where Described**
get_water_indecies	Sec. 2.12.1.4.1
segment_thru_river	Sec. 2.12.1.4.6
FIRST	See Appendix A
segment_thru_lake	Sec. 2.12.1.4.7
SECOND	See Appendix A
free_water_list	Sec. 2.12.1.4.2

Table 2.12-18: segment_thru_water CSU [8.12.1.4.5]

2.12.1.4.6 segment_thru_river CSU

This CSU checks whether a line segment intersects an unfordable river segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
index_list	pointer to LIST	Sec. 2.12.1.18
first_water	pointer to WATER_INT	Sec. 2.12.1.18

ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Line segment intersects an unfordable river segment.
FALSE	int	Line segment does not intersect an unfordable river segment.

Calls	
Function	**Where Described**
possible_intersection	Sec. 2.12.1.14.2
seg_intersection	Sec. 2.12.1.14.1

Table 2.12-19: segment_thru_river CSU [8.12.1.4.6]

2.12.1.4.7 segment_thru_lake CSU

This CSU checks whether a line segment intersects a lake polygon, or whether it is completely contained within a lake polygon.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
index_list	pointer to LIST	Sec. 2.12.1.18
first_water	pointer to WATER_INT	Sec. 2.12.1.18

ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Line segment intersects or is completely contained within a lake polygon.
FALSE	int	Line segment does not intersect and is not completely contained within a lake polygon.

Calls	
Function	**Where Described**
possible_intersection	Sec. 2.12.1.14.2
seg_intersection	Sec. 2.12.1.14.1
segment_inside_polygon	Sec. 2.12.1.14.5

Table 2.12-20: segment_thru_lake CSU [8.12.1.4.7]

2.12.1.4.8 all_wide_segments_thru_water CSU

This CSU returns all of the intersections of a line segment, or lines parallel to the segment, with lakes and rivers.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
offset	INT32	Sec. 2.12.1.18
water_index_list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
original_list	pointer to LIST	List of intersections.
current_list	pointer to LIST	List of intersections.
Calls		
Function	**Where Described**	
water_thru	Sec. 2.12.1.4.9	
FIRST	See Appendix A	
VEC_SUB_2D	See Appendix A	
vec_normalize_2d	Sec. 2.12.1.9.2	
vec_rotate_2d	Sec. 2.12.1.9.3	
VEC_SCALE_2D	See Appendix A	
VEC_ADD_2D	See Appendix A	
free_list	Sec. 2.12.1.7.4	

Table 2.12-21: all_wide_segments_thru_water CSU [8.12.1.4.8]

2.12.1.4.9 water_thru CSU

This CSU returns a list of all of the intersections of a line segment with rivers and lakes. This list is sorted by distance from the first point of the line segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
xy1	pointer to VECTOR_2D	Sec. 2.12.1.18
xy2	pointer to VECTOR_2D	Sec. 2.12.1.18
water_index_list	pointer to LIST	Sec. 2.12.1.18
water_list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
water_list	pointer to LIST	Sorted intersections.

Calls	
Function	**Where Described**
get_water_indecies	Sec. 2.12.1.4.1
water_segments_thru	Sec. 2.12.1.4.11
FIRST	See Appendix A
lakes_thru	Sec. 2.12.1.4.12
SECOND	See Appendix A
sort_water	Sec. 2.12.1.4.10
free_water_list	Sec. 2.12.1.4.2

Table 2.12-22: water_thru CSU [8.12.1.4.9]

2.12.1.4.10 sort_water CSU

This CSU sorts a list of water intersections by finding the two closest intersections to a point and puting them on the front of the list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x	INT32	Sec. 2.12.1.18
y	INT32	Sec. 2.12.1.18
water_list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
water_list	pointer to LIST	Successful.
NULL_PTR	pointer to LIST	Unsuccessful.
Calls		
Function	**Where Described**	
distance	Sec. 2.12.1.11.1	
nthcdr	Sec. 2.12.1.7.8	
delete	Sec. 2.12.1.7.7	
push	Sec. 2.12.1.7.3	

Table 2.12-23: sort_water CSU [8.12.1.4.10]

2.12.1.4.11 water_segments_thru CSU

This CSU returns a list of all river segments that intersect a line segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
index_list	pointer to LIST	Sec. 2.12.1.18
water_list	pointer to LIST	Sec. 2.12.1.18

ReturnValues		
Return Value	**Type**	**Meaning**
water_list	pointer to LIST	Intersecting river segments.
Calls		
Function	**Where Described**	
possible_intersection	Sec. 2.12.1.14.2	
line_intersection	Sec. 2.12.1.14.8	
heap_allocate	Sec. 2.14.2.1.1	
make_vector_2d	Sec. 2.12.1.9.1	
push	Sec. 2.12.1.7.3	

Table 2.12-24: water_segments_thru CSU [8.12.1.4.11]

2.12.1.4.12 lakes_thru CSU

This CSU returns a list of all lakes that intersect a line segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
index_list	pointer to LIST	Sec. 2.12.1.18
water_list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
water_list	pointer to LIST	Intersecting lakes.
Calls		
Function	**Where Described**	
possible_intersection	Sec. 2.12.1.14.2	
segment_inside_polygon	Sec. 2.12.1.14.5	
heap_allocate	Sec. 2.14.2.1.1	
make_vector_2d	Sec. 2.12.1.9.1	
push	Sec. 2.12.1.7.3	
line_intersection	Sec. 2.12.1.14.8	

Table 2.12-25: lakes_thru CSU [8.12.1.4.12]

2.12.1.5 search.c CSC

/simnet/libsrc/libquad/search.c

This CSC contain CSUs that deal with searching for, obtaining, analyzing, and testing the validity of quad nodes.

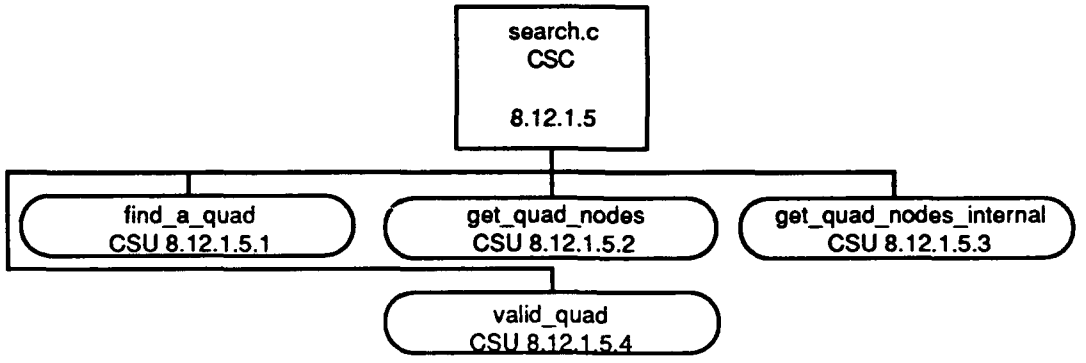


Figure 2.12-6: search.c CSC Structure

2.12.1.5.1 find_a_quad CSU

This CSU returns the left quad tree node at the given location.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
node	pointer to QUAD_NODE	Sec. 2.12.1.18
qx	INT32	Sec. 2.12.1.18
qy	INT32	Sec. 2.12.1.18
size	INT32	Sec. 2.12.1.18
x	INT32	Sec. 2.12.1.18
y	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to QUAD_FEATURES	No node found.
feature_array + node-> feature_index	pointer to QUAD_FEATURES	Successful.
find_a_quad(...)	pointer to QUAD_FEATURES	Recursion.
Calls		
Function	**Where Described**	
find_a_quad	Sec. 2.12.1.5.1	

Table 2.12-26: find_a_quad CSU [8.12.1.5.1]

2.12.1.5.2 get_quad_nodes CSU

This CSU is the top level call to the recursive function get_quad_nodes_internal.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
node	pointer to QUAD_NODE	Sec. 2.12.1.18
list	pointer to LIST	Sec. 2.12.1.18
qx	INT32	Sec. 2.12.1.18
qy	INT32	Sec. 2.12.1.18
size	INT32	Sec. 2.12.1.18
x_min	INT32	Sec. 2.12.1.18
y_min	INT32	Sec. 2.12.1.18
x_max	INT32	Sec. 2.12.1.18
y_max	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
get_quad_nodes_internal(...)	pointer to LIST	Function call.
Calls		
Function	**Where Described**	
get_quad_nodes_internal	Sec. 2.12.1.5.3	

Table 2.12-27: get_quad_nodes CSU [8.12.1.5.2]

2.12.1.5.3 get_quad_nodes_internal CSU

This CSU recursively finds all of the quad tree nodes that a given area intersects.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
node	pointer to register QUAD_NODE	Sec. 2.12.1.18
list	pointer to register LIST	Sec. 2.12.1.18
qx	register INT32	Sec. 2.12.1.18
qy	register INT32	Sec. 2.12.1.18
size	register INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to LIST	No node found.
list	pointer to register LIST	Successful.
Calls		
Function	**Where Described**	
valid_quad	Sec. 2.12.1.5.4	
push	Sec. 2.12.1.7.3	
get_quad_nodes_internal	Sec. 2.12.1.5.3	

Table 2.12-28: get_quad_nodes_internals CSU [8.12.1.5.3]

2.12.1.5.4 valid_quad CSU

This CSU tests whether an area is totally within, totally without, or partially within a quad tree node, or whether a quad tree node is totally within or totally without a given area.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
qx	register INT32	Sec. 2.12.1.18
qy	register INT32	Sec. 2.12.1.18
size	register INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
0	int	Area totally out of quad, or quad totally out of area.
1	int	Quad totally in area, or area totally in quad, or area partially in quad.

Table 2.12-29: valid_quad CSU [8.12.1.5.4]

2.12.1.6 print.c CSC

/simnet/libsrc/libquad/print.c

This CSC consists of a single CSU, print_quad_node.

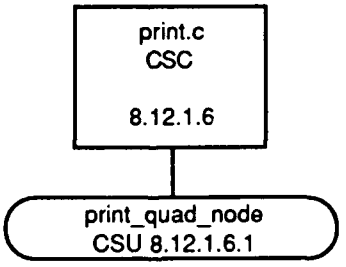


Figure 2.12-7: print.c CSC Structure

2.12.1.6.1 print_quad_node CSU

This CSU prints a quad node, "Leaf Node". This information includes all water segments, water intersection indices, road segments, road intersection indices, bridge segments, and the child node.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
node_index	int	Standard

Table 2.12-30: print_quad_node CSU [8.12.1.6.1]

2.12.1.7 list.c CSC

/simnet/libsrc/libquad/list.c

This CSC contains CSUs that deal with LIST structures.

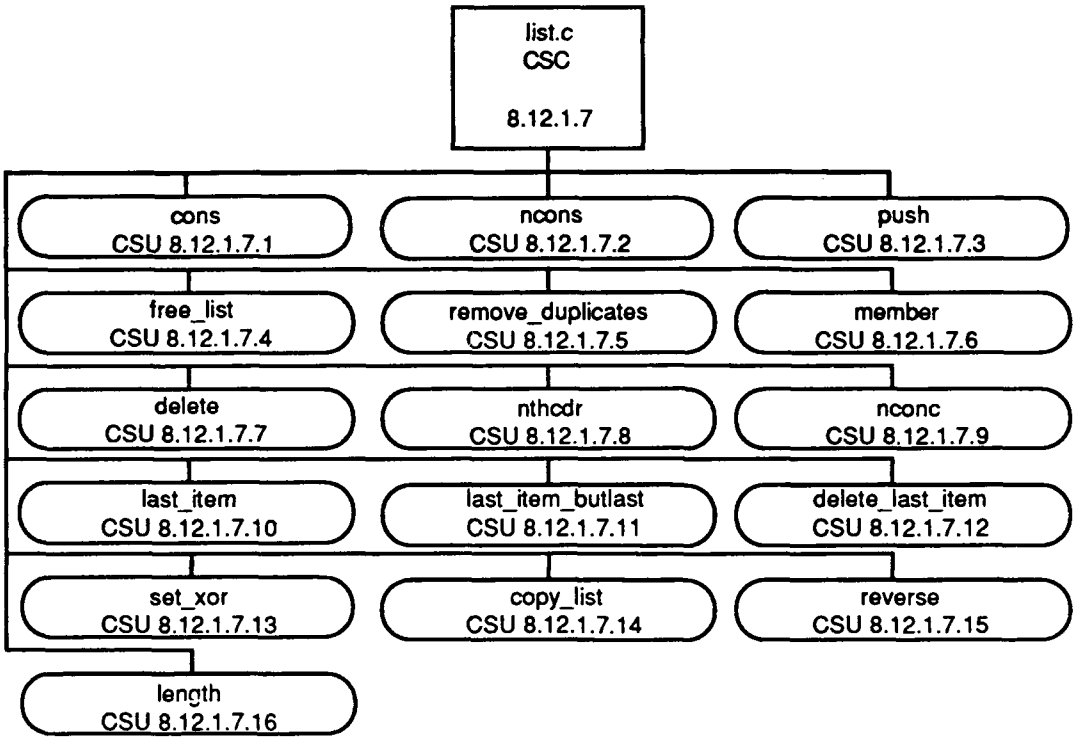


Figure 2.12-8: list.c CSC Structure

2.12.1.7.1 cons CSU

This CSU creates a new cons.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
car	int	Standard
cdr	pointer to LIST	Sec. 2.12.1.18

ReturnValues		
Return Value	**Type**	**Meaning**
new_cons	pointer to LIST	New cons created.

Errors	
Error Name	**Reason for Error**
Error: Can't allocate a cons	Unable to allocate space for a new cons.

Calls	
Function	**Where Described**
heap_allocate	Sec. 2.14.2.1.1

Table 2.12-31: cons CSU [8.12.1.7.1]

2.12.1.7.2 ncons CSU

This CSU creates a new cons, with a cdr of nil.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
car	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
cons(car, NULL_PTR)	pointer to LIST	Function call to create a new cons.
Calls		
Function	**Where Described**	
cons	Sec. 2.12.1.7.1	

Table 2.12-32: ncons CSU [8.12.1.7.2]

2.12.1.7.3 push CSU

This CSU pushes an item onto the front of a list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
value	int	Standard
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
list	pointer to LIST	The resulting list.
Calls		
Function	**Where Described**	
cons	Sec. 2.12.1.7.1	

Table 2.12-33: push CSU [8.12.1.7.3]

2.12.1.7.4 free_list CSU

This CSU frees all of the cons on a list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list	pointer to LIST	Sec. 2.12.1.18
Calls		
Function	**Where Described**	
free_list	Sec. 2.12.1.7.4	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.12-34: free_list CSU [8.12.1.7.4]

2.12.1.7.5 remove_duplicates CSU

This CSU removes all duplicate items of a list, leaving only one instance of that item on the list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list	pointer to LIST	Sec. 2.12.1.18
Calls		
Function	**Where Described**	
heap_deallocate	Sec. 2.14.2.1.4	
remove_duplicates	Sec. 2.12.1.7.5	

Table 2.12-35: remove_duplicates CSU [8.12.1.7.5]

2.12.1.7.6 member CSU

This CSU determines if an item is a member of a list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
item	int	Standard
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Item is a member of the list.
FALSE	int	Item is not a member of the list.

Table 2.12-36: member CSU [8.12.1.7.6]

2.12.1.7.7 delete CSU

This CSU deletes an item from a list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
item	int	Standard
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
list	pointer to LIST	The resulting list.
Calls		
Function	**Where Described**	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.12-37: delete CSU [8.12.1.7.7]

2.12.1.7.8 nthcdr CSU

This CSU returns the portion of a list that starts at the nth cdr of the input list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
n	int	Standard
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
cons	pointer to LIST	The resulting list.

Table 2.12-38: nthcdr CSU [8.12.1.7.8]

2.12.1.7.9 nconc CSU

This CSU appends one list on the end of another list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list1	pointer to LIST	Sec. 2.12.1.18
list2	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
list1	pointer to LIST	list2 appended to list1.
list2	pointer to LIST	list1 empty.

Table 2.12-39: nconc CSU [8.12.1.7.9]

2.12.1.7.10 last_item CSU

This CSU returns the last item of a list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to LIST	List empty.
cons->car	pointer to LIST	Last list item.

Table 2.12-40: last_item CSU [8.12.1.7.10]

2.12.1.7.11 last_item_butlast CSU

This CSU returns the second to the last item of a list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_VAL	pointer to LIST	List empty.
list->car	pointer to LIST	Only one item in list; that item returned.
cons->car	pointer to LIST	Second from last item.

Table 2.12-41: last_item_butlast CSU [8.12.1.7.11]

2.12.1.7.12 delete_last_item CSU

This CSU deletes the last item of a list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to LIST	List was empty; or list had only one item, which was deleted and the space deallocated.
list	pointer to LIST	Last item deleted.
Calls		
Function	**Where Described**	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.12-42: delete_last_item CSU [8.12.1.7.12]

2.12.1.7.13 set_xor CSU

This CSU removes all duplicate items from two lists and returns a new list of all the items remaining in the two lists.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list1	pointer to LIST	Sec. 2.12.1.18
list2	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
list1	pointer to LIST	List 2 is null.
new_1	pointer to LIST	The resulting list.
Calls		
Function	**Where Described**	
copy_list	Sec. 2.12.1.7.14	
member	Sec. 2.12.1.7.6	
delete	Sec. 2.12.1.7.7	
nconc	Sec. 2.12.1.7.9	

Table 2.12-43: set_xor CSU [8.12.1.7.13]

2.12.1.7.14 copy_list CSU

This CSU copies a list, creating a new list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
new	pointer to LIST	The new list.
Calls		
Function	**Where Described**	
nconc	Sec. 2.12.1.7.9	
ncons	Sec. 2.12.1.7.2	

Table 2.12-44: copy_list CSU [8.12.1.7.14]

2.12.1.7.15 reverse CSU

This CSU returns a new list composed of the specified list in reverse order.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
new	pointer to LIST	The new list.
Calls		
Function	**Where Described**	
push	Sec. 2.12.1.7.3	

Table 2.12-45: reverse CSU [8.12.1.7.15]

2.12.1.7.16 length CSU

This CSU returns the number of items in a list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
i	int	Number of items in the list.

Table 2.12-46: length CSU [8.12.1.7.16]

2.12.1.8 water_avoidance.c CSC

/simnet/libsrc/libquad/water_avoidance.c

This CSU contains CSUs that accomplish finding a suitable route, eliminating any route through impassable water.

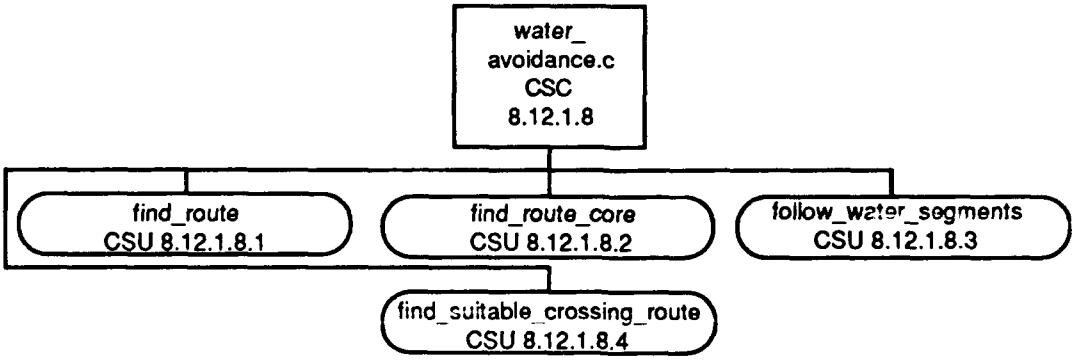


Figure 2.12-9: water_avoidance.c CSC Structure

2.12.1.8.1 find_route CSU

This CSU creates a new route that eliminates all unfordable water crossings along a given route.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
offset	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
new_route	pointer to LIST	New route found.
Calls		
Function	**Where Described**	
get_water_indecies	Sec. 2.12.1.4.1	
all_wide_segments_thru_water	Sec. 2.12.1.4.8	
free_list	Sec. 2.12.1.7.4	
find_route_core	Sec. 2.12.1.8.2	
nconc	Sec. 2.12.1.7.9	
FIRST	See Appendix A	
skirt_river_bend	Sec. 2.12.1.12.9	
skirt_lake	Sec. 2.12.1.12.7	
SECOND	See Appendix A	
heap_deallocate	Sec. 2.14.2.1.4	
print_route_list	Sec. 2.12.1.11.22	
last_item	Sec. 2.12.1.7.10	
push	Sec. 2.12.1.7.3	
get_quads_in_rt_point_region	Sec. 2.12.1.11.14	
final_relax_points	Sec. 2.12.1.11.12	
CDR	See Appendix A	
ncons	Sec. 2.12.1.7.2	

Table 2.12-47: find_route CSU [8.12.1.8.1]

2.12.1.8.2 find_route_core CSU

This CSU recursively finds points to cross unfordable water crossed by a given line segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
xy1	pointer to VECTOR_2D	Sec. 2.12.1.18
xy2	pointer to VECTOR_2D	Sec. 2.12.1.18
water_crossings	pointer to RIVER_SEG	Sec. 2.12.1.18
offset	INT32	Sec. 2.12.1.18
new_section	pointer to RIVER_SEG	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to LIST	No crossing exists.
new_section	pointer to LIST	Water crossing points.
Calls		
Function	**Where Described**	
FIRST	See Appendix A	
CDR	See Appendix A	
append_vector_2d_on_route	Sec. 2.12.1.11.19	
ODD_P	See Appendix A	
skirt_river_bend	Sec. 2.12.1.12.9	
skirt_lake	Sec. 2.12.1.12.7	
SECOND	See Appendix A	
intersection_direction	Sec. 2.12.1.11.2	
follow_water_segments	Sec. 2.12.1.8.3	
free_list	Sec. 2.12.1.7.4	
length	Sec. 2.12.1.7.16	
free_crossings	Sec. 2.12.1.11.20	
find_closer_crossing	Sec. 2.12.1.11.7	
push_vector_2d_on_points	Sec. 2.12.1.11.17	
crossing_location	Sec. 2.12.1.11.9	
get_quads_in_region	Sec. 2.12.1.11.13	
find_suitable_crossing_route	Sec. 2.12.1.8.4	

Table 2.12-48: find_route_core CSU [8.12.1.8.2]

2.12.1.8.3 follow_water_segments CSU

This CSU recursively follows the river network looking for bridges, fordable segments, or ends of river segments.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
int_index	int	Standard
water_index	int	Standard
direction	int	Standard
level	int	Standard
segment_list	pointer to LIST	Sec. 2.12.1.18
offset	int	Standard
water_crossings	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
water_crossings	pointer to LIST	The water crossings found.
Calls		
Function	**Where Described**	
member	Sec. 2.12.1.7.6	
nconc	Sec. 2.12.1.7.9	
ncons	Sec. 2.12.1.7.2	
push	Sec. 2.12.1.7.3	
heap_allocate	Sec. 2.14.2.1.1	
follow_water_segments	Sec. 2.12.1.8.3	
free_list	Sec. 2.12.1.7.4	
free_crossings	Sec. 2.12.1.11.20	

Table 2.12-49: follow_water_segments CSU [8.12.1.8.3]

2.12.1.8.4 find_suitable_crossing_route CSU

This CSU generates the route points necessary to read a water crossing point, skirting river bends or lakes.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
start	pointer to VECTOR_2D	Sec. 2.12.1.18
destination	pointer to VECTOR_2D	Sec. 2.12.1.18
next_crossing	pointer to RIVER_CROSSING	Sec. 2.12.1.18
last_crossing	pointer to RIVER_CROSSING	Sec. 2.12.1.18
rest_crossings	pointer to LIST	Sec. 2.12.1.18
prev_segment_list	pointer to LIST	Sec. 2.12.1.18
new_section	pointer to LIST	Sec. 2.12.1.18
direction	int	Standard
offset	int	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
new_section	pointer to LIST	Route points.
Calls		
Function	**Where Described**	
all_wide_segments_thru_water	Sec. 2.12.1.4.8	
FIRST	See Appendix A	
skirt_river	Sec. 2.12.1.12.1	
skirt_lake	Sec. 2.12.1.12.7	
SECOND	See Appendix A	
extend_crossing	Sec. 2.12.1.10.1	
rt_point_to_vector	Sec. 2.12.1.11.21	
reverse	Sec. 2.12.1.7.15	
set_xor	Sec. 2.12.1.7.13	
copy_list	Sec. 2.12.1.7.14	
nconc	Sec. 2.12.1.7.9	
last_item	Sec. 2.12.1.7.10	
find_suitable_crossing_route	Sec. 2.12.1.8.4	
CDR	See Appendix A	
free_list	Sec. 2.12.1.7.4	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.12-50: find_suitable_crossing_route CSU [8.12.1.8.4]

2.12.1.9 vector_2d.c CSC

/simnet/libsrc/libquad/vector_2d.c

This CSC contains CSUs that deal with two-dimensional vectors, each consisting of a single x,y pair.

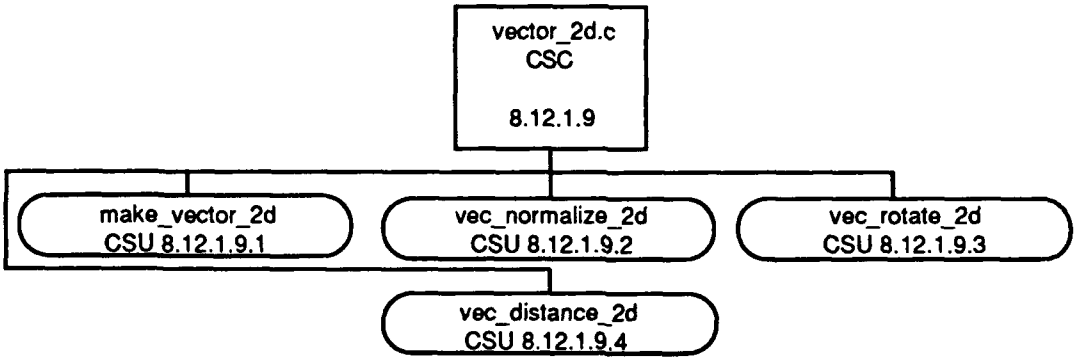


Figure 2.12-10: vector_2d.c CSC Structure

2.12.1.9.1 make_vector_2d CSU

This CSU allocates memory for a vector, if available, and sets the vector's values to the passed parameters.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x	double	Standard
y	double	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
vector	pointer to VECTOR_2D	Pointer to vector.
Errors		
Error Name	**Reason for Error**	
Can't allocate new vector	Unable to allocate memory for new vector.	
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	

Table 2.12-51: make_vector_2d CSU [8.12.1.9.1]

2.12.1.9.2 vec_normalize_2d CSU

This CSU normalizes the first input vector, and places the result in the second input vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
a	pointer to VECTOR_2D	Sec. 2.12.1.18
b	pointer to VECTOR_2D	Sec. 2.12.1.18

Table 2.12-52: vec_normalize_2d CSU [8.12.1.9.2]

2.12.1.9.3 vec_rotate_2d CSU

This CSU rotates vector *a* by *angle,* and places the result in vector *b.*

Parameters		
Parameters	**Type**	**Where Typedef Declared**
a	pointer to VECTOR_2D	Sec. 2.12.1.18
b	pointer to VECTOR_2D	Sec. 2.12.1.18
angle	double	Standard

Table 2.12-53: vec_rotate_2d CSU [8.12.1.9.3]

2.12.1.9.4 vec_distance_2d CSU

This CSU returns the distance between the end points of two normalized vectors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
a	pointer to VECTOR_2D	Sec. 2.12.1.18
b	pointer to VECTOR_2D	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
sqrt(dx*dx + dy*dy)	double	Square root of the sum of the squares of the vector differences.

Table 2.12-54: vec_distance_2d CSU [8.12.1.9.4]

2.12.1.10 extend_crossings.c CSC

/simnet/libsrc/libquad/extend_crossings.c

This CSC contains CSUs that deal with the extension of crossing routes.

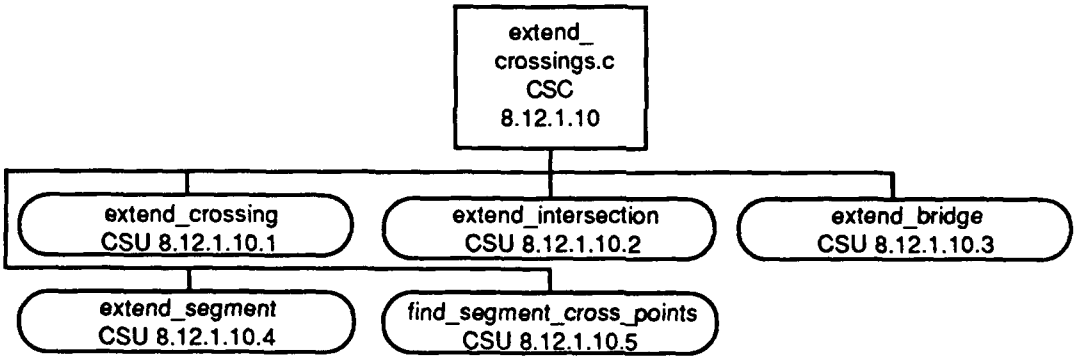


Figure 2.12-11: extend_crossings.c CSC Structure

2.12.1.10.1 extend_crossing CSU

This CSU calls the appropriate routine to extend a river crossing point, in order to facilitate crossing at that point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
crossing	pointer to RIVER_CROSSING	Sec. 2.12.1.18
direction	int	Standard
offset	int	Standard
extended_crossing	pointer to LIST	Sec. 2.12.1.18

ReturnValues		
Return Value	**Type**	**Meaning**
extended_crossing	pointer to LIST	Pointer to extended crossing type.
Calls		
Function	**Where Described**	
extend_bridge	Sec. 2.12.1.10.3	
extend_intersection	Sec. 2.12.1.10.2	
extend_segment	Sec. 2.12.1.10.4	

Table 2.12-55: extend_crossing CSU [8.12.1.10.1]

2.12.1.10.2 extend_intersection CSU

This CSU extends the end of a river segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
intersection	pointer to RIVER_INT	Sec. 2.12.1.18
direction	int	Standard
offset	int	Standard
extended_crossing	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
extended_crossing	pointer to LIST	The extended river segment.
Calls		
Function	**Where Described**	
find_next_point	Sec. 2.12.1.11.8	
VEC_SUB_2D	See Appendix A	
vec_normalize_2d	Sec. 2.12.1.9.2	
VEC_SCALE_2D	See Appendix A	
vec_rotate_2d	Sec. 2.12.1.9.3	
VEC_ADD_2D	See Appendix A	
push_vector_2d_on_route	Sec. 2.12.1.11.18	

Table 2.12-56: extend_intersection CSU [8.12.1.10.2]

2.12.1.10.3 extend_bridge CSU

This CSU extends a bridge crossing point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
bridge	pointer to BRIDGE	Sec. 2.12.1.18
direction	int	Standard
offset	int	Standard
segment_list	pointer to LIST	Sec. 2.12.1.18
extended_crossing	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
extended_crossing	pointer to LIST	The extended crossing point.
Calls		
Function	**Where Described**	
last_item	Sec. 2.12.1.7.10	
find_next_point	Sec. 2.12.1.11.8	
VEC_SUB_2D	See Appendix A	
vec_normalize_2d	Sec. 2.12.1.9.2	
VEC_SCALE_2D	Sec. 2.12.1.18	
find_first_vector	Sec. 2.12.1.11.5	
VEC_ADD_2D	See Appendix A	
push_vector_2d_on_route	Sec. 2.12.1.11.18	

Table 2.12-57: extend_bridge CSU [8.12.1.10.3]

2.12.1.10.4 extend_segment CSU

This CSU extends a fordable river segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
segment	pointer to RIVER_SEG	Sec. 2.12.1.18
segment_list	pointer to LIST	Sec. 2.12.1.18
direction	int	Standard
offset	int	Standard
extended_crossing	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
extended_crossing	pointer to LIST	The extended fordable river segment.

Calls	
Function	**Where Described**
find_segment_cross_points	Sec. 2.12.1.10.5
last_item_butlast	Sec. 2.12.1.7.11
VEC_SUB_2D	See Appendix A
vec_rotate_2d	Sec. 2.12.1.9.3
vec_normalize_2d	Sec. 2.12.1.9.2
VEC_SCALE_2D	See Appendix A
find_first_vector	Sec. 2.12.1.11.5
VEC_ADD_2D	See Appendix A
push_vector_2d_on_route	Sec. 2.12.1.11.18

Table 2.12-58: extend_segment CSU [8.12.1.10.4]

2.12.1.10.5 find_segment_cross_points CSU

This CSU finds two points that form a line perpendicular to a line segment at a distance offset from one end.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
segment	pointer to RIVER_SEG	Sec. 2.12.1.18
prev_segment_index	int	Standard
offset	int	Standard
cross1	pointer to VECTOR_2D	Sec. 2.12.1.18
cross2	pointer to VECTOR_2D	Sec. 2.12.1.18

Calls	
Function	**Where Described**
distance	SEC. 2.12.1.11.1
VEC_SUB_2D	See Appendix A
VEC_SCALE_2D	See Appendix A

Table 2.12-59: find_segment_cross_points CSU [8.12.1.10.5]

2.12.1.11 water_utilities.c CSC

/simnet/libsrc/libquad/water_utilities.c

This CSC contains CSUs dealing with water crossing and intersection.

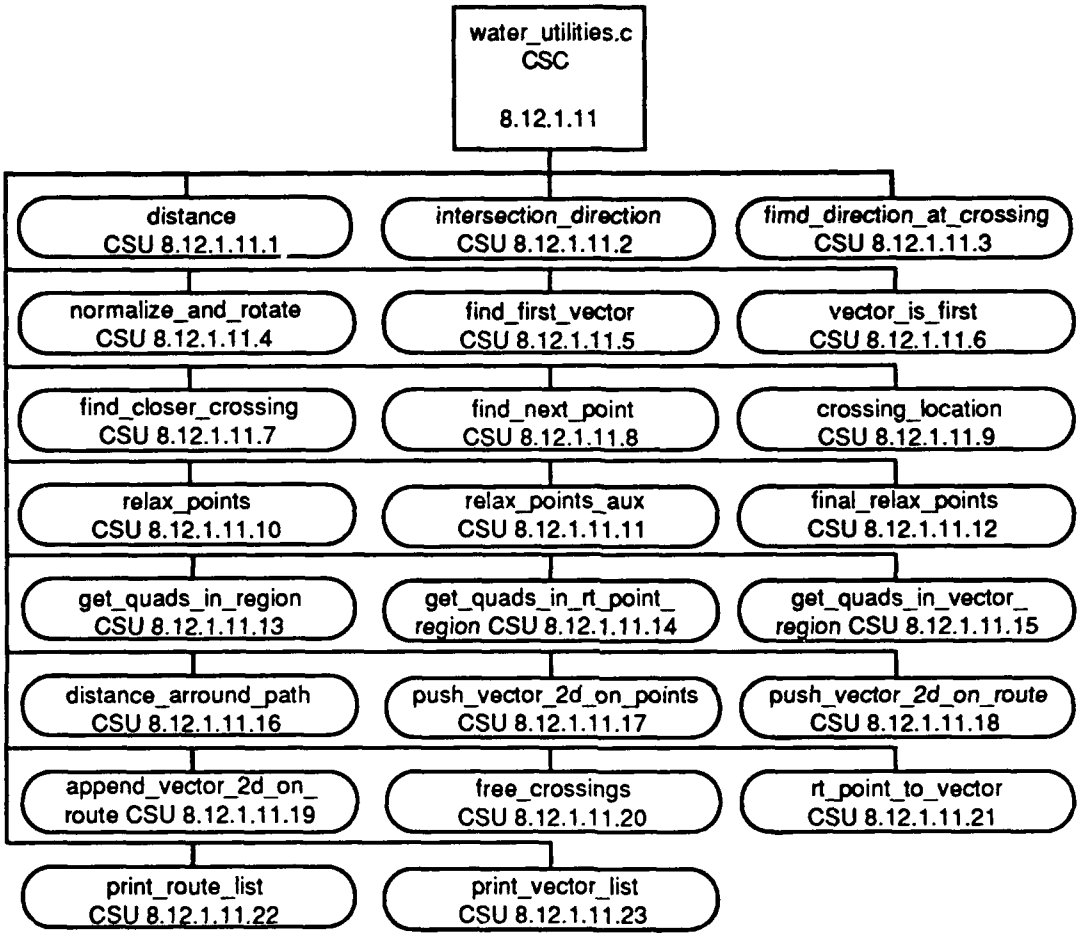


Figure 2.12-12: water_utilities.c CSC Structure

2.12.1.11.1 distance CSU

This CSU calculates the distance between two points.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18

ReturnValues		
Return Value	**Type**	**Meaning**
sqrt(dx*dx + dy*dy)	double	Square root of the sum of the squares of the vector differences.

Table 2.12-60: distance CSU [8.12.1.11.1]

2.12.1.11.2 intersection_direction CSU

This CSU determines whether the tributaries of a river should be searched in clockwise or counterclockwise order.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
origin	pointer to VECTOR_2D	Sec. 2.12.1.18
water	pointer to WATER_INT	Sec. 2.12.1.18
int_index	int	Standard
Calls		
Function	**Where Described**	
find_direction_at_crossing	Sec. 2.12.1.11.3	

Table 2.12-61: intersection_direction CSU [8.12.1.11.2]

2.12.1.11.3 find_direction_at_crossing CSU

This CSU determines whether a set of points form a clockwise or a counterclockwise angle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to INT32	Sec. 2.12.1.18
num_pts	int	Standard
reversed	int	Standard
origin	pointer to VECTOR_2D	Sec. 2.12.1.18
water	pointer to WATER_INT	Sec. 2.12.1.18
pt_direction	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
CLOCKWISE	int	Direction
ANTICLOCKWISE	int	Direction
Calls		
Function	**Where Described**	
VEC_SUB_2D	See Appendix A	
normalize_and_rotate	Sec. 2.12.1.11.4	
VEC_ANGLE_2D	See Appendix A	

Table 2.12-62: find_direction_at_crossing CSU [8.12.1.11.3]

2.12.1.11.4 normalize_and_rotate CSU

This CSU normalizes a vector and rotates it.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
origin	pointer to VECTOR_2D	Sec. 2.12.1.18
a	pointer to VECTOR_2D	Sec. 2.12.1.18
b	pointer to VECTOR_2D	Sec. 2.12.1.18
result	pointer to VECTOR_2D	Sec. 2.12.1.18
Calls		
Function	**Where Described**	
VEC_SUB_2D	See Appendix A	
vec_normalize_2d	Sec. 2.12.1.9.2	
VEC_ANGLE_2D	See Appendix A	
vec_rotate_2d	Sec. 2.12.1.9.3	

Table 2.12-63: normalize_and_rotate CSU [8.12.1.11.4]

2.12.1.11.5 find_first_vector CSU

This CSU returns the first vector found by rotating the origin vector in the given direction.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
origin	pointer to VECTOR_2D	Sec. 2.12.1.18
a	pointer to VECTOR_2D	Sec. 2.12.1.18
b	pointer to VECTOR_2D	Sec. 2.12.1.18
direction	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
vector_is_first(...)	pointer to VECTOR_2D	Found vector.
Calls		
Function	**Where Described**	
VEC_ANGLE_2D	See Appendix A	
vec_rotate_2d	Sec. 2.12.1.9.3	
vector_is_first	Sec. 2.12.1.11.6	

Table 2.12-64: find_first_vector CSU [8.12.1.11.5]

2.12.1.11.6 vector_is_first CSU

This CSU determines whether the first angle is smaller than the second angle in the direction given.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
angle1	double	Standard
angle2	double	Standard
direction	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	First angle is smaller than the second angle.
FALSE	int	First angle is not smaller than the second angle.

Table 2.12-65: vector_is_first CSU [8.12.1.11.6]

2.12.1.11.7 find_closer_crossing CSU

This CSU calculates the distances from a point to two crossing points, and returns the closest crossing point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
xy	pointer to VECTOR_2D	Sec. 2.12.1.18
crossings_1	pointer to LIST	Sec. 2.12.1.18
crossings_2	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
crossings_1	pointer to LIST	The first crossing point is the closest .
crossings_2	pointer to LIST	The second crossing point is the closest .
Calls		
Function	**Where Described**	
crossing_location	Sec. 2.12.1.11.9	
FIRST	See Appendix A	
vec_distance_2d	Sec. 2.12.1.9.4	

Table 2.12-66: find_closer_crossing CSU [8.12.1.11.7]

2.12.1.11.8 find_next_point CSU

This CSU finds the next point in a point list from an intersection point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
int_vec	pointer to VECTOR_2D	Sec. 2.12.1.18
return_vec	pointer to VECTOR_2D	Sec. 2.12.1.18
points	pointer to INT32	Sec. 2.12.1.18
num_pts	int	Standard

Table 2.12-67: find_next_point CSU [8.12.1.11.8]

2.12.1.11.9 crossing_location CSU

This CSU returns the location of a water crossing point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
crossing	pointer to RIVER_CROSSING	Sec. 2.12.1.18
vector	pointer to VECTOR_2D	Sec. 2.12.1.18
Calls		
Function	**Where Described**	
last_item	Sec. 2.12.1.7.10	
length	Sec. 2.12.1.7.16	
last_item_butlast	Sec. 2.12.1.7.11	

Table 2.12-68: crossing_location CSU [8.12.1.11.9]

2.12.1.11.10 relax_points CSU

This CSU eliminates route points such that the resulting route does not cross any unfordable water.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
start	pointer to VECTOR_2D	Sec. 2.12.1.18
destination	pointer to VECTOR_2D	Sec. 2.12.1.18
river_points	pointer to LIST	Sec. 2.12.1.18
offset	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
river_points	pointer to LIST	*Resulting route.*

Calls	
Function	**Where Described**
nconc	Sec. 2.12.1.7.9
ncons	Sec. 2.12.1.7.2
get_quads_in_vector_region	Sec. 2.12.1.11.15
delete_last_item	Sec. 2.12.1.7.12
relax_points_aux	Sec. 2.12.1.11.11
CDR	See Appendix A
last_item	Sec. 2.12.1.7.10

Table 2.12-69: relax_points CSU [8.12.1.11.10]

2.12.1.11.11 relax_points_aux CSU

This CSU removes points from a point list, checking if the resulting list crosses any unfordable water.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
start	pointer to VECTOR_2D	Sec. 2.12.1.18
points	pointer to LIST	Sec. 2.12.1.18
offset	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
ncons(start)	pointer to LIST	The list is empty.
new_points	pointer to LIST	The resulting point list.
Calls		
Function	**Where Described**	
ncons	Sec. 2.12.1.7.2	
reverse	Sec. 2.12.1.7.15	
any_wide_segment_thru_water	Sec. 2.12.1.4.3	
heap_deallocate	Sec. 2.14.2.1.4	
push	Sec. 2.12.1.7.3	
free_list	Sec. 2.12.1.7.4	
nconc	Sec. 2.12.1.7.9	
relax_points	Sec. 2.12.1.11.10	

Table 2.12-70: relax_points_aux CSU [8.12.1.11.11]

2.12.1.11.12 final_relax_points CSU

This CSU removes points from a point list, checking if the resulting list crosses any unfordable water. It does not remove bridge points, however.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
start	pointer to VECTOR_2D	Sec. 2.12.1.18
points	pointer to LIST	Sec. 2.12.1.18
offset	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
ncons(start)	pointer to LIST	The list is empty.
new_points	pointer to LIST	The resulting point list.
Calls		
Function	**Where Described**	
ncons	Sec. 2.12.1.7.2	
reverse	Sec. 2.12.1.7.15	
any_wide_segment_thru_water	Sec. 2.12.1.4.3	
heap_deallocate	Sec. 2.14.2.1.4	
push	Sec. 2.12.1.7.3	
free_list	Sec. 2.12.1.7.4	
nconc	Sec. 2.12.1.7.9	
final_relax_points	Sec. 2.12.1.11.12	
CDR	See Appendix A	

Table 2.12-71: final_relax_points CSU [8.12.1.11.12]

2.12.1.11.13 get_quads_in_region CSU

This CSU returns all of the water indices within a bounding box around a river segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
get_water_indecies(...)	pointer to LIST	Resulting water indices.
Calls		
Function	**Where Described**	
CDDR	See Appendix A	
get_water_indecies	Sec. 2.12.1.4.1	

Table 2.12-72: get_quads_in_region CSU [8.12.1.11.13]

2.12.1.11.14 get_quads_in_rt_point_region CSU

This CSU returns all of the water indices within a bounding box around a route.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
rt_points	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
get_water_indecies(...)	pointer to LIST	Resulting water indices.
Calls		
Function	**Where Described**	
get_water_indecies	Sec. 2.12.1.4.1	

Table 2.12-73: get_quads_in_rt_point_region CSU [8.12.1.11.14]

2.12.1.11.15 get_quads_in_vector_region CSU

This CSU returns all of the water indices within a bounding box around a list of connected vectors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vectors	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
get_water_indecies(...)	pointer to LIST	Resulting water indices.
Calls		
Function	**Where Described**	
get_water_indecies	Sec. 2.12.1.4.1	

Table 2.12-74: get_quads_in_vector_region CSU [8.12.1.11.15]

2.12.1.11.16 distance_around_path CSU

This CSU returns the distance of all of the segments in a list of connected vectors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vectors	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
0	int	Empty list.
sum/count	int	Distance.

Calls	
Function	**Where Described**
vec_distance_2d	Sec. 2.12.1.9.4
FIRST	See Appendix A
SECOND	See Appendix A

Table 2.12-75: distance_around_path CSU [8.12.1.11.16]

2.12.1.11.17 push_vector_2d_on_points CSU

This CSU converts a vector to an x-y point and pushes it onto a point list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vector	pointer to VECTOR_2D	Sec. 2.12.1.18
points	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
points	pointer to LIST	Point list.
Calls		
Function	**Where Described**	
push	Sec. 2.12.1.7.3	

Table 2.12-76: push_vector_2d_on_points CSU [8.12.1.11.17]

2.12.1.11.18 push_vector_2d_on_route CSU

This CSU converts a vector to a route point and pushes it onto a route.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vector	pointer to VECTOR_2D	Sec. 2.12.1.18
type	int	Standard
route	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
route	pointer to LIST	Resulting route.
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
push	Sec. 2.12.1.7.3	

Table 2.12-77: push_vector_2d_on_route CSU [8.12.1.11.18]

2.12.1.11.19 append_vector_2d_on_route CSU

This CSU converts a vector to a route point and appends it to the end of a route.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vector	pointer to VECTOR_2D	Sec. 2.12.1.18
type	int	Standard
route	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
route	pointer to LIST	Resulting route.
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
nconc	Sec. 2.12.1.7.9	
ncons	Sec. 2.12.1.7.2	

Table 2.12-78: append_vector_2d_on_route CSU [8.12.1.11.19]

2.12.1.11.20 free_crossings CSU

This CSU frees a list of river crossings.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
crossings	pointer to LIST	Sec. 2.12.1.18
Calls		
Function	**Where Described**	
free_list	Sec. 2.12.1.7.4	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.12-79: free_crossings CSU [8.12.1.11.20]

2.12.1.11.21 rt_point_to_vector CSU

This CSU converts a route point to a vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
rt_point	pointer to ROUTE_PT	Sec. 2.12.1.18
vector	pointer to VECTOR_2D	Sec. 2.12.1.18

Table 2.12-80: rt_point_to_vector CSU [8.12.1.11.21]

2.12.1.11.22 print_route_list CSU

This CSU prints a list of route points.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to LIST	Sec. 2.12.1.18

Table 2.12-81: print_route_list CSU [8.12.1.11.22]

2.12.1.11.23 print_vector_list CSU

This CSU prints a list of vectors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
route	pointer to LIST	Sec. 2.12.1.18

Table 2.12-82: print_vector_list CSU [8.12.1.11.23]

2.12.1.12 skirt_water.c CSC

/simnet/libsrc/libquad/skirt_water.c

This CSC contains CSUs that deal with traveling around impassable water.

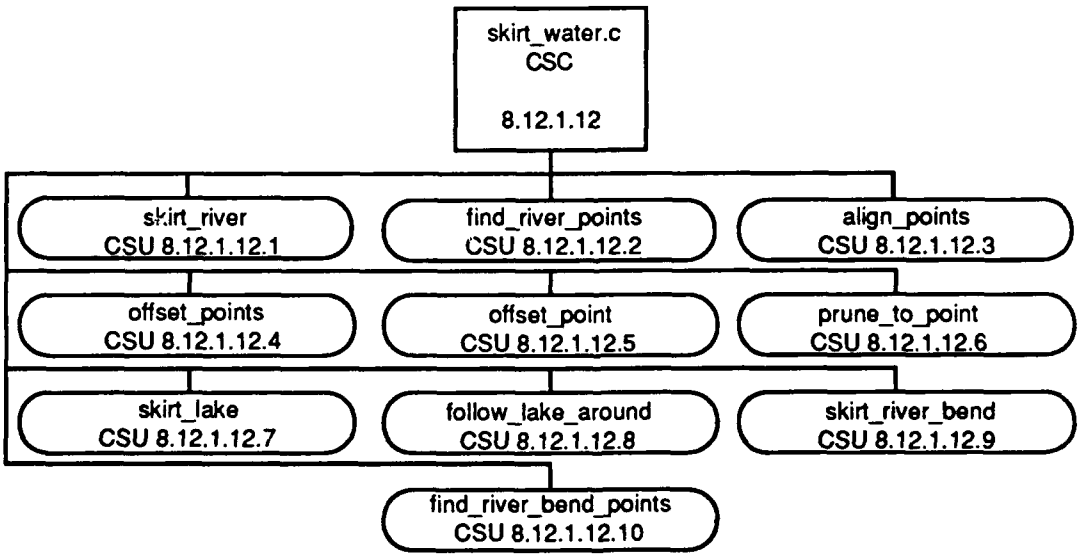


Figure 2.12-13: skirt_water.c CSC Structure

2.12.1.12.1 skirt_river CSU

This CSU calculates the route to skirt a river segment by following a route parallel to the river.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
start	pointer to VECTOR_2D	Sec. 2.12.1.18
destination	pointer to VECTOR_2D	Sec. 2.12.1.18
crossing	pointer to RIVER_CROSSING	Sec. 2.12.1.18
last_crossing	pointer to RIVER_CROSSING	Sec. 2.12.1.18
segment_list	pointer to LIST	Sec. 2.12.1.18
direction	int	Standard
offset	int	Standard
new_section	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
new_section	pointer to LIST	Resulting route.
Calls		
Function	**Where Described**	
find_river_points	Sec. 2.12.1.12.2	
prune_to_point	Sec. 2.12.1.12.6	
length	Sec. 2.12.1.7.16	
offset_points	Sec. 2.12.1.12.4	
ncons	Sec. 2.12.1.7.2	
nconc	Sec. 2.12.1.7.9	
relax_points	Sec. 2.12.1.11.10	
append_vector_2d_on_route	Sec. 2.12.1.11.19	
heap_deallocate	Sec. 2.14.2.1.4	
free_list	Sec. 2.12.1.7.4	

Table 2.12-83: skirt_river CSU [8.12.1.12.1]

2.12.1.12.2 find_river_points CSU

This CSU finds the river segment points between a line segment intersection and a river intersection.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
segment_list	pointer to LIST	Sec. 2.12.1.18
crossing	pointer to RIVER_CROSSING	Sec. 2.12.1.18
last_crossing	pointer to RIVER_CROSSING	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
align_points(...)	pointer to LIST	Empty river segment list.
river_points	pointer to LIST	Resulting river segment points

Calls	
Function	**Where Described**
FIRST	See Appendix A
CDR	See Appendix A
crossing_location	Sec. 2.12.1.11.9
align_points	Sec. 2.12.1.12.3
SECOND	See Appendix A
nconc	Sec. 2.12.1.7.9

Table 2.12-84: find_river_points CSU [8.12.1.12.2]

2.12.1.12.3 align_points CSU

This CSU returns a list of vectors, generated from a point list, that point in a given direction.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to INT32	Sec. 2.12.1.18
num_pts	int	Standard
loc_vec	pointer to VECTOR_2D	Sec. 2.12.1.18
direction	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to LIST	No vectors generated.
river_points	pointer to LIST	Resulting list.
Calls		
Function	**Where Described**	
push	Sec. 2.12.1.7.3	
make_vector_2d	Sec. 2.12.1.9.1	

Table 2.12-85: align_points CSU [8.12.1.12.3]

2.12.1.12.4 offset_points CSU

This CSU calculates a set of points that are offset from the input points, based on the input direction.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to LIST	Sec. 2.12.1.18
origin	pointer to VECTOR_2D	Sec. 2.12.1.18
offset	int	Standard
direction	int	Standard

Calls	
Function	**Where Described**
FIRST	See Appendix A
SECOND	See Appendix A
VEC_SUB_2D	See Appendix A
VEC_ADD_2D	See Appendix A
offset_point	Sec. 2.12.1.12.5

Table 2.12-86: offset_points CSU [8.12.1.12.4]

2.12.1.12.5 offset_point CSU

This CSU offsets a point by calculating a perpendicular vector from that point, based on the preceding and following points.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
p1	pointer to VECTOR_2D	Sec. 2.12.1.18
p2	pointer to VECTOR_2D	Sec. 2.12.1.18
p3	pointer to VECTOR_2D	Sec. 2.12.1.18
direction	int	Standard
offset	int	Standard

Calls	
Function	**Where Described**
VEC_SUB_2D	See Appendix A
vec_normalize_2d	Sec. 2.12.1.9.2
VEC_ANGLE_2D	See Appendix A
vec_rotate_2d	Sec. 2.12.1.9.3
find_first_vector	Sec. 2.12.1.11.5
VEC_SCALE_2D	See Appendix A
VEC_ADD_2D	See Appendix A

Table 2.12-87: offset_point CSU [8.12.1.12.5]

2.12.1.12.6 prune_to_point CSU

This CSU returns a list of points, starting with the point closest to the origin point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to LIST	Sec. 2.12.1.18
origin	pointer to VECTOR_2D	Sec. 2.12.1.18
direction	int	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
points	pointer to LIST	Resulting list of points.

Calls	
Function	**Where Described**
vec_distance_2d	Sec. 2.12.1.9.4
nthcdr	Sec. 2.12.1.7.8
heap_deallocate	Sec. 2.14.2.1.4

Table 2.12-88: prune_to_point CSU [8.12.1.12.6]

2.12.1.12.7 skirt_lake CSU

This CSU calculates a route around a lake.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
start	pointer to VECTOR_2D	Sec. 2.12.1.18
lake_list	pointer to WATER_INT	Sec. 2.12.1.18
next_list	pointer to WATER_INT	Sec. 2.12.1.18
offset	int	Standard
new_section	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to LIST	There is no lake.
new_section	pointer to LIST	Resulting route.
Calls		
Function	**Where Described**	
offset_point	Sec. 2.12.1.12.5	
follow_lake_around	Sec. 2.12.1.12.8	
distance_around_path	Sec. 2.12.1.11.16	
heap_deallocate	Sec. 2.14.2.1.4	
free_list	Sec. 2.12.1.7.4	
append_vector_2d_on_route	Sec. 2.12.1.11.19	

Table 2.12-89: skirt_lake CSU [8.12.1.12.7]

2.12.1.12.8 follow_lake_around CSU

This CSU generates the route points to follow around a lake.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
start	pointer to VECTOR_2D	Sec. 2.12.1.18
dest	pointer to VECTOR_2D	Sec. 2.12.1.18
lake	pointer to LAKE	Sec. 2.12.1.18
min_start	int	Standard
min_dest	int	Standard
increment	int	Standard
offset	int	Standard

ReturnValues		
Return Value	**Type**	**Meaning**
new_points	pointer to LIST	Resulting list of route points.
Calls		
Function	**Where Described**	
nconc	Sec. 2.12.1.7.9	
ncons	Sec. 2.12.1.7.2	
make_vector_2d	Sec. 2.12.1.9.1	
offset_points	Sec. 2.12.1.12.4	
relax_points	Sec. 2.12.1.11.10	

Table 2.12-90: follow_lake_around CSU [8.12.1.12.8]

2.12.1.12.9 skirt_river_bend CSU

This CSU calculates a route around a river bend.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
start	pointer to VECTOR_2D	Sec. 2.12.1.18
destination	pointer to VECTOR_2D	Sec. 2.12.1.18
water_crossings	pointer to LIST	Sec. 2.12.1.18
new_section	pointer to LIST	Sec. 2.12.1.18
offset	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
new_section	pointer to LIST	Resulting route.
Calls		
Function	**Where Described**	
FIRST	See Appendix A	
reverse	Sec. 2.12.1.7.15	
free_list	Sec. 2.12.1.7.4	
find_river_bend_points	Sec. 2.12.1.12.10	
find_direction_at_crossing	Sec. 2.12.1.11.3	
prune_to_point	Sec. 2.12.1.12.6	
length	Sec. 2.12.1.7.16	
offset_points	Sec. 2.12.1.12.4	
nconc	Sec. 2.12.1.7.9	
ncons	Sec. 2.12.1.7.2	
relax_points	Sec. 2.12.1.11.10	
append_vector_2d_on_route	Sec. 2.12.1.11.19	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.12-91: skirt_river_bend CSU [8.12.1.12.9]

2.12.1.12.10 find_river_bend_points CSU

This CSU returns the list of river segment points between two intersections.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
river_seg	pointer to RIVER_SEG	Sec. 2.12.1.18
first_crossing	pointer to WATER_INT	Sec. 2.12.1.18
last_crossing	pointer to WATER_INT	Sec. 2.12.1.18
reversed	pointer to int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
river_points	pointer to LIST	Resulting list of river segment points.
Calls		
Function	**Where Described**	
push	Sec. 2.12.1.7.3	
make_vector_2d	Sec. 2.12.1.9.1	

Table 2.12-92: find_river_bend_points CSU [8.12.1.12.10]

2.12.1.13 minimum_clip.c CSC

/simnet/libsrc/libquad/minimum_clip.c

This CSC contains the CSUs necessary to determine if a line segment intersects a box. In addition, it contains macros, defined in Appendix A, used by these CSUs.

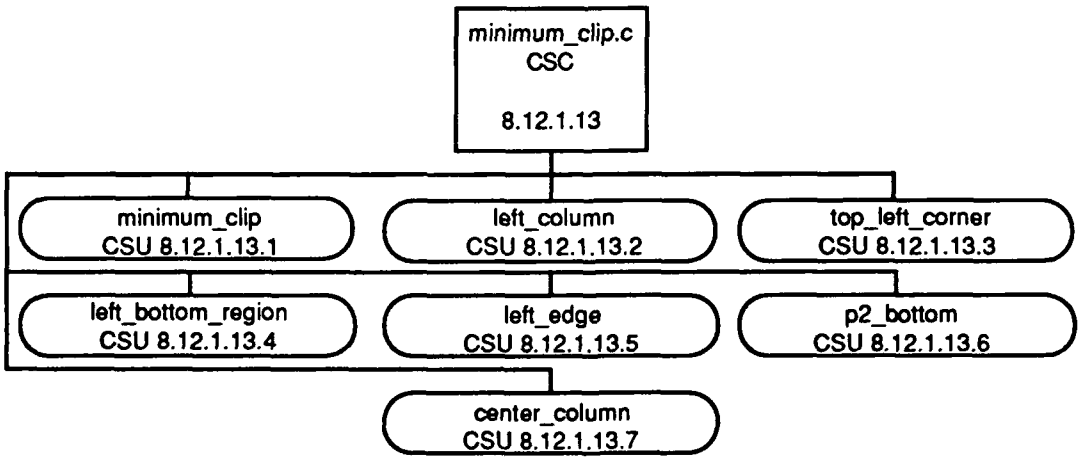


Figure 2.12-14: minimum_clip.c CSC Structure

2.12.1.13.1 minimum_clip CSU

This CSU determines whether a line segment intersects an area.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x_left	INT32	Sec. 2.12.1.18
y_top	INT32	Sec. 2.12.1.18
x_right	INT32	Sec. 2.12.1.18
y_bottom	INT32	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
display=TRUE	int	Line segment intersects area.
display=FALSE	int	Line segment does not intersect area.
Calls		
Function	**Where Described**	
left_column	Sec. 2.12.1.13.2	
rotate_180_c	See Appendix A	
center_column	Sec. 2.12.1.13.7	

Table 2.12-93: minimum_clip CSU [8.12.1.13.1]

2.12.1.13.2 left_column CSU

This CSU determines whether a line segment intersects an area if one point is beyond the left boundary.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x_left	INT32	Sec. 2.12.1.18
y_top	INT32	Sec. 2.12.1.18
x_right	INT32	Sec. 2.12.1.18
y_bottom	INT32	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
Calls		
Function	**Where Described**	
top_left_corner	Sec. 2.12.1.13.3	
reflect_x_axis	See Appendix A	
left_edge	Sec. 2.12.1.13.5	

Table 2.12-94: left_column CSU [8.12.1.13.2]

2.12.1.13.3 top_left_corner CSU

This CSU determines an intersection if one point is in the top left corner and the other point is not beyond the left boundary.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x_left	INT32	Sec. 2.12.1.18
y_top	INT32	Sec. 2.12.1.18
x_right	INT32	Sec. 2.12.1.18
y_bottom	INT32	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18

Calls	
Function	**Where Described**
left_bottom_region	Sec. 2.12.1.13.4
reflect_x_minus_y	See Appendix A

Table 2.12-95: top_left_corner CSU [8.12.1.13.3]

2.12.1.13.4 left_bottom_region CSU

This CSU determines an intersection if one point is to the right of the vector from the other point to the top left corner.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x_left	INT32	Sec. 2.12.1.18
y_top	INT32	Sec. 2.12.1.18
x_right	INT32	Sec. 2.12.1.18
y_bottom	INT32	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
rel_x2	double	Standard
rel_y2	double	Standard
left_product	double	Standard

Table 2.12-96: left_bottom_region CSU [8.12.1.13.4]

2.12.1.13.5 left_edge CSU

This CSU determines an intersection if one point is in the left edge region.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x_left	INT32	Sec. 2.12.1.18
y_top	INT32	Sec. 2.12.1.18
x_right	INT32	Sec. 2.12.1.18
y_bottom	INT32	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
Calls		
Function	**Where Described**	
p2_bottom	Sec. 2.12.1.13.6	
reflect_x_axis	See Appendix A	

Table 2.12-97: left_edge CSU [8.12.1.13.5]

2.12.1.13.6 p2_bottom CSU

This CSU determines an intersection if one point is in the left edge region, and the other point is not beyond the left boundary and is beyond the bottom boundary.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x_left	INT32	Sec. 2.12.1.18
y_top	INT32	Sec. 2.12.1.18
x_right	INT32	Sec. 2.12.1.18
y_bottom	INT32	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18

Table 2.12-98: p2_bottom CSU [8.12.1.13.6]

2.12.1.13.7 center_column CSU

This CSU determines an intersection if one point is between the left and right boundaries.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x_left	INT32	Sec. 2.12.1.18
y_top	INT32	Sec. 2.12.1.18
x_right	INT32	Sec. 2.12.1.18
y_bottom	INT32	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18

Calls	
Function	**Where Described**
rotate_270_c	See Apendix A
left_edge	Sec. 2.12.1.13.5
rotate_90_c	See Apendix A

Table 2.12-99: center_column CSU [8.12.1.13.7]

2.12.1.14 intersection.c CSC

/simnet/libsrc/libquad/intersection.c

This CSC contains CSUs that deal with intersections.

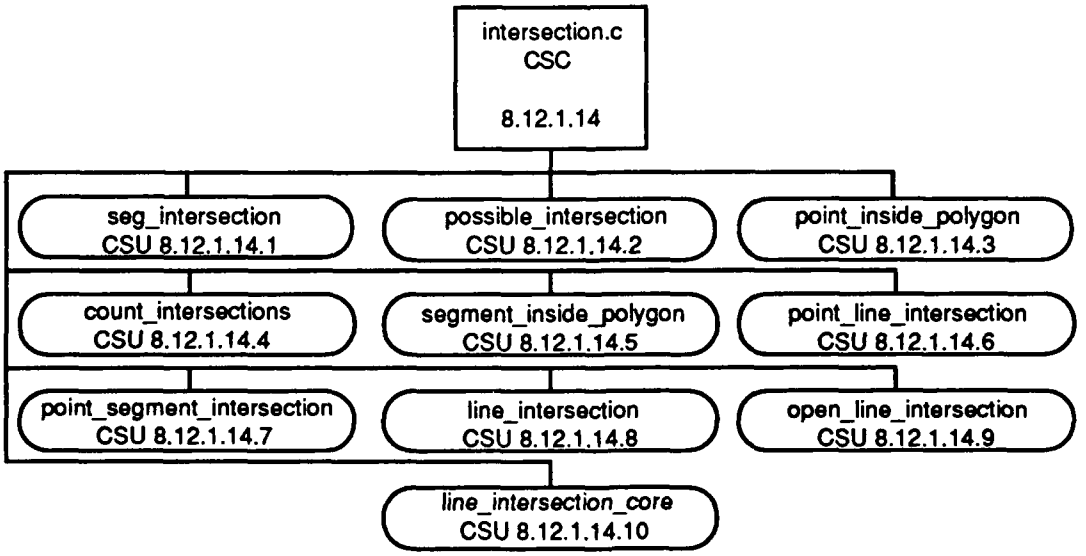


Figure 2.12-15: intersection.c CSC Structure

2.12.1.14.1 seg_intersection CSU

This CSU determines whether a line segment intersects a point list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to INT32	Sec. 2.12.1.18
num_pts	INT16	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Line intersects point list.
FALSE	int	Line does not intersect point list.
Calls		
Function	**Where Described**	
line_intersection	Sec. 2.12.1.14.8	

Table 2.12-100: seg_intersection CSU [8.12.1.14.1]

2.12.1.14.2 possible_intersection CSU

This CSU determines whether a line segment intersects an area.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x_min	INT32	Sec. 2.12.1.18
y_min	INT32	Sec. 2.12.1.18
x_max	INT32	Sec. 2.12.1.18
y_max	INT32	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
minimum_clip(...)	int	Function call to determine if line intersects the area.
Calls		
Function	**Where Described**	
minimum_clip	Sec. 2.12.1.13.1	

Table 2.12-101: possible_intersection CSU [8.12.1.14.2]

2.12.1.14.3 point_inside_polygon CSU

This CSU determines whether a point is inside a polygon.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to INT32	Sec. 2.12.1.18
num_pts	INT16	Sec. 2.12.1.18
x	INT32	Sec. 2.12.1.18
y	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Point is inside polygon.
ODD_P(...) = 1	int	Point is inside polygon.
ODD_P(...) = 0	int	Point is not inside polygon.
Calls		
Function	**Where Described**	
count_intersections	Sec. 2.12.1.14.4	
ODD_P	See Appendix A	

Table 2.12-102: point_inside_polygon CSU [8.12.1.14.3]

2.12.1.14.4 count_intersections CSU

This CSU counts the number of times a line segment intersects a point list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to INT32	Sec. 2.12.1.18
num_pts	INT16	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
-1	int	Line does not intersect point list.
sum	int	Number of times line intersects point list.
Calls		
Function	**Where Described**	
line_intersection	Sec. 2.12.1.14.8	
open_line_intersection	Sec. 2.12.1.14.9	

Table 2.12-103: count_intersections CSU [8.12.1.14.4]

2.12.1.14.5 segment_inside_polygon CSU

This CSU determines whether both ends of a line segment are inside a polygon.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to INT32	Sec. 2.12.1.18
num_pts	INT16	Sec. 2.12.1.18
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
point_inside_polygon(...) && point_inside_polygon(...) = TRUE	int	Both ends of line are inside polygon.
point_inside_polygon(...) && point_inside_polygon(...) = FALSE	int	Both ends of line are not inside polygon.
Calls		
Function	**Where Described**	
point_inside_polygon	Sec. 2.12.1.14.3	

Table 2.12-104: segment_inside_polygon CSU [8.12.1.14.5]

2.12.1.14.6 point_line_intersection CSU

This CSU calculates the perpendicular intersection of a point with a line segment.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
p1	INT32	Sec. 2.12.1.18
p2	INT32	Sec. 2.12.1.18
ix	pointer to INT32	Sec. 2.12.1.18
iy	pointer to INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
line_intersection_core(...)	int	Intersection point.

Calls	
Function	**Where Described**
VEC_SUB_2D	See Appendix A
vec_normalize_2d	Sec. 2.12.1.9.2
vec_rotate_2d	Sec. 2.12.1.9.3
VEC_ADD_2D	See Appendix A
line_intersection_core	Sec. 2.12.1.14.10

Table 2.12-105: point_line_intersection CSU [8.12.1.14.6]

2.12.1.14.7 point_segment_intersection CSU

This CSU calculates the perpendicular intersection of a point with a point list.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to INT32	Sec. 2.12.1.18
num_points	INT32	Sec. 2.12.1.18
p1	INT32	Sec. 2.12.1.18
p2	INT32	Sec. 2.12.1.18
px	pointer to INT32	Sec. 2.12.1.18
py	pointer to INT32	Sec. 2.12.1.18

Calls	
Function	**Where Described**
point_line_intersection	Sec. 2.12.1.14.6
distance	Sec. 2.12.1.11.1

Table 2.12-106: point_segment_intersection CSU [8.12.1.14.7]

2.12.1.14.8 line_intersection CSU

This CSU calculates the intersection point of two line segments.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
x3	INT32	Sec. 2.12.1.18
y3	INT32	Sec. 2.12.1.18
x4	INT32	Sec. 2.12.1.18
y4	INT32	Sec. 2.12.1.18
ix	pointer to INT32	Sec. 2.12.1.18
iy	pointer to INT32	Sec. 2.12.1.18

ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Lines intersect.
FALSE	int	Lines do not intersect.
Calls		
Function	**Where Described**	
line_intersection_core	Sec. 2.12.1.14.10	

Table 2.12-107: line_intersection CSU [8.12.1.14.8]

2.12.1.14.9 open_line_intersection CSU

This CSU calculates the intersection point of two line segments, but does not consider intersections at the end points of either segment as valid.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
x3	INT32	Sec. 2.12.1.18
y3	INT32	Sec. 2.12.1.18
x4	INT32	Sec. 2.12.1.18
y4	INT32	Sec. 2.12.1.18
ix	pointer to INT32	Sec. 2.12.1.18
iy	pointer to INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Lines intersect.
FALSE	int	Lines do not intersect.
Calls		
Function	**Where Described**	
line_intersection_core	Sec. 2.12.1.14.10	

Table 2.12-108: open_line_intersection CSU [8.12.1.14.9]

2.12.1.14.10 line_intersection_core CSU

This CSU calculates the location of the intersection point of two line segments.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
x3	INT32	Sec. 2.12.1.18
y3	INT32	Sec. 2.12.1.18
x4	INT32	Sec. 2.12.1.18
y4	INT32	Sec. 2.12.1.18
ix	pointer to INT32	Sec. 2.12.1.18
iy	pointer to INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Lines intersect.
FALSE	int	Lines do not intersect.

Table 2.12-109: line_intersection_core CSU [8.12.1.14.10]

2.12.1.15 roads.c CSC

/simnet/libsrc/libquad/roads.c

This CSC contains CSUs that deal with roads.

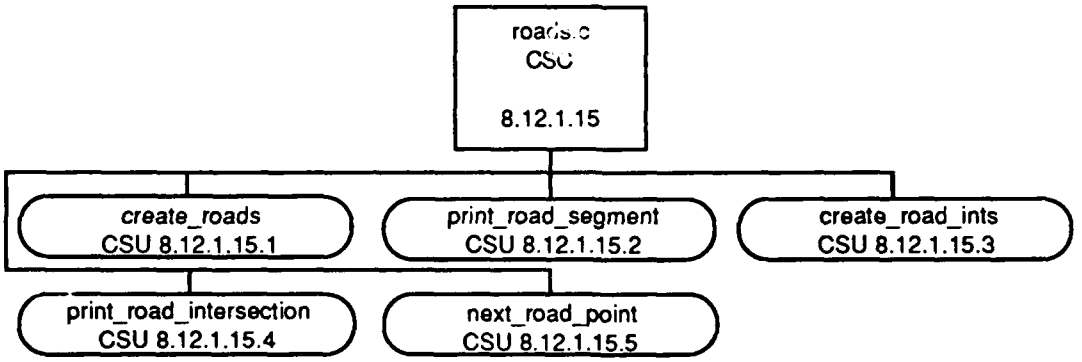


Figure 2.12-16: roads.c CSC Structure

2.12.1.15.1 create_roads CSU

This CSU reads road segment structures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Errors		
Error Name	**Reason for Error**	
File Error : ... : Open to read	Unable to open file.	
Error: Can't allocate ... road segments	Unable to allocate memory for road segments.	
Error: Can't allocate ... points	Unable to allocate memory for road points.	
Error: Reading road point list	The correct number of road points was not read.	
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
allocate_points	Sec. 2.12.1.1.5	
print_road_segment	Sec. 2.12.1.15.2	

Table 2.12-110: create_roads CSU [8.12.1.15.1]

2.12.1.15.2 print_road_segment CSU

This CSU displays road segments.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
segment_index	int	Standard

Table 2.12-111: print_road_segment CSU [8.12.1.15.2]

2.12.1.15.3 create_road_ints CSU

This CSU reads road intersection structures.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Errors		
Error Name	**Reason for Error**	
File Error : ... : Open to read	Unable to open file.	
Error: Can't allocate ... road intersections	Unable to allocate memory for road intersections.	
Error: Can't allocate ... pairs	Unable to allocate memory for number pairs.	
Error: Reading road intersection pair list	The correct number of road intersection pairs was not read.	

Calls	
Function	**Where Described**
heap_allocate	Sec. 2.14.2.1.1
print_road_intersection	Sec. 2.12.1.15.4

Table 2.12-112: create_road_ints CSU [8.12.1.15.3]

2.12.1.15.4 print_road_intersection CSU

This CSU displays road intersections.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
int_index	int	Standard

Table 2.12-113: print_road_intersection CSU [8.12.1.15.4]

2.12.1.15.5 next_road_point CSU

This CSU contains two constant definitions.

Constant	Value
LEEWAY	10
MIN_ANGLE /* 15 degrees */	0.25

Table 2.12-114: next_road_point Constant Definitions

This CSU calculates the next road point given a point on a road and a direction vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x	INT32	Sec. 2.12.1.18
y	INT32	Sec. 2.12.1.18
vector	pointer to VECTOR_2D	Sec. 2.12.1.18
first_id	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to ROAD_INFO	No features found.
&real_road_ret	pointer to ROAD_INFO	Next road point.
Calls		
Function	**Where Described**	
find_a_quad	Sec. 2.12.1.5.1	
VEC_SUB_2D	See Appendix A	
interior_angle_between_vectors	Sec. 2.14.3.5.8	

Table 2.12-115: next_road_point CSU [8.12.1.15.5]

2.12.1.16 trees.c CSC

/simnet/libsrc/libquad/trees.c

This CSC contains CSUs dealing with trees, canopies, and buildings.

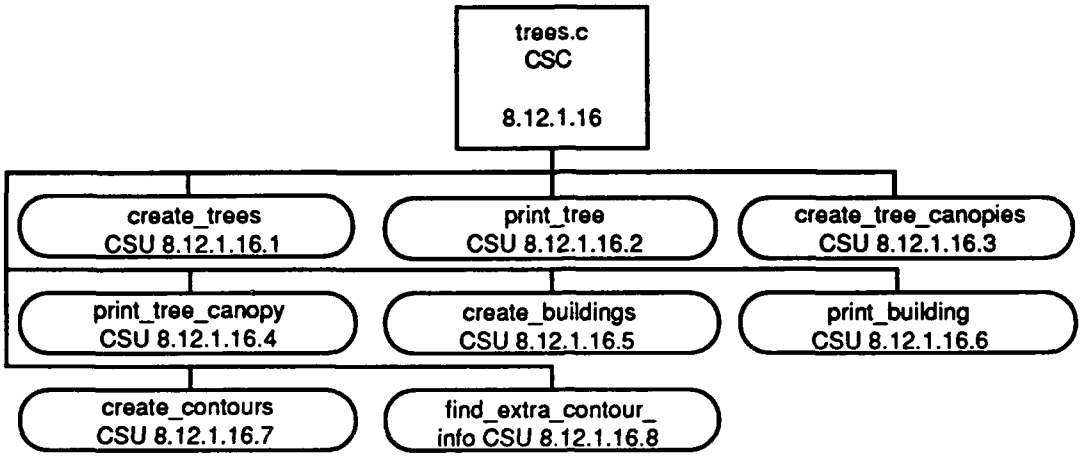


Figure 2.12-17: trees.c CSC Structure

2.12.1.16.1 create_trees CSU

This CSU reads tree structures.

Parameters		
Parameter	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Errors		
Error Name	**Reason for Error**	
File Error : ... : Open to read	Unable to open file.	
Error: Can't allocate ... trees	Unable to allocate memory for trees.	
Error: Can't allocate ... points	Unable to allocate memory for tree points.	
Error: Reading tree point list	The correct number of tree points was not read.	
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
allocate_points	Sec. 2.12.1.1.5	
print_tree	Sec. 2.12.1.16.2	

Table 2.12-116: create_trees CSU [8.12.1.16.1]

2.12.1.16.2 print_tree CSU

This CSU displays trees.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
index	int	Standard

Table 2.12-117: print_tree CSU [8.12.1.16.2]

2.12.1.16.3 create_tree_canopies CSU

This CSU reads tree canopy structures.

Parameters		
Parameter	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Errors		
Error Name	**Reason for Error**	
File Error : ... : Open to read	Unable to open file.	
Error: Can't allocate ... tree canopies	Unable to allocate memory for tree canopies.	
Error: Can't allocate ... points	Unable to allocate memory for tree canopy points.	
Error: Reading tree canopy point list	The correct number of tree canopy points was not read.	
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
print_tree_canopy	Sec. 2.12.1.16.4	

Table 2.12-118: create_tree_canopies CSU [8.12.1.16.3]

2.12.1.16.4 print_tree_canopy CSU

This CSU displays tree canopies.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
canopy_index	int	Standard

Table 2.12-119: print_tree_canopy CSU [8.12.1.16.4]

2.12.1.16.5 create_buildings CSU

This CSU reads building structures.

Parameters		
Parameter	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Errors		
Error Name	**Reason for Error**	
File Error : ... : Open to read	Unable to open file.	
Error: Can't allocate ... buildings	Unable to allocate memory for buildings.	
Error: Can't allocate ... points	Unable to allocate memory for building points.	
Error: Reading building point list	The correct number of building points was not read	
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
allocate_points	Sec. 2.12.1.1.5	
print_building	Sec. 2.12.1.16.6	

Table 2.12-120: create_buildings CSU [8.12.1.16.5]

2.12.1.16.6 print_building CSU

This CSU displays buildings.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
index	int	Standard

Table 2.12-121: print_building CSU [8.12.1.16.6]

2.12.1.16.7 create_contours CSU

This CSU reads contour line structures.

Parameters		
Parameter	**Type**	**Where Typedef Declared**
directory	pointer to char	Standard
Errors		
Error Name	**Reason for Error**	
File Error : ... : Open to read	Unable to open file.	
Error: Can't allocate ... contours	Unable to allocate memory for contours.	
Error: Can't allocate ... points	Unable to allocate memory for contour points.	
Error: Reading contour point list	The correct number of contour points was not read.	

Calls	
Function	**Where Described**
heap_allocate	Sec. 2.14.2.1.1
allocate_points	Sec. 2.12.1.1.5
find_extra_contour_info	Sec. 2.12.1.16.8
print_tree_canopy	Sec. 2.12.1.16.4

Table 2.12-122: create_contours CSU [8.12.1.16.7]

2.12.1.16.8 find_extra_contour_info CSU

This CSU calculates the boundary box around a contour line.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
contour	pointer to CANOPY	Sec. 2.12.1.18

Table 2.12-123: find_extra_contour_info CSU [8.12.1.16.8]

2.12.1.17 building_check.c CSC

/simnet/libsrc/libquad/building_check.c

This CSC checks the relationships between building locations and view.

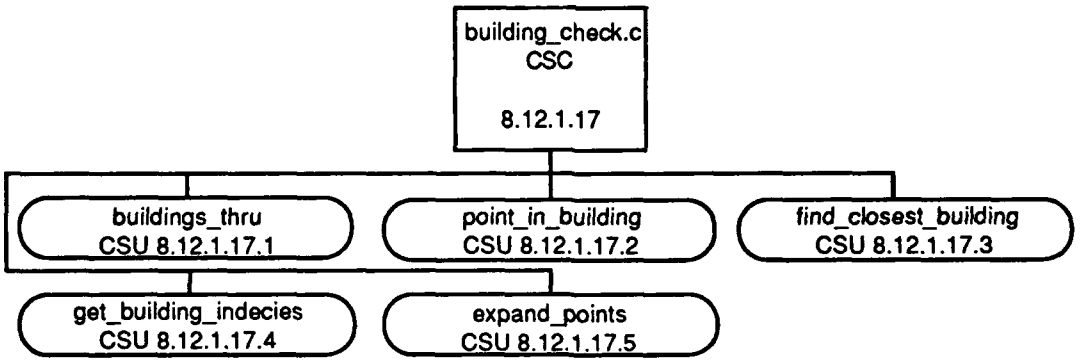


Figure 2.12-18: building_check.c CSC Structure

2.12.1.17.1 buildings_thru CSU

This CSU determines whether a line segment intersects a building structure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
offset	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to BOUNDING_BOX	No intersection.
bbox	pointer to BOUNDING_BOX	Intersected building.
Calls		
Function	**Where Described**	
get_building_indecies	Sec. 2.12.1.17.4	
minimum_clip	Sec. 2.12.1.13.1	
push	Sec. 2.12.1.7.3	
free_list	Sec. 2.12.1.7.4	
find_closest_building	Sec. 2.12.1.17.3	

Table 2.12-124: buildings_thru CSU [8.12.1.17.1]

2.12.1.17.2 point_in_building CSU

This CSU determines whether a point is within a building structure.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x	INT32	Sec. 2.12.1.18
y	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Point is in building.
FALSE	int	Point is not in building.
Calls		
Function	**Where Described**	
get_building_indecies	Sec. 2.12.1.17.4	
minimum_clip	Sec. 2.12.1.13.1	
free_list	Sec. 2.12.1.7.4	

Table 2.12-125: point_in_building CSU [8.12.1.17.2]

2.12.1.17.3 find_closest_building CSU

This CSU finds the closest building to a point.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x	INT32	Sec. 2.12.1.18
y	INT32	Sec. 2.12.1.18
building_list	pointer to LIST	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
NULL_PTR	pointer to BUILDING	No buildings in list.
building_list->car	pointer to BUILDING	Only one building in list.
building1	pointer to BUILDING	Closest building found.
Calls		
Function	**Where Described**	
distance	Sec. 2.12.1.11.1	

Table 2.12-126: find_closest_building CSU [8.12.1.17.3]

2.12.1.17.4 get_building_indecies CSU

This CSU returns all of the building indices within an area.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	INT32	Sec. 2.12.1.18
y1	INT32	Sec. 2.12.1.18
x2	INT32	Sec. 2.12.1.18
y2	INT32	Sec. 2.12.1.18
ReturnValues		
Return Value	**Type**	**Meaning**
building_index_list	pointer to LIST	List of building indices.
Calls		
Function	**Where Described**	
get_quad_nodes	Sec. 2.12.1.5.2	
MIN	See Appendix A	
MAX	See Appendix A	
push	Sec. 2.12.1.7.3	
free_list	Sec. 2.12.1.7.4	
remove_duplicates	Sec. 2.12.1.7.5	

Table 2.12-127: get_building_indecies CSU [8.12.1.17.4]

2.12.1.17.5 expand_points CSU

This CSU expands the size of a building by the value of *offset*.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
points	pointer to INT32	Sec. 2.12.1.18
mid_x	double	Standard
mid_y	double	Standard
offset	double	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
new_points	pointer to INT32	Expanded building.
Calls		
Function	**Where Described**	
VEC_SUB_2D	See Appendix A	
vec_normalize_2d	Sec. 2.12.1.9.2	
VEC_SCALE_2D	See Appendix A	
VEC_ADD_2D	See Appendix A	
heap_allocate	Sec. 2.14.2.1.1	

Table 2.12-128: expand_points CSU [8.12.1.17.5]

2.12.1.18 libquad.h CSU

/simnet/libsrc/libquad/libquad.h

This header CSC contains defines and structures relative to this library. The constant defines are in the next table.

Constant	Value
POINTS_IN_TDB	1000000
HALF_PI	1.5707963267948966
MAX_DISTANCE	100000
NULL_PTR	0
NULL_VAL	-1
BRIDGE_CROSSING	0
INTERSECTION_CROSSING	1
FORDABLE_CROSSING	2
RIVER_TYPE	0
LAKE_TYPE	1
FORDABLE_TYPE	2
CROSS_COUNTRY_POINT	0
BRIDGE_POINT	1
CLOCKWISE	0
ANTICLOCKWISE	1
FROM_GOAL	0
TO_GOAL	1
START	0
END	1
AS_IS	0
ROTATED	1

Table 2.12-129: libquad.h Constant Definitions

Macros are expanded in Appendix A. The following three typedefs deal with integer sizes.

Data Type	Assigned Type
long	INT32
short	INT16
char	INT8

Table 2.12-130: libquad.h Integer Type Definitions

The typedef structs are in the following tables.

The following typedef struct is tagged list.

Item	Type	Where Type Defined
car	int	Standard
cdr	pointer to struct list	This typedef struct

Table 2.12-131: LIST Structure Definition

Following the LIST typedef struct are the external declares LIST *intersections_searched and LIST *quad_index_list.

The quadtree typedef structure below is tagged quadtree.

Item	Type	Where Type Defined
size	INT32	This file
resolution	INT32	This file
x	INT32	This file
y	INT32	This file
num_modes	int	Standard
name	pointer to char	Standard
version	int	Standard
max_x	INT32	This file
max_y	INT32	This file

Table 2.12-132: QUADTREE Structure Definition

Following the QUADTREE typedef struct is the declare, QUADTREE quadtree.

The quadtree nodes structure is tagged quadnode.

Item	Type	Where Type Defined
feature_index	int	Standard
child_index	int	Standard

Table 2.12-133: QUAD_NODE Structure Definition

Following the QUAD_NODE typedef struct is the declare, QUAD_NODE *quad_nodes.

The features structure is tagged features.

Item	Type	Where Type Defined
water_indecies	pointer to int	Standard
water_int_indecies	pointer to int	Standard
road_indecies	pointer to int	Standard
road_int_indecies	pointer to int	Standard
bridge_indecies	pointer to int	Standard
lake_indecies	pointer to int	Standard
canopy_indecies	pointer to int	Standard
tree_indecies	pointer to int	Standard
building_indecies	pointer to int	Standard
contour_indecies	pointer to int	Standard

Table 2.12-134: QUAD_FEATURES Structure Definition

Following the QUAD_FEATURES typedef struct is the declare, QUAD_FEATURES *features_array.

The river segments structure is tagged river_segment.

Item	Type	Where Type Defined
num_pts	INT16	This file
points	pointer to INT32	This file
width	INT16	This file
distance	INT16	This file
int1	INT16	This file
int2	INT16	This file
x_min	INT32	This file
y_min	INT32	This file
x_max	INT32	This file
y_max	INT32	This file
fordable	INT8	This file

Table 2.12-135: RIVER_SEG Structure Definition

The river segments structure is tagged river_segments.

Item	Type	Where Type Defined
num_segs	INT32	This file
segments	pointer to RIVER_SEG	The previous typedef struct

Table 2.12-136: RIVER_SEGS Structure Definition

Following the RIVER_SEGS typedef struct is the declare, RIVER_SEGS river.

The river intersection structure is tagged river_int.

Item	Type	Where Type Defined
num_pairs	INT16	This file
pairs	pointer to INT16	This file
x	INT32	This file
y	INT32	This file
bridge	INT16	This file

Table 2.12-137: RIVER_INT Structure Definition

The river intersections structure is tagged river_ints_struct.

Item	Type	Where Type Defined
num_ints	INT32	This file
ints	pointer to RIVER_INT	The previous typedef struct

Table 2.12-138: RIVER_INTS Structure Definition

Following the RIVER_INTS typedef struct is the declare, RIVER_INTS river_ints.

The road segment structure is tagged road_segment.

Item	Type	Where Type Defined
num_pts	INT16	This file
points	pointer to INT32	This file
width	INT16	This file
distance	INT16	This file
int1	INT16	This file
int2	INT16	This file
x_min	INT32	This file
y_min	INT32	This file
x_max	INT32	This file
y_max	INT32	This file
bridge	INT8	This file

Table 2.12-139: ROAD_SEG Structure Definition

The road segments structure is tagged road_segments.

Item	Type	Where Type Defined
num_segs	INT32	This file
segments	pointer to ROAD_SEG	The previous typedef struct

Table 2.12-140: ROAD_SEGS Structure Definition

Following the ROAD_SEGS typedef struct is the declare, ROAD_SEGS roads.

The road intersection structure is tagged road_int.

Item	Type	Where Type Defined
num_pairs	INT16	This file
pairs	pointer to INT16	This file
x	INT32	This file
y	INT32	This file
bridge	INT16	This file

Table 2.12-141: ROAD_INT Structure Definition

The road intersections structure is tagged road_ints_struct.

Item	Type	Where Type Defined
num_ints	INT32	This file
ints	pointer to ROAD_INT	The previous typedef struct

Table 2.12-142: ROAD_INTS Structure Definition

Following the ROAD_INTS typedef struct is the declare, ROAD_INTS road_ints.

The bridge structure is tagged bridge_struct.

Item	Type	Where Type Defined
num_pts	INT16	This file
points	pointer to INT16	This file
x	INT32	This file
y	INT32	This file
width	INT16	This file

Table 2.12-143: BRIDGE Structure Definition

The bridges structure is tagged bridges_struct.

Item	Type	Where Type Defined
num_bridges	INT32	This file
bridge_ptr	pointer to BRIDGE	The previous typedef struct

Table 2.12-144: BRIDGES Structure Definition

Following the BRIDGES typedef struct is the declare, BRIDGES bridges.

The lake structure is tagged lake_struct.

Item	Type	Where Type Defined
num_pts	INT16	This file
points	pointer to INT32	This file
x_min	INT32	This file
y_min	INT32	This file
x_max	INT32	This file
y_max	INT32	This file

Table 2.12-145: LAKE Structure Definition

The lakes structure is tagged lakes_struct.

Item	Type	Where Type Defined
num_lakes	INT32	This file
lake_ptr	pointer to LAKE	The previous typedef struct

Table 2.12-146: LAKES Structure Definition

Following the LAKES typedef struct is the declare, LAKES lakes.

The tree canopy structure is tagged canopy_struct.

Item	Type	Where Type Defined
num_pts	INT16	This file
points	pointer to INT32	This file
x_min	INT32	This file
y_min	INT32	This file
x_max	INT32	This file
y_max	INT32	This file
height	INT16	This file

Table 2.12-147: CANOPY Structure Definition

The tree canopies structure is tagged canopies_struct.

Item	Type	Where Type Defined
num_canopies	INT32	This file
canopy_ptr	pointer to CANOPY	The previous typedef struct

Table 2.12-148: CANOPIES Structure Definition

Following the CANOPIES typedef struct is the declare, CANOPIES canopies.

The tree and treeline structure is tagged tree_struct.

Item	Type	Where Type Defined
num_pts	INT16	This file
points	pointer to INT32	This file
height	INT16	This file

Table 2.12-149: TREE Structure Definition

The trees and treeline structure is tagged trees_struct.

Item	Type	Where Type Defined
num_trees	INT32	This file
tree_ptr	pointer to TREE	The previous typedef struct

Table 2.12-150: TREES Structure Definition

Following the TREES typedef struct is the declare, TREES trees.

The building structure is tagged building_struct.

Item	Type	Where Type Defined
num_pts	INT16	This file
points	pointer to INT32	This file
x_min	INT32	This file
y_min	INT32	This file
x_max	INT32	This file
y_max	INT32	This file
height	INT16	This file

Table 2.12-151: BUILDING Structure Definition

The buildings structure is tagged buildings_struct.

Item	Type	Where Type Defined
num_buildings	INT32	This file
building_ptr	pointer to BUILDING	The previous typedef struct

Table 2.12-152: BUILDINGS Structure Definition

Following the BUILDINGS typedef struct is the declare, BUILDINGS buildings.

The following typedef struct is tagged crossing.

Item	Type	Where Type Defined
type	int	Standard
segment_list	pointer to LIST	This file
union{		
bridge	pointer to BRIDGE	This file
intersection	pointer to RIVER_INT	This file
segment	pointer to RIVER_SEG	This file
	} crossing	

Table 2.12-153: RIVER_CROSSING Structure Definition

The following typedef struct is tagged vector.

Item	Type	Where Type Defined
x	double	Standard
y	double	Standard

Table 2.12-154: VECTOR_2D Structure Definition

The following typedef struct is tagged water_int.

Item	Type	Where Type Defined
type	int	Standard
index	int	Standard
intersection	pointer to VECTOR_2D	The previous typedef struct
seg_point_1	pointer to VECTOR_2D	The previous typedef struct
seg_point_2	pointer to VECTOR_2D	The previous typedef struct

Table 2.12-155: WATER_INT Structure Definition

The following typedef struct is tagged route_pt.

Item	Type	Where Type Defined
x	INT32	This file
y	INT32	This file
type	int	Standard

Table 2.12-156: ROUTE_PT Structure Definition

The following typedef struct is tagged bounding_box.

Item	Type	Where Type Defined
x_min	INT32	This file
y_min	INT32	This file
x_max	INT32	This file
y_max	INT32	This file

Table 2.12-157: BOUNDING_BOX Structure Definition

The following typedef struct is tagged road_info.

Item	Type	Where Type Defined
index	int	Standard
x	INT32	This file
y	INT32	This file
bridge	INT8	This file

Table 2.12-158: ROAD_INFO Structure Definition

2.13 GLOBAL CSC

The Global CSC defines the global variables, constants, macros, and CSUs which are used throughout the SAF Simhost CSCI.

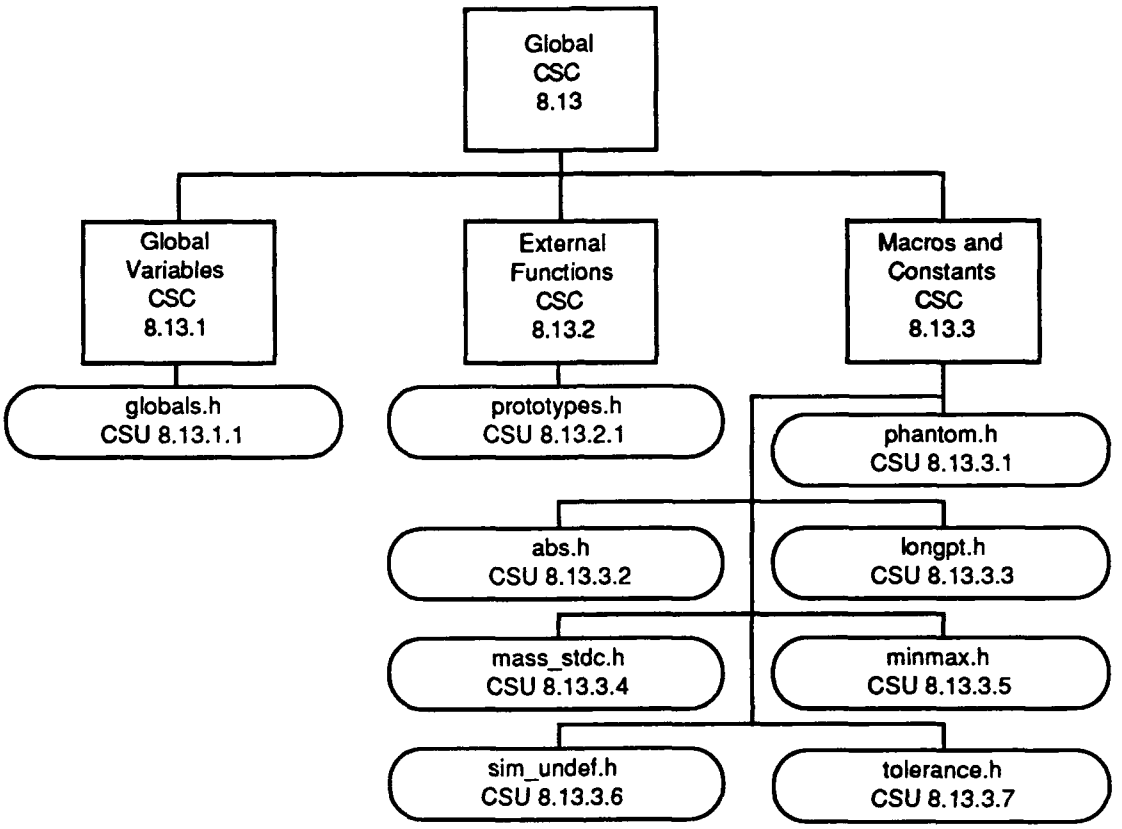


Figure 2.13-1: Global CSC Structure

2.13.1 Global Variables CSC

The Global Variables CSC [8.13.1] contains one file, globals.h.

2.13.1.1 globals.h CSU

/simnet/src/host/globals.h

This file contains the external declarations for the globally accessed variables of the phantom. State variables and variables which hold the tables of information loaded in from the .lisp files are declared.

2.13.2 External Functions CSC

The External Functions CSC [8.13.2] contains one file, prototype.h

2.13.2.1 prototypes.h CSU

/simnet/src/host/prototypes.h

This file contains external declarations for a number of functions which are used in the phantom.

2.13.3 Macros and Constants CSC

The Macros and Contants CSC [8.13.3] contains much of the globally used macros and constants in eight header files.

2.13.3.1 phantom.h CSU

/simnet/src/host/phantom.h

This CSU acts as a central repository for all of the include files which will be needed during compilation. All of the .c files need only to include this file, and all of the phantom .h's and library and standard includes will be used. The file also contains macro definitions for speed conversions and vehicle status querys, which are shown in Appendix A, and constants contained in the following table.

Constant	Value
MAX_VEH	2048
SOIL_NA	0 /* Unknown type of soil */
SOIL_ROAD	1 /* Asphalt or other hard surface */
SOIL_RC1250	2 /* Packed soil */
SOIL_RC1050	3 /* Sandy soil */
SOIL_WATER	5 /* River that is passable */
SOIL_MUCK	4 /* River that is impassable */
MAX_VIEWS	3
SMALL_VELOCITY_SQUARED	25
TERRAIN_GRID_SIZE	100
GRAVITY	9.8
COMPOSITE_MONITOR_TIME	120000
MAX_VISION_RANGE	3500.0 /* meters (same as sims) */
ADA_MAX_VISION_RANGE	7000.0
SECONDS2GALLONS	0.0125 /* Fuel consumption constant */
MAX_TDB_EXTENSION	5000.0
VEHICLE_STATUS_IMMOBILE	0x01 /* Vehicle status bits */
VEHICLE_STATUS_CANTFIRE	0x02
VEHICLE_STATUS_DESTROYED	0x03
VEHICLE_STATUS_OUTOFGAS	0x08
VEHICLE_STATUS_OUT)FAMMO	0x10
VEHICLE_STATUS_STUCK	0x80
VEHICLE_STATUS_CANTMOVE	0x8D /*immobile |destroyed | outofgas | stuck*/
HULL_STATE_NORMAL	0 /* Ground vehicle states */
HULL_STATE_TRASHING	1
HULL_STATE_TRASHED	2

Table 2.13-1: phantom.h Constant Definitions

2.13.3.2 abs.h CSU

/simnet/include/saf/src/abs.h

This CSU contains the macro definition for abs(x) which is described in Appendix A.

2.13.3.3 longpt.h CSU

/simnet/include/saf/src/longpt.h

This CSU defines types used for describing points (LongPt) and rectangles (LongRect) with sufficient precision to locate to within a meter in a few hundred kilometers.

Item	Type	Where Type Defined
x, y	long	Standard

Table 2.13-2: LongPt Structure

Item	Type	Where Type Defined
bottom, left, top, right	long	Standard

Table 2.13-3: LongRect Structure

2.13.3.4 mass_stdc.h CSU

/simnet/include/saf/src/mass_stdc.h

This CSU contains the definitions for the basic data types caused by differences in C compliers.

Storage Allocated	Masscomp C Language	Independent Data Type
16 bit signed integer	short	INT_2
32 bit signed integer	int	INT_4
8 bit unsigned integer	unsigned char	BYTE
8 bit unsigned integer	unsigned char	BOOLEAN
16 bit unsigned integer	unsigned short	HWORD
32 bit unsigned integer	unsigned int	WORD
real	float	REAL_4
real	double	REAL_8
8 bit char	char	*STRING

Table 2.13-4: Data type definitions in mass_stdc.h

2.13.3.5 minmax.h CSU

/simnet/include/saf/src/minmax.h

This CSU contains the macro definitions min(a, b) and max(a, b) which are described in Appendix A.

2.13.3.6 sim_undef.h CSU

/simnet/include/saf/src/sim_undef.h

This CSU undefines min, max, and eq. The first two are defined in minmax.h (Sec. 2.13.3.5) and eq is contained in tolerance.h (Sec. 2.13.3.7).

2.13.3.7 tolerance.h CSU

/simnet/include/saf/src/tolerance.h

This CSU contains a selection of constants, listed below, used as tolerance and threshold levels.

Constant	Value
E_PICO	1.0E-12
E_NANO	1.0E-09
E_MICRO	1.0E-06
E_MILLI	1.0E-03
E_CENTI	1.0E-02
E_DECI	1.0E-01
E_HALF	5.0E-01
E_NINE_TENTHS	9.0E-01
E_ONE	1.0
E_DECA	1.0E01
E_HECTO	1.0E02
E_KILO	1.0E03
E_MEGA	1.0E06
E_GIGA	1.0E09

Table 2.13-5: tolerance.h Constants

The tolerance.h CSU also contains two macro definitions eq(x, y, e) and step(x, z, e) which are described in Appendix A.

2.14 UTILITIES CSC

This CSC contains CSUs which are used by the remainder of the simhost code. This includes mathematical functions, memory allocation and deallocation utilities.

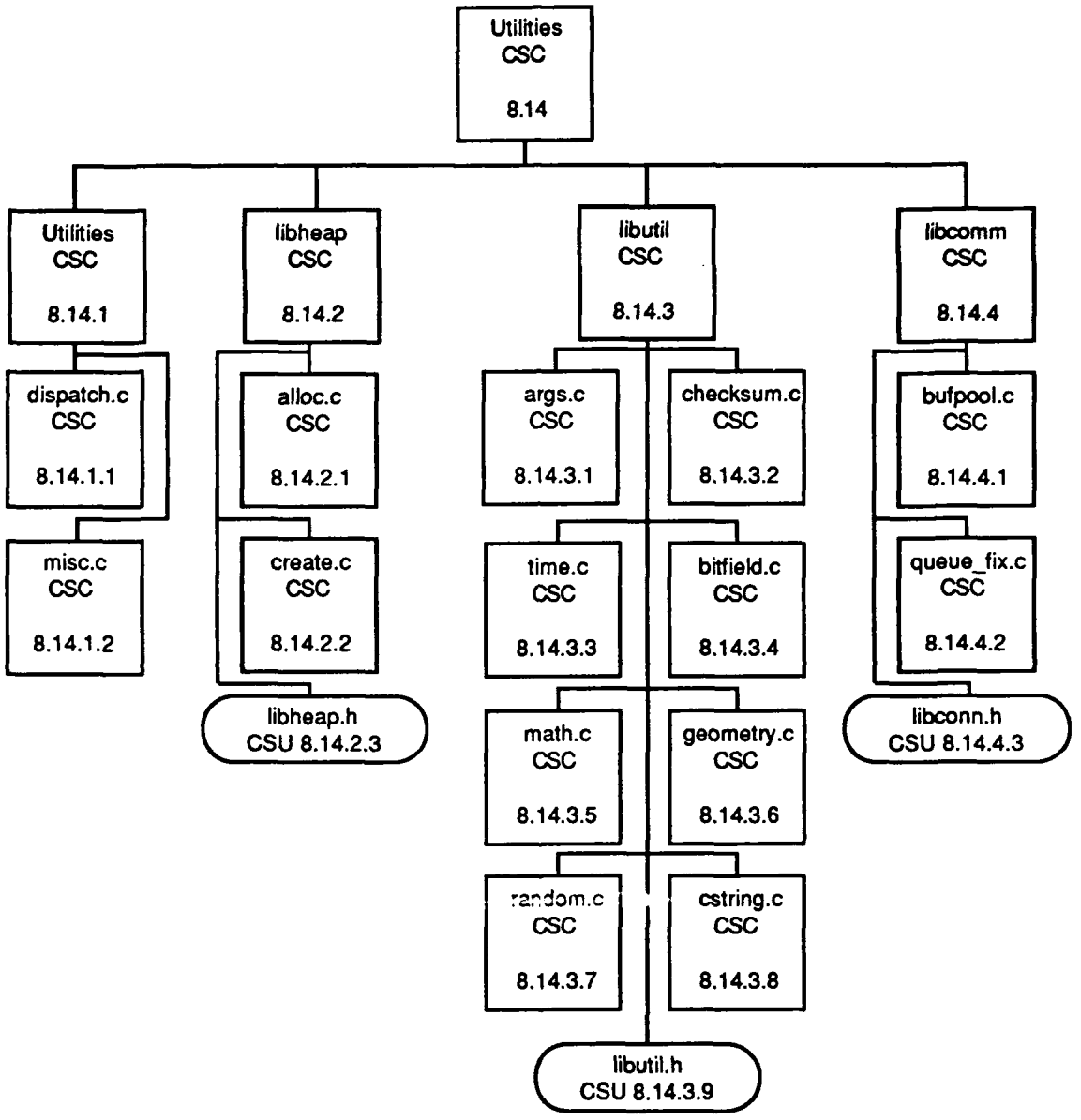


Figure 2.14-1: Utilities CSC Structure

2.14.1 Utilities CSC

The Utilities CSC [8.14.1] consists of two files dispatch.c [8.14.1.1], and misc.c [8.14.1.2], .

2.14.1.1 dispatch.c CSC

/simnet/src/host/dispatch.c

This CSC contains various CSUs dealing with SAF objects.

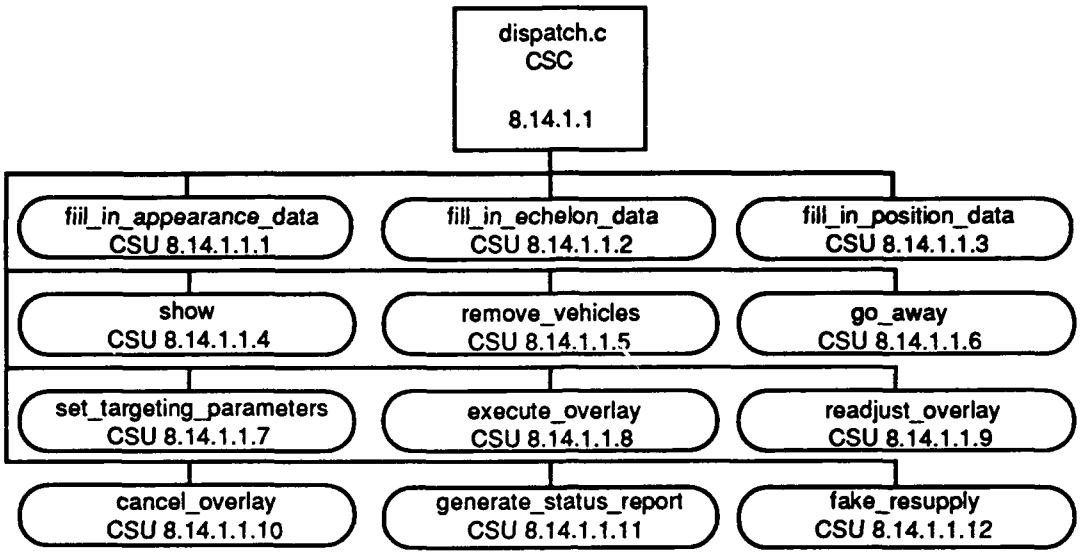


Figure 2.14-2: dispatch.c CSC Structure

2.14.1.1.1 fill_in_appearance_data CSU

This CSU fills in a SAF object's appearance data based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
*desc_ptr	pointer to VEHICLE_APPEARANCE_ DESCRIPTOR	Sec. 2.4.1.1

Errors	
Error Name	**Reason for Error**
ERROR_OUT	No fill in appearance data function for SAF type

Calls	
Function	**Where Described**
remote_fill_in_appearance_ data	Sec. 2.7.1.6
saf_vehicle_fill_in_ appearance_data	Sec. 2.6.1.1.13
composite_fill_in_ appearance_data	Sec. 2.8.1.3.14
ERROR_OUT	Sec. 2.5.2.2

Table 2.14-1: fill_in_appearance_data CSU [8.14.1.1.1]

2.14.1.1.2 fill_in_echelon_data CSU

This CSU fills in a SAF object's echelon data based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
*desc_ptr	pointer to VEHICLE_ECHELON_ DESCRIPTOR	Sec. 2.4.1.1
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	No fill in echelon data function for SAF type	
Calls		
Function	**Where Described**	
remote_fill_in_echelon_data	Sec. 2.7	
saf_vehicle_fill_in_echelon_ data	Sec. 2.6.1.1.14	
composite_fill_in_echelon_ data	Sec. 2.8.1.3.12	
ERROR_OUT	Sec. 2.5.2.2	

Table 2.14-2: fill_in_echelon_data CSU [8.14.1.1.2]

2.14.1.1.3 fill_in_position_data CSU

This CSU fills in a SAF object's position data based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
*desc_ptr	pointer to VEHICLE_POSITION_ DESCRIPTOR	Sec. 2.4.1.1
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	No fill in position data function for SAF type	
Calls		
Function	**Where Described**	
remote_fill_in_position_data	Sec. 2.7.1.8	
saf_vehicle_fill_in_position_ data	Sec. 2.6.1.1.15	
composite_fill_in_position_ data	Sec. 2.8.1.3.13	
ERROR_OUT	Sec. 2.5.2.2	

Table 2.14-3: fill_in_position_data CSU [8.14.1.1.3]

2.14.1.1.4 show CSU

This CSU prints information about a SAF object based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
flags	int	Standard
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	No show function for SAF type	
Calls		
Function	**Where Described**	
remote_show	Sec. 2.7.1.4	
saf_vehicle_show	Sec. 2.6.1.1.11	
composite_show	Sec. 2.8.1.3.3	
ERROR_OUT	Sec. 2.5.2.2	

Table 2.14-4: show CSU [8.14.1.1.4]

2.14.1.1.5 remove_vehicles CSU

This CSU removes a SAF object based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
num	int	Standard
v_list[ ]	unsigned int	Standard
Calls		
Function	**Where Described**	
saf_vehicle_remove_vehicles	Sec. 2.6.1.1.35	
composite_remove_vehicles	Sec. 2.8.1.3.34	

Table 2.14-5: remove_vehicles CSU [8.14.1.1.5]

2.14.1.1.6 go_away CSU

This CSU causes a SAF object to go away based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	No go away function for remote type	

Calls	
Function	**Where Described**
saf_vehicle_go_away	Sec. 2.6.1.1.2
remote_go_away	Sec. 2.7.1.3
composite_go_away	Sec. 2.8.1.3.2
ERROR_OUT	Sec. 2.5.2.2

Table 2.14-6: go_away CSU [8.14.1.1.6]

2.14.1.1.7 set_targeting_parameters CSU

This CSU sets targeting parameters based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
firestatus	int	Standard
max_engagement_range	REAL	sim_types.h
marksmanship	REAL	sim_types.h
position_x	REAL	sim_types.h
position_y	REAL	sim_types.h
radius	REAL	sim_types.h
targets[ ]	unsigned short	Standard
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	Cannot set targeting parameters for remote vehicle	
ERROR_OUT	No set targeting parameters for SAF type	
Calls		
Function	**Where Described**	
saf_vehicle_set_targeting_parameters	Sec. 2.6.1.1.12	
ERROR_OUT	Sec. 2.5.2.2	
OBJ_VEHICLEID	Sec. 2.9.1.1	
composite_set_targeting_parameters	Sec. 2.8.1.3.31	

Table 2.14-7: set_targeting_parameters CSU [8.14.1.1.7]

2.14.1.1.8 execute_overlay CSU

This CSU executes an overlay for a local vehicle or composite based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
*overlay	pointer to OVERLAY	
*cm	pointer to CONTROL_MEASURE	

Errors	
Error Name	**Reason for Error**
ERROR_OUT	No execute overlay function for SAF type
Calls	
Function	**Where Described**
saf_vehicle_execute_overlay	Sec. 2.6.1.1.46
composite_execute_overlay	Sec. 2.8.1.3.40
ERROR_OUT	Sec. 2.5.2.2

Table 2.14-8: execute_overlay CSU [8.14.1.1.8]

2.14.1.1.9 readjust_overlay CSU

This CSU readjusts an overlay for a local vehicle or composite based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
*overlay	pointer to OVERLAY	
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	No readjust overlay function for SAF type	
Calls		
Function	**Where Described**	
OBJ_VEHICLEID	Sec. 2.9.1.1	
composite_readjust_overlay	Sec. 2.8.1.3.41	
ERROR_OUT	Sec. 2.5.2.2	

Table 2.14-9: readjust_overlay CSU [8.14.1.1.9]

2.14.1.1.10 cancel_overlay CSU

This CSU cancels an overlay for a local vehicle or composite based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	No cancel overlay function for SAF type	

Calls	
Function	**Where Described**
saf_vehicle_cancel_overlay	Sec. 2.6.1.1.47
composite_cancel_overlay	Sec. 2.8.1.3.42
ERROR_OUT	Sec. 2.5.2.2

Table 2.14-10: cancel_overlay CSU [8.14.1.1.10]

2.14.1.1.11 generate_status_report CSU

This CSU generates a status report for a local vehicle or composite based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	No generate status report function for SAF type	
Calls		
Function	**Where Described**	
saf_vehicle_generate_status_report	Sec. 2.6.1.1.34	
composite_generate_status_report	Sec. 2.8.1.3.33	
ERROR_OUT	Sec. 2.5.2.2	

Table 2.14-11: generate_status_report CSU [8.14.1.1.11]

2.14.1.1.12 fake_resupply CSU

This CSU fakes the resupply of a local vehicle or composite based on SAF type.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*safobj	pointer to SAF_OBJECT	Sec. 2.9.1.1
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	No fake resupply function for SAF type	
Calls		
Function	**Where Described**	
saf_vehicle_fake_resupply	Sec. 2.6.1.1.33	
composite_fake_resupply	Sec. 2.8.1.3.32	
ERROR_OUT	Sec. 2.5.2.2	

Table 2.14-12: fake_resupply CSU [8.14.1.1.12]

2.14.1.2 misc.c CSC

/simnet/src/host/misc.c

This CSC contains various CSUs involving inquiries to the terrain database.

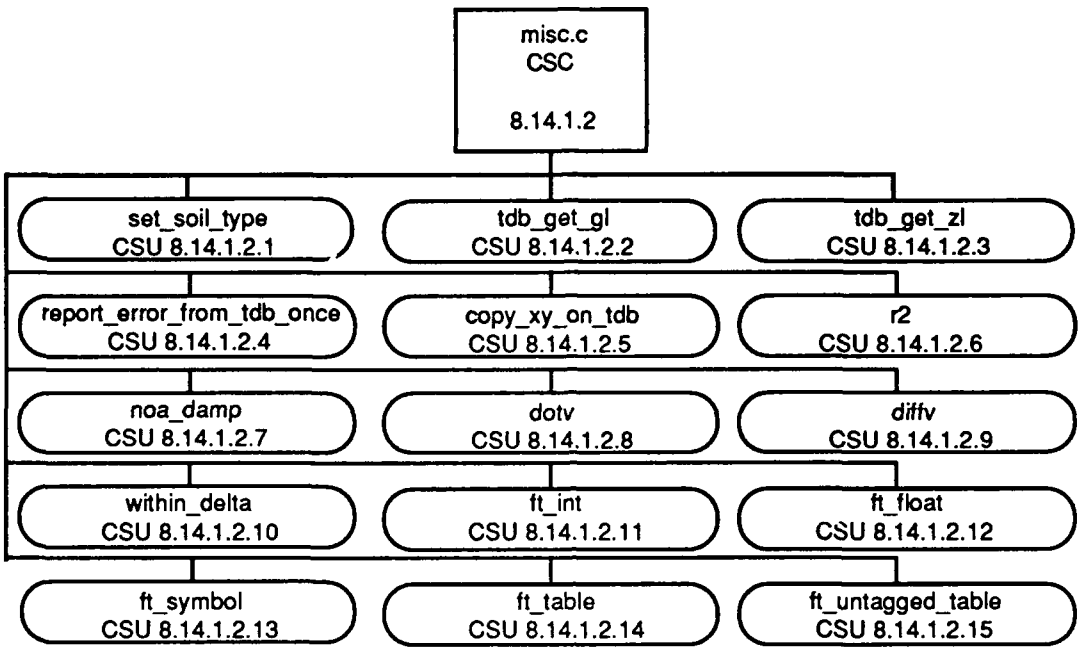


Figure 2.14-3: misc.c CSC Structure

2.14.1.2.1 get_soil_type CSU

This CSU returns the soil type for the vector passed, as long as the vector is within the defined world.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
vec	VECTOR	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
ret	int	Soil type
SOIL_NA	int	Soil not available
Errors		
Error Name	**Reason for Error**	
return(SOIL_NA)	Vector outside world	

Calls	
Function	**Where Described**
vec_copy	Sec. 2.6.2.59.1 Vehicles CSCI SDD
coords_within_database	Sec. 2.13.3.1
tdb_get_z	Sec. 21.7.16.2 MCC CSCI SDD
report_error_from_tdb_once	Sec. 2.14.1.2.4

Table 2.14-13: get_soil_type CSU [8.14.1.2.1]

2.14.1.2.2 tdb_get_gl CSU

This routine finds the ground level.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*vec	pointer to REAL	sim_types.h
Calls		
Function	**Where Described**	
coords_within_database	Sec. 2.13.3.1	
tdb_get_z	Sec. 2.21.7.16.2 MCC CSCI SDD	
report_error_from_tdb_once	Sec. 2.14.1.2.4	

Table 2.14-14: tdb_get_gl CSU [8.14.1.2.2]

2.14.1.2.3 tdb_get_zl CSU

This routine finds the height above ground.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*vec	pointer to REAL	sim_types.h
Calls		
Function	**Where Described**	
coords_within_database	Sec. 2.13.3.1	
tdb_get_z	Sec. 2.21.7.16.2 MCC CSCI SDD	
report_error_from_tdb_once	Sec. 2.14.1.2.4	

Table 2.14-15: tdb_get_zl CSU [8.14.1.2.3]

2.14.1.2.4 report_error_from_tdb_once CSU

This routine generates an error report for the terrain database functions.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*vec	pointer to REAL	sim_types.h
Errors		
Error Name	**Reason for Error**	
ERROR_OUT	Error from libtdb	
ERROR_OUT	Error from tdb	
Calls		
Function	**Where Described**	
ERROR_OUT	Sec. 2.5.2.2	
tdb_error	2.21.17.7.1 MCC CSCI SDD	

Table 2.14-16: report_error_from_tdb_once CSU [8.14.1.2.4]

2.14.1.2.5 copy_xy_on_tdb CSU

This routine copies a coordinate into the terrain database.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
position	VECTOR	sim_types.h
replace	VECTOR	sim_types.h

Table 2.14-17: copy_xy_on_tdb CSU [8.14.1.2.5]

2.14.1.2.6 r2 CSU

This CSU returns the sum of the square of the differences of the two "x" and "y" values passed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x1	REAL	sim_types.h
y1	REAL	sim_types.h
x2	REAL	sim_types.h
y2	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
(dx*dx+dy*dy)	REAL	Sum of squared differences

Table 2.14-18: r2 CSU [8.14.1.2.6]

2.14.1.2.7 noa_damp CSU

This CSU returns a dampened value for delta_z, if delta_z is greater than or equals -6.0.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
delta_z	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
(-sqrt(-6.0*delta_z)	int	Dampened value
delta_z	int	delta_z greater than or equal to -6.0

Table 2.14-19: noa_damp CSU [8.14.1.2.7]

2.14.1.2.8 dotv CSU

This CSU computes the dot product ("result" sum) of the "x" and "y" values of two "n" element vectors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x[ ]	REAL	sim_types.h
ry]	REAL	sim_types.h
n	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
result	REAL	Dot product

Table 2.14-20: dotv CSU [8.14.1.2.8]

2.14.1.2.9 diffv CSU

This CSU computes the "r" differences of "x" and "y" values of two "n" element vectors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
x[ ]	REAL	sim_types.h
y[ ]	REAL	sim_types.h
r[ ]	REAL	sim_types.h
n	int	Standard

Table 2.14-21: diffv CSU [8.14.1.2.9]

2.14.1.2.10 within_delta CSU

This CSU tests for the tolerance of a point, either two dimensional (as with a tank) or three dimensional (as with a plane), depending on the number of dimentions passed in "dims".

Parameters		
Parameters	**Type**	**Where Typedef Declared**
start[ ]	REAL	sim_types.h
finish[ ]	REAL	sim_types.h
goal[ ]	REAL	sim_types.h
delta2	REAL	sim_types.h
dims	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Point within delta
FALSE	int	Point not within delta
Calls		
Function	**Where Described**	
diffv	Sec. 2.14.1.2.9	
dotv	Sec. 2.14.1.2.8	

Table 2.14-22: within_delta CSU [8.14.1.2.10]

2.14.1.2.11 ft_int CSU

This CSU tries to find an integer tag through a call to find_tag.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*symbol	pointer to char	Standard
*table	pointer to DATA_UNION	Sec 2.1.1.5
ReturnValues		
Return Value	**Type**	**Meaning**
0	int	No tag found
result[2].integer	int	Results
Calls		
Function	**Where Described**	
find_tag	Sec. 2.1.1.4.3	

Table 2.14-23: ft_int CSU [8.14.1.2.11]

2.14.1.2.12 ft_float CSU

This CSU tries to find a float tag through a call to find_tag.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*symbol	pointer to char	Standard
*table	pointer to DATA_UNION	Sec 2.1.1.5
ReturnValues		
Return Value	**Type**	**Meaning**
0	real	No tag found
result[2].ral	real	Results
Calls		
Function	**Where Described**	
find_tag	Sec. 2.1.1.4.3	

Table 2.14-24: ft_float CSU [8.14.1.2.12]

2.14.1.2.13 ft_symbol CSU

This CSU tries to find a symbol (character) tag through a call to find_tag.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*symbol	pointer to char	Standard
*table	pointer to DATA_UNION	Sec 2.1.1.5
ReturnValues		
Return Value	**Type**	**Meaning**
0	pointer to char	No tag found
result[2].charptr	pointer to char	Results
Calls		
Function	**Where Described**	
find_tag	Sec. 2.1.1.4.3	

Table 2.14-25: ft_symbol CSU [8.14.1.2.13]

2.14.1.2.14 ft_table CSU

This CSU tries to find a tagged table tag through a call to find_tag.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*symbol	pointer to char	Standard
*table	pointer to DATA_UNION	Sec 2.1.1.5

ReturnValues		
Return Value	**Type**	**Meaning**
0	pointer to DATA_UNION	No tag
result	pointer to DATA_UNION	Pointer to tag
Calls		
Function	**Where Described**	
find_tag	Sec. 2.1.1.4.3	

Table 2.14-26: ft_table CSU [8.14.1.2.14]

2.14.1.2.15 ft_untagged_table CSU

This CSU tries to find an untagged table tag through a call to find_tag.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*symbol	pointer to char	Standard
*table	pointer to DATA_UNION	Sec 2.1.1.5
ReturnValues		
Return Value	**Type**	**Meaning**
0	pointer to DATA_UNION	No tag
result	pointer to DATA_UNION	Pointer to tag
Calls		
Function	**Where Described**	
find_tag	Sec. 2.1.1.4.3	

Table 2.14-27: ft_untagged_table CSU [8.14.1.2.15]

2.14.2 libheap CSC

/simnet/libsrc/libheap

This library speeds up the program by making storage allocation much faster. Instead of handling dynamic storage requests during the execution of the program, which is relatively slow, the program grabs a large chunk of code at start up time, and gives out pieces as they come to be needed. This code also handles the deallocation of the previously requested space, and does it's own garbage collection.

2.14.2.1 alloc.c CSC

/simnet/libsrc/libheap/alloc.c

This CSC contains CSUs which handle heap allocation.

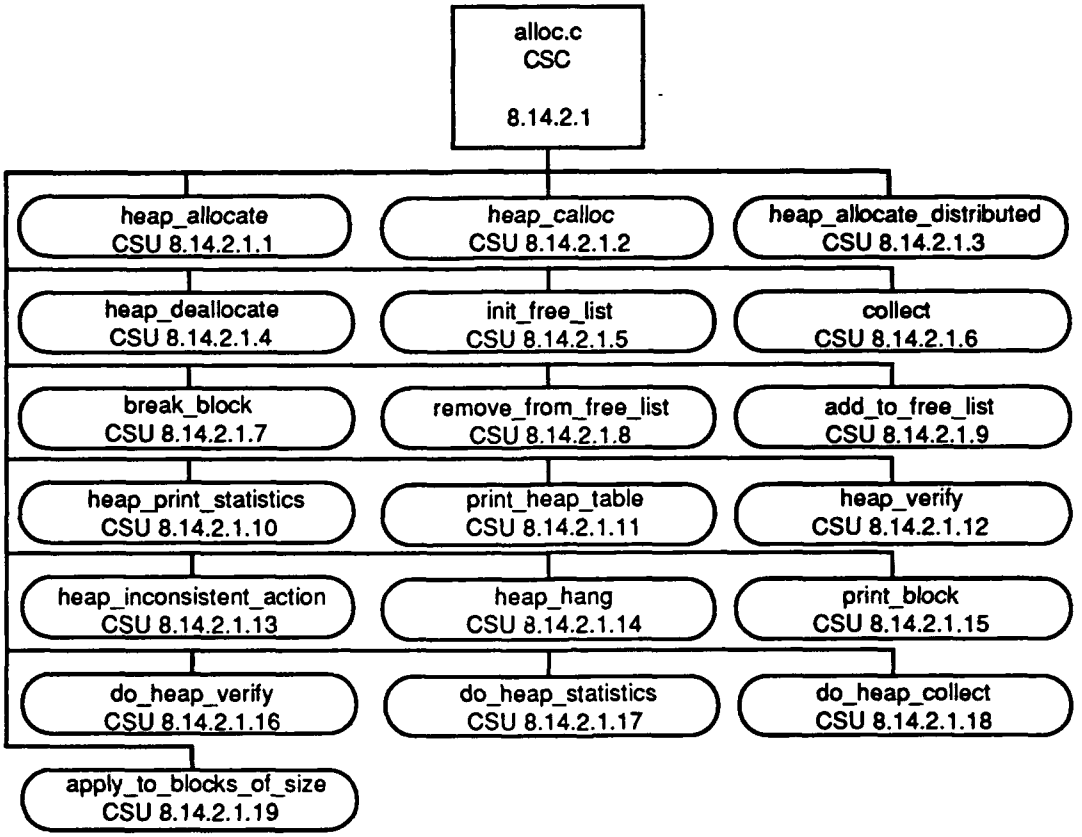


Figure 2.14-4: misc.c CSC Structure

This CSC contains the following constant definitions.

Constant	Value
FALSE	0
TRUE	1
YES_FREE	0xf0f0f0f0
NOT_FREE	0xfafafafafa
IN_FREE_LIST	0x20202020
USED_BLOCK_NEXT_FLAG	0x00000042

Table 2.14-28: alloc.c Constant Definitions

2.14.2.1.1 heap_allocate CSU

This CSU allocates a block of memory and initializes it.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
size	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
NULL	pointer to char	Ran out of heap
((char *)((int)free_block + sizeof(BLOC_HEADER	pointer to char	Pointer to block if USE_BBN_HEAP defined
((char *)mallor(size))	pointer to char	Block allocatated if USE_BBN_HEAP not defined
Calls		
Function	**Where Described**	
heap_inconsistent_action	Sec. 2.14.2.1.13	
heap_print_statistics	Sec. 2.14.2.1.10	
collect	Sec. 2.14.2.1.6	
remove_from_free_list	Sec. 2.14.2.1.8	
add_to_free_list	Sec. 2.14.2.1.9	
break_block	Sec. 2.14.2.1.7	

Table 2.14-29: heap_allocate CSU [8.14.2.1.1]

2.14.2.1.2 heap_calloc CSU

This routine returns a pointer to a block of memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
count	int	Standard
size_of_each	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
p	pointer to char	Pointer to heap if USE_BBN_HEAP defined
((char *)calloc (count,size_of_each)	pointer to char	Pointer to heap if USE_BBN_HEAP not defined

Table 2.14-30: heap_calloc CSU [8.14.2.1.2]

2.14.2.1.3 heap_allocate_distributed CSU

This CSU calls heap_allocate. It returns a pointer to the location of a block of memory that has been allocated.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
size	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
(heap_allocate(size))	pointer to char	Pointer to heap allocated
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	

Table 2.14-31: heap_allocate_distributed CSU [8.14.2.1.3]

2.14.2.1.4 heap_deallocate CSU

This CSU deallocates a block of memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*space	pointer to char	Standard
Calls		
Function	**Where Described**	
add_to_free_list	Sec. 2.14.2.1.9	
heap_inconsistent_action	Sec. 2.14.2.1.13	

Table 2.14-32: heap_deallocate CSU [8.14.2.1.4]

2.14.2.1.5 init_free_list CSU

This routine initializes the pointers to the free memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*data_space	pointe to char	Standard
size	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
free_block	pointer to BLOCK_HEADER	Point to block

Table 2.14-33: init_free_list CSU [8.14.2.1.5]

2.14.2.1.6 collect CSU

This CSU goes through the block area and collects free blocks of memory into larger blocks of memory.

Calls	
Function	**Where Described**
NEXT_BLOCK	Sec. 2.14.2.3
add_to_free_list	Sec. 2.14.2.1.9
heap_inconsistent_action	Sec. 2.14.2.1.13

Table 2.14-34: collect CSU [8.14.2.1.6]

2.14.2.1.7 break_block CSU

This CSU breaks a block into two pieces. The first piece is of size "size". It returns the second piece, or NULL if it can't break the block.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*block	pointer to BLOCK_HEADER	Sec. 2.14.2.3
size	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
NULL	pointer to BLOCK_HEADER	Can't break block
new_block	pointer to BLOCK_HEADER	Pointer to second piece

Table 2.14-35: break_block CSU [8.14.2.1.7]

2.14.2.1.8 remove_from_free_list CSU

This routijne removes a block of free memory from the list of available memory blocks.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*prev	pointer to BLOCK_HEADER	Sec. 2.14.2.3
*block	pointer to BLOCK_HEADER	Sec. 2.14.2.3

Table 2.14-36: remove_from_free_list CSU [8.14.2.1.8]

2.14.2.1.9 add_to_free_list CSU

This routijne adds a block of free memory to the list of available memory blocks.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*block	pointer to BLOCK_HEADER	Sec. 2.14.2.3

Table 2.14-37: add_to_free_list CSU [8.14.2.1.9]

The following constant was defined at this point in the file for use with the heap statistic.

Constant	Value
TABLE_SIZE	512

Table 2.14-38: TABLE_SIZE Constant Definition

2.14.2.1.10 heap_print_statistics CSU

This CSU prints the statistics concerning the heap.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
long_version	int	Standard
Calls		
Function	**Where Described**	
heap_inconsistent_action	Sec. 2.14.2.1.13	
NEXT_BLOCK	Sec. 2.14.2.3	
print_heap_table	Sec. 2.14.2.1.11	

Table 2.14-39: heap_print_statistics CSU [8.14.2.1.10]

2.14.2.1.11 print_heap_table CSU

This CSU prints the heap table's statistics.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*table	pointer to char	Standard

Table 2.14-40: print_heap_table CSU [8.14.2.1.11]

2.14.2.1.12 heap_verify CSU

This routine checks the free list and verifies all of the blocks.

Calls	
Function	**Where Described**
print_block	Sec. 2.14.2.1.15
heap_hang	Sec. 2.14.2.1.14

Table 2.14-41: heap_verify CSU [8.14.2.1.12]

2.14.2.1.13 heap_inconsistent_action CSU

This routine indicates that a block is corrupt.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*block	pointer to BLOCK_HEADER	Sec. 2.14.2.3
Calls		
Function	**Where Described**	
print_block	Sec. 2.14.2.1.15	
heap_hang	Sec. 2.14.2.1.14	

Table 2.14-42: heap_inconsistent_action CSU [8.14.2.1.13]

2.14.2.1.14 heap_hang CSU

This CSU continually prints out a heap error message, looping indefinately for debugging purposes.

Calls	
Function	**Where Described**
getpid	Sec. 2.

Table 2.14-43: heap_hang CSU [8.14.2.1.14]

2.14.2.1.15 print_block CSU

This CSU prints previous and troubled block statistics, along with a passed message.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*block	pointer to BLOCK_HEADER	Sec. 2.14.2.3
*prev	pointer to BLOCK_HEADER	Sec. 2.14.2.3
*message	pointer to char	Standard

Table 2.14-44: print_block CSU [8.14.2.1.15]

2.14.2.1.16 do_heap_verify CSU

If USE_BBN_HEAP is defined, this CSU verifies the heap through a call to heap_verify.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
n	int	Standard
Calls		
Function	**Where Described**	
heap_verify	Sec. 2.14.2.1.12	

Table 2.14-45: do_heap_verify CSU [8.14.2.1.16]

2.14.2.1.17 do_heap_statistics CSU

If USE_BBN_HEAP is defined, this CSU prints the heap statistics through a call to heap_print_statistics.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
n	int	Standard
long_version	int	Standard
Calls		
Function	**Where Described**	
heap_print_statistics	Sec. 2.14.2.1.10	

Table 2.14-46: do_heap_statistics CSU [8.14.2.1.17]

2.14.2.1.18 do_heap_collect CSU

If USE_BBN_HEAP is defined, this CSU does a heap collect through a call to collect.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
n	int	Standard
Calls		
Function	**Where Described**	
collect	Sec. 2.14.2.1.6	

Table 2.14-47: do_heap_collect CSU [8.14.2.1.18]

2.14.2.1.19 apply_to_blocks_of_size CSU

If USE_BBN_HEAP is defined, this CSU analyzes the memory blocks dependent on the function passed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
n	int	Standard
(*function)()	pointer to int	Standard
Calls		
Function	**Where Described**	
NEXT_BLOCK	Sec. 2.14.2.3	

Table 2.14-48: apply_to_blocks_of_size CSU [8.14.2.1.19]

2.14.2.2 create.c CSC

/simnet/libsrc/libheap/create.c

The CSUs in this CSC deal with heap creation and distruction.

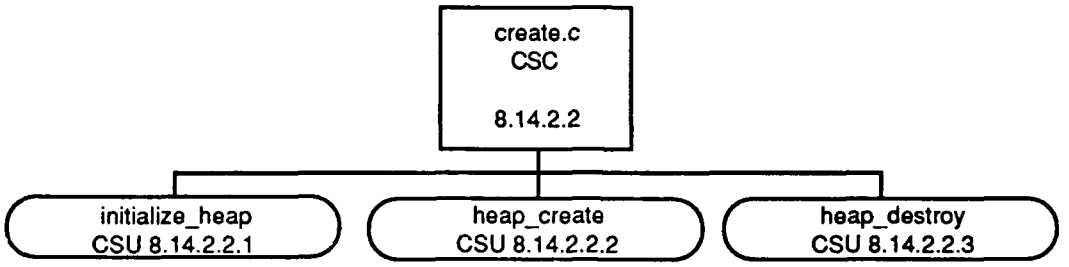


Figure 2.14-5: create.c CSC Structure

2.14.2.2.1 initialize_heap CSU

This CSU initializes a heap, passing in a pointer to its address, proc, and its size.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*address	pointer to char	Standard
proc	int	Standard
size	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
1	int	
Calls		
Function	**Where Described**	
init_free_list	Sec. 2.14.2.1.5	

Table 2.14-49: CSU [8.14.2.2.1]

2.14.2.2.2 heap_create CSU

If USE_BBN_HEAP is defined, this CSU creates a shared heap.

ReturnValues		
Return Value	**Type**	**Meaning**
1	int	heap created
Calls		
Function	**Where Described**	
initialize_heap	Sec. 2.14.2.2.1	

Table 2.14-50: heap_create CSU [8.14.2.2.2]

2.14.2.2.3 heap_destroy CSU

If USE_BBN_HEAP is defined, this CSU destroys a shared heap.

ReturnValues		
Return Value	Type	Meaning
1	int	heap destroyed

Table 2.14-51: heap_destroy CSU [8.14.2.2.3]

2.14.2.3 libheap.h CSU

/simnet/libsrc/libheap/libheap.h

This CSU contains the heap includes. Among these are the constant for the heap size, two structure definitions, and an expansion macro for NEXT_BLOCK(block), the block after the current one. With the exception of the macro, defined in Appendix A, tables for these items follow.

Constant	Value
HEAP_SIZE /* 16 meg */	(16 * 1024 * 1024)

Table 2.14-52: HEAP_SIZE Constant Definition

The following typedef struct is tagged block_header.

Item	Type	Where Type Defined
free	int	Standard
size	int	Standard
*next	pointer to struct block_header	This typedef struct

Table 2.14-53: BLOCK_HEADER Structure Definition

The following typedef struct is tagged heap.

Item	Type	Where Type Defined
*free_list	pointer to BLOCK_HEADER	The previous typedef struct
*heap_end	pointer to BLOCK_HEADER	The previous typedef struct
*data	pointer to BLOCK_HEADER	The previous typedef struct

Table 2.14-54: HEAP Structure Definition

2.14.3 libutil CSC

/simnet/libsrc/libutil

This library contains various utilities which are used by the simhost code.

2.14.3.1 args.c CSC

/simnet/libsrc/libutil/args.c

The CSC contains a single CSU to process switch arguments.

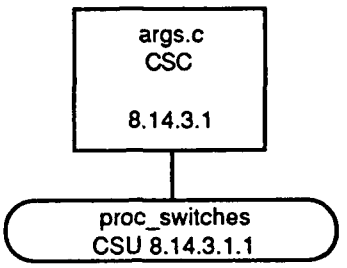


Figure 2.14-6: args.c CSC Structure

2.14.3.1.1 proc_switches CSU

This CSU processes switch arguments.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
table[ ][3]	int	Standard
argc	int	Standard
*argv[ ]	pointer to char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
i	int	index corresponding to a switch.

Table 2.14-55: proc_switches CSU [8.14.3.1.1]

2.14.3.2 checksum.c CSC

/simnet/libsrc/libutil/checksum.c

The CSC contains a single CSU to determine a checksum.

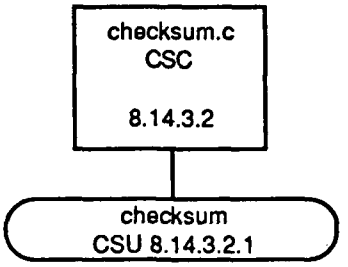


Figure 2.14-7: checksum.c CSC Structure

2.14.3.2.1 checksum CSU

This CSU calculates a checksum.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
argc	pointer to register unsigned short	Standard
nbytes	*int*	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
sum	unsigned short	Checksum

Table 2.14-56: checksum CSU [8.14.3.2.1]

2.14.3.3 time.c CSC

/simnet/libsrc/libutil/time.c

This CSC contains the macro CONVERT_FROM_SIXTEENTH_TIME(time) which divides the time value by shifting the bits down four places (see definition in Appendix A). In addition, it contains the following two CSUs. The manner of operation of these CSUs depend on whether CMU has been defined.

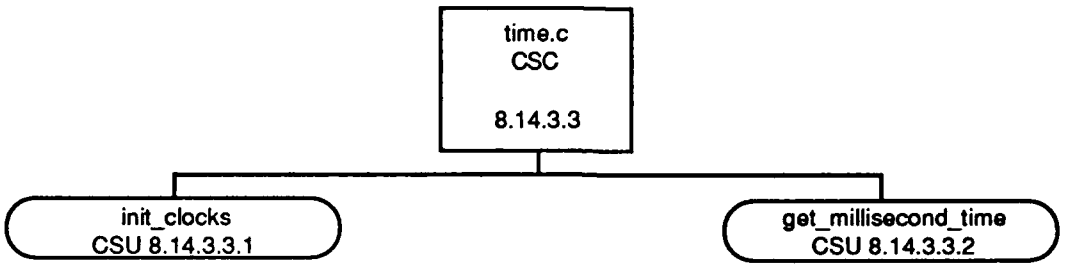


Figure 2.14-8: time.c CSC Structure

2.14.3.3.1 init_clocks CSU

If CMU is defined the base_time_value is determined by calling getrtc and dividing the returned binary value by 16, via four right bit shifts, using the macro CONVERT_FROM_SIXTEENTH_TIME. If not defined, a call to ftime(&tloc) provides a base_time_value equal to 1000 times tloc.time plus tloc.millithm.

Calls	
Function	**Where Described**
CONVERT_FROM_ SIXTEENTH_TIME	Sec. 2.2.14.3.3

Table 2.14-57: init_clocks CSU [8.14.3.3.1]

2.14.3.3.2 get_millisecond_time CSU

If CMU is defined, the g_last_millisecond_time is returned by calling getrtc, subtracting the base_time_value, and dividing the resultant binary value by 16, via four right bit shifts, using the macro CONVERT_FROM_SIXTEENTH_TIME. If CMU is not defined, the g_last_millisecond_time is returned by calling ftime(&tloc). g_last_millisecond_time is 1000 times tloc.time plus tloc.millithm, minus the base_time_value.

ReturnValues		
Return Value	**Type**	**Meaning**
g_last_millisecond_time	*unsigned int*	Time in milliseconds
Calls		
Function	**Where Described**	
CONVERT_FROM_ SIXTEENTH_TIME	Sec. 2.2.14.3.3	

Table 2.14-58: get_millisecond_time CSU [8.14.3.3.2]

2.14.3.4 bitfield.c CSC

/simnet/libsrc/libutil/bitfield.c

This CSC contains bitfield CSUs.

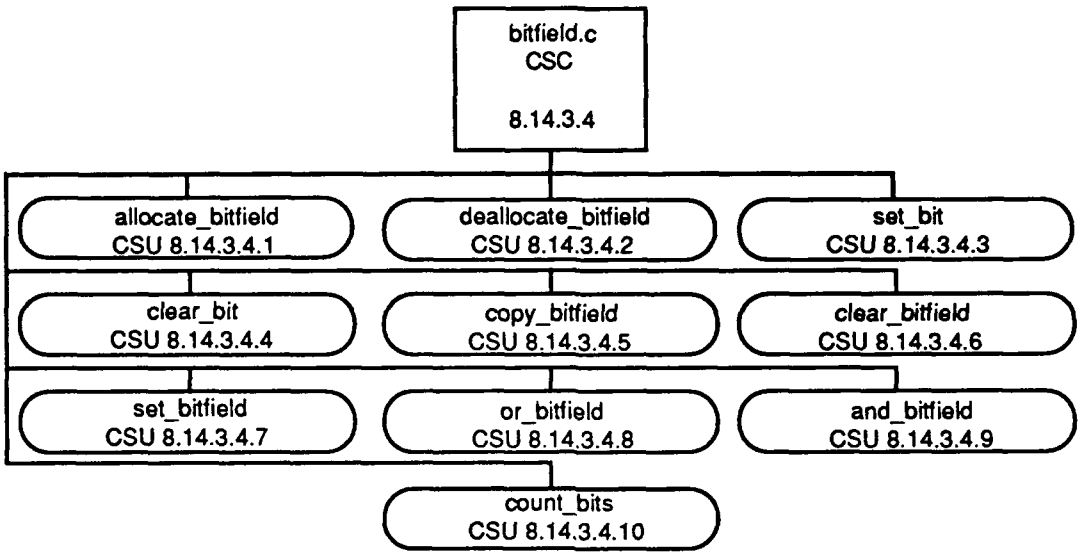


Figure 2.14-9: bitfield.c CSC Structure

2.14.3.4.1 allocate_bitfield CSU

This CSU allocates a bitfield through a call to heap_allocate and clears the memory.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
size	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
a	pointer to unsigned char	Allocated bitfield
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	

Table 2.14-59: allocate_bitfield CSU [8.14.3.4.1]

2.14.3.4.2 deallocate_bitfield CSU

This CSU releases a bitfield through a call to heap_deallocate.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*x	pointer to unsigned char	Standard
Calls		
Function	**Where Described**	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.14-60: deallocate_bitfield CSU [8.14.3.4.2]

2.14.3.4.3 set_bit CSU

This CSU sets a bit in a bitfield.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
bit_num	int	Standard
*bitfield	pointer to unsigned char	Standard

Table 2.14-61: set_bit CSU [8.14.3.4.3]

2.14.3.4.4 clear_bit CSU

This CSU clears a bit in a bitfield.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
bit_num	int	Standard
*bitfield	pointer to unsigned char	Standard

Table 2.14-62: clear_bit CSU [8.14.3.4.4]

2.14.3.4.5 copy_bitfield CSU

This CSU copies the contents of one portion of a bitfield to another.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*src	pointer to unsigned char	Standard
*dest	pointer to unsigned char	Standard
size	int	Standard

Table 2.14-63: copy_bitfield CSU [8.14.3.4.5]

2.14.3.4.6 clear_bitfield CSU

This CSU clears a length (size) of a bitfield through a call to bzero.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*src	pointer to unsigned char	Standard
size	int	Standard

Table 2.14-64: clear_bitfield CSU [8.14.3.4.6]

2.14.3.4.7 set_bitfield CSU

This CSU sets a bitfield by writing 0xFF into each byte.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*src	pointer to unsigned char	Standard
size	int	Standard

Table 2.14-65: set_bitfield CSU [8.14.3.4.7]

2.14.3.4.8 or_bitfield CSU

This CSU ORs two source bitfields and places the reults in a destination bitfield.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*src1	pointer to unsigned char	Standard
*src2	pointer to unsigned char	Standard
*dest	pointer to unsigned char	Standard
size	int	Standard

Table 2.14-66: or_bitfield CSU [8.14.3.4.8]

2.14.3.4.9 and_bitfield CSU

This CSU ANDs two source bitfields and places the reults in a destination bitfield.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*src1	pointer to unsigned char	Standard
*src2	pointer to unsigned char	Standard
*dest	pointer to unsigned char	Standard
size	int	Standard

Table 2.14-67: and_bitfield CSU [8.14.3.4.9]

2.14.3.4.10 count_bits CSU

This routine returns the number of bits counted.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
size	int	Standard
*bitfield	pointer to unsigned char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
count	int	Number of bits

Table 2.14-68: count_bits CSU [8.14.3.4.10]

2.14.3.5 math.c CSC

/simnet/libsrc/libutil/math.c

This CSC consists of a number of math CSUs.

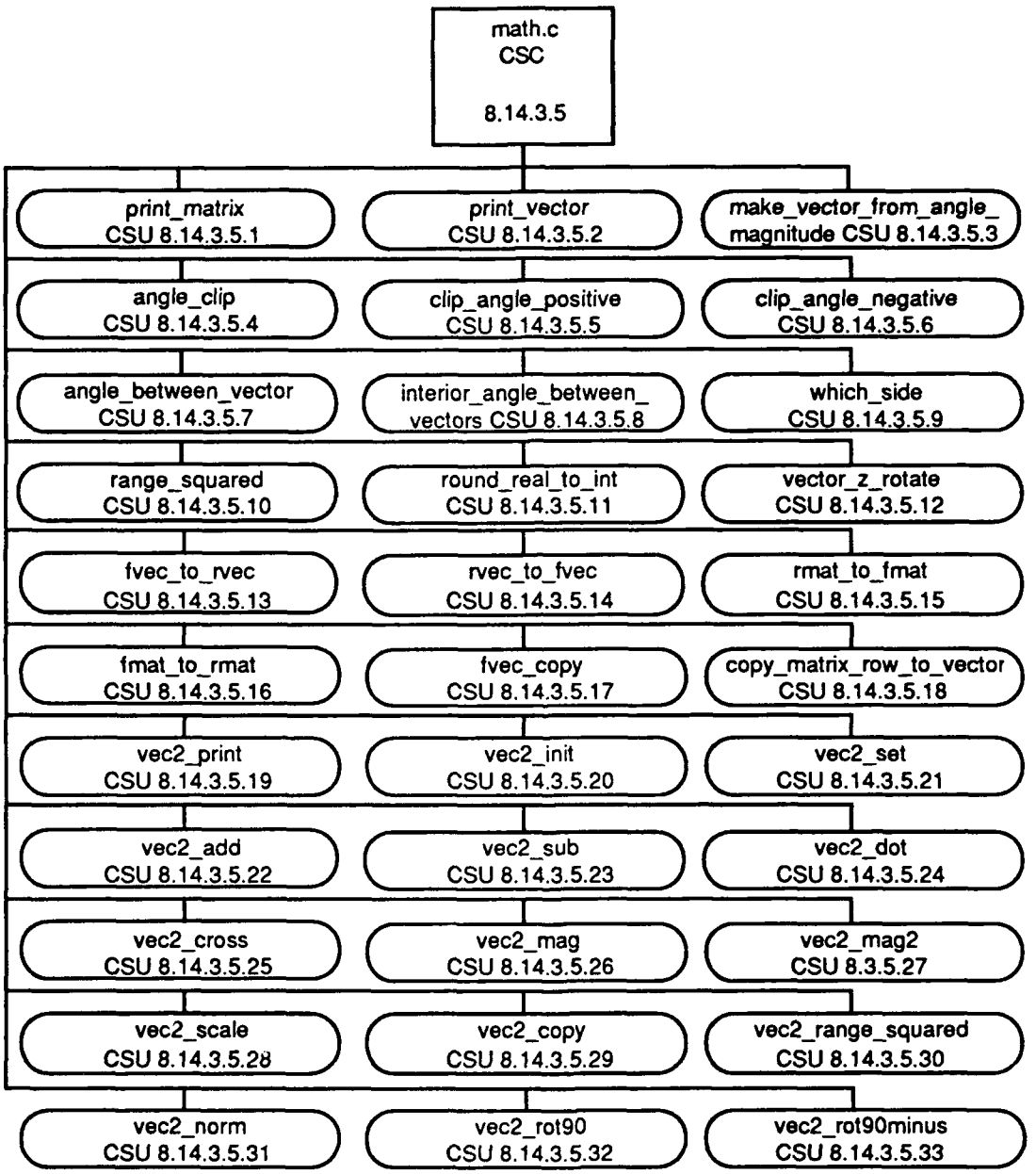


Figure 2.14-10: math.c CSC Structure

2.14.3.5.1 print_matrix CSU

This CSU prints the values in a 3 by 3 T_MATRIX passed as the parameter.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
m	T_MATRIX	sim_types.h

Table 2.14-69: print_matrix CSU [8.14.3.5.1]

2.14.3.5.2 print_vector CSU

This CSU prints the 3 values in a VECTOR passed as the sole parameter.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
v	VECTOR	sim_types.h

Table 2.14-70: print_vector CSU [8.14.3.5.2]

2.14.3.5.3 make_vector_from_angle_magnitude CSU

This CSU creates a vector by multiplying the sine (for X) and cosine (for Y) of the angle by the magnitude.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
theta	REAL	sim_types.h
mag	REAL	sim_types.h
result	VECTOR	sim_types.h

Table 2.14-71: make_vector_from_angle_magnitude CSU [8.14.3.5.3]

2.14.3.5.4 angle_clip CSU

This CSU returns 0.0 if the angle passed is 30.0 degrees or greater.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
angle	REAL	sim-_ypes.h
ReturnValues		
Return Value	**Type**	**Meaning**
0.0	REAL	angle is 30° or greater

Table 2.14-72: angle_clip CSU [8.14.3.5.4]

2.14.3.5.5 clip_angle_positive CSU

This CSU adjusts an angle in two pi increments until it becomes between zero and two pi.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
angle	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
angle	REAL	Resulting angle

Table 2.14-73: clip_angle_positive CSU [8.14.3.5.5]

2.14.3.5.6 clip_angle_negative CSU

This CSU adjusts an angle in two pi increments until it becomes between zero and minus two pi.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
angle	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
angle	REAL	Resulting angle

Table 2.14-74: clip_angle_negative CSU [8.14.3.5.6]

2.14.3.5.7 angle_between_vectors CSU

This CSU returns the angle, along with sign formed by two vectors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
v1	VECTOR	sim_types.h
v2	VECTOR	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
(which_side(v1,v2) * interior_ angle_between_vectors (v1,v2))	REAL	Signed angle between two vectors
Calls		
Function	**Where Described**	
which_side	Sec. 2.14.3.5.9	
interior_angle_between_ vectors	Sec. 2.14.3.5.8	

Table 2.14-75: angle_between_vectors CSU [8.14.3.5.7]

2.14.3.5.8 interior_angle_between_vectors CSU

This CSU returns 1 if vec2 is left of vec1, -1 if vec2 is right of vec1, and 0 if vec2 is colinear with vec1.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*vec1	pointer to REAL	sim_types.h
*vec2	pointer to REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
(0.0)	REAL	vec2 is colinear with vec1
(acos(RANGE_CLIP(-1.0, quot, 1.0)))	REAL	1 if vec2 is left of vec1 -1 if vec2 is right of vec1
Calls		
Function	**Where Described**	
almost_eq	Sec. 2.14.3.9	
RANGE_CLIP	Sec. 2.14.3.9	

Table 2.14-76: interior_angle_between_vectors CSU [8.14.3.5.8]

2.14.3.5.9 which_side CSU

This CSU returns the sign of the difference of two cross multiplied vectors: 1 if vec2 is left of vec1; -1 if vec2 is right of vec1; and 0 if the two vectors are colinear.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*vec1	pointer to REAL	sim_types.h
*vec2	pointer to REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
(sign(vec2[Y]*vec1[X] - vec2[X]*vec1[Y])	int	1 if vec2 is left of vec1; -1 if vec2 is right of vec1; and 0 if the two vectors are colinear

Table 2.14-77: which_side CSU [8.14.3.5.9]

2.14.3.5.10 range_squared CSU

This CSU returns the sum of the squares of the differences of two, three element vectors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*vec1	pointer to register REAL	sim_types.h
*vec2	pointer to register REAL	sim_types.h

ReturnValues		
Return Value	**Type**	**Meaning**
(square(delta_x) + square(delta_y) + square(delta_z))	REAL	sum of the squares of the differences of two vectors

Table 2.14-78: range_squared CSU [8.14.3.5.10]

2.14.3.5.11 round_real_to_int CSU

This CSU converts a REAL into a rounded integer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*value	pointer to REAL	sim_types.h
angle	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
(retval/modulus)	int	Rounded integer value

Table 2.14-79: round_real_to_int CSU [8.14.3.5.11]

2.14.3.5.12 vector_z_rotate CSU

This CSU rotates a two dimensional vector by "angle" radians around the origin (with respect to the Z axis).

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*vec	pointer to REAL	sim_types.h
angle	REAL	sim_types.h
Calls		
Function	**Where Described**	
mat_rot_init	Sec. 2.6.2.47.1 Vehicles CSCI SDD	
vec_mat_mul	Sec. 2.6.2.56.1 Vehicles CSCI SDD	

Table 2.14-80: vector_z_rotate CSU [8.14.3.5.12]

2.14.3.5.13 fvec_to_rvec CSU

This CSU converts a floating point vector to a REAL vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*float_vector	pointer to register float	Standard
*real_vector	pointer to register REAL	sim_types.h

Table 2.14-81: fvec_to_rvec CSU [8.14.3.5.13]

2.14.3.5.14 rvec_to_fvec CSU

This CSU converts a REAL vector to a floating point vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*real_vector	pointer to register REAL	sim_types.h
*float_vector	pointer to register float	Standard

Table 2.14-82: rvec_to_fvec CSU [8.14.3.5.14]

2.14.3.5.15 rmat_to_fmat CSU

This CSU converts a REAL matrix to a floating point matrix.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
real_mat[ ]	register REAL	sim_types.h
float_mat[ ]	register float	Standard

Table 2.14-83: rmat_to_fmat CSU [8.14.3.5.15]

2.14.3.5.16 fmat_to_rmat CSU

This CSU converts a floating point matrix to a REAL matrix.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
float_mat[ ]	register float	Standard
real_mat[ ]	register REAL	sim_types.h

Table 2.14-84: fmat_to_rmat CSU [8.14.3.5.16]

2.14.3.5.17 fvec_copy CSU

This CSU copies the three values in the source vector into the destination vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*src_vec	pointer to register float	Standard
*dst_vec	pointer to register float	Standard

Table 2.14-85: fvec_copy CSU [8.14.3.5.17]

2.14.3.5.18 copy_matrix_row_to_vector CSU

This CSU copies the three values in a matrix row into a vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
src_mat	register T_MATRIX	sim_types.h
row	register int	Standard
dst_vec	register VECTOR	sim_types.h

Table 2.14-86: copy_matrix_row_to_vector CSU [8.14.3.5.18]

2.14.3.5.19 vec2_print CSU

This and the remaining CSUs in this file deal with vectors. This one prints the two values x, the REAL pointer passed, and y, the REAL value pointed at by *y.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*v	pointer to REAL	sim_types.h

Table 2.14-87: vec2_print CSU [8.14.3.5.19]

2.14.3.5.20 vec2_init CSU

This CSU initiates the two values of a vector to zero.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
v	VECTOR	sim_types.h

Table 2.14-88: vec2_init CSU [8.14.3.5.20]

2.14.3.5.21 vec2_set CSU

This CSU set two values of a vector (v[0] and v[1]) to x and y respectively.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
v	VECTOR	sim_types.h
x	REAL	sim_types.h
y	REAL	sim_types.h

Table 2.14-89: vec2_set CSU [8.14.3.5.21]

2.14.3.5.22 vec2_add CSU

This CSU adds the vector values in v1 and v2, placing the results in vr.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
v1	VECTOR	sim_types.h
v2	VECTOR	sim_types.h
vr	VECTOR	sim_types.h

Table 2.14-90: vec2_add CSU [8.14.3.5.22]

2.14.3.5.23 vec2_sub CSU

This CSU subtracts the vector values in v2 from v1, puting the results in vr.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
v1	VECTOR	sim_types.h
v2	VECTOR	sim_types.h
vr	VECTOR	sim_types.h

Table 2.14-91: vec2_sub CSU [8.14.3.5.23]

2.14.3.5.24 vec2_dot CSU

This routine returns the quantity which results when two vectors are multiplied and summed with the product of their z components.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*v1	pointer to REAL	sim_types.h
*v2	pointer to REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
*v1 * *v2 + v1[Y] * v2[Y]	REAL	See description above

Table 2.14-92: vec2_dot CSU [8.14.3.5.24]

2.14.3.5.25 vec2_cross CSU

This routine returns a value which is equal to the difference between the product of ref and the z component of v and the product of v and the z component of ref.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*ref	pointer to REAL	sim_types.h
*v	pointer to REAL	sim_types.h

ReturnValues		
Return Value	**Type**	**Meaning**
*ref * v[Y] - ref[Y] * *v	REAL	the value of this expression

Table 2.14-93: vec2_cross CSU [8.14.3.5.25]

2.14.3.5.26 vec2_mag CSU

This CSU returns the square root of the sum of the squares of the vector and magnitude.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*v	pointer to REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
sqrt(*v * *v + v[Y] * v[Y])	REAL	Square root of the sum of the squares of the vector and magnitude

Table 2.14-94: vec2_mag CSU [8.14.3.5.26]

2.14.3.5.27 vec2_mag2 CSU

This CSU returns the sum of the squares of the vector and magnitude.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*v	pointer to REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
(*v * *v + v[Y] * v[Y])	REAL	Sum of the squares of the vector and magnitude

Table 2.14-95: vec2_mag2 CSU [8.14.3.5.27]

2.14.3.5.28 vec2_scale CSU

This CSU multiplies the elements in vector v by s, placing the results in vector r.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
v	VECTOR	sim_types.h
r	VECTOR	sim_types.h
s	REAL	sim_types.h

Table 2.14-96: vec2_scale CSU [8.14.3.5.28]

2.14.3.5.29 vec2_copy CSU

This CSU copies vector a to vector b.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
a	VECTOR	sim_types.h
b	VECTOR	sim_types.h

Table 2.14-97: vec2_copy CSU [8.14.3.5.29]

2.14.3.5.30 vec2_range_squared CSU

This CSU returns the sum of the squares of the differences of the two vectors.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*vec1	pointer to register REAL	sim_types.h
*vec2	pointer to register REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
(square(delta_x) + square(delta_y))	REAL	Sum of the squares of the differences of the vectors

Table 2.14-98: vec2_range_squared CSU [8.14.3.5.30]

2.14.3.5.31 vec2_norm CSU

This CSU artificially normalizes a vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
v	VECTOR	sim_types.h
r	VECTOR	sim_types.h
abort	int	Standard

Table 2.14-99: vec2_norm CSU [8.14.3.5.31]

2.14.3.5.32 vec2_rot90 CSU

This CSU manipulates a vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*v	pointer to REAL	sim_types.h
*r	pointer to REAL	sim_types.h

Table 2.14-100: vec2_rot90 CSU [8.14.3.5.32]

2.14.3.5.33 vec2_rot90minus CSU

This CSU manipulates a vector.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*v	pointer to REAL	sim_types.h
*r	pointer to REAL	sim_types.h

Table 2.14-101: vec2_rot90minus CSU [8.14.3.5.33]

2.14.3.6 geometry.c CSC

/simnet/libsrc/libutil/geometry.c

This CSC contains geometry CSUs.

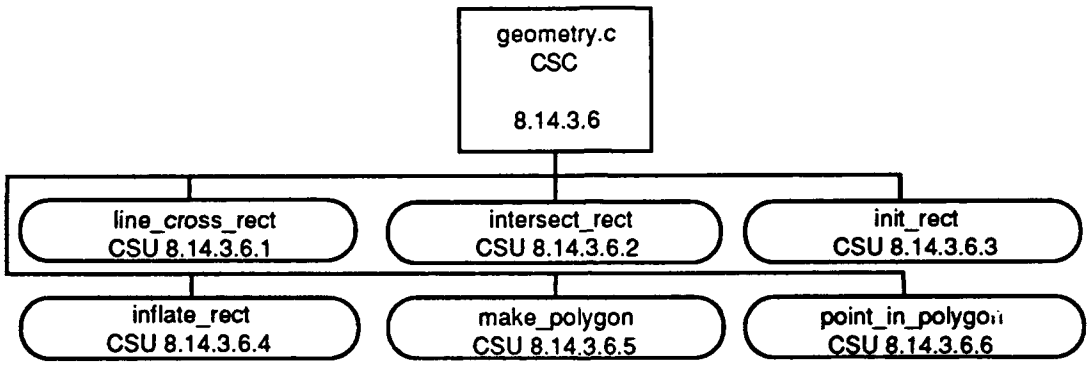


Figure 2.14-11: geometry.c CSC Structure

2.14.3.6.1 line_cross_rect CSU

For a given position and destination, this CSU return TRUE if a line segment intersects a rectangle.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*pos	pointer to REAL	sim_types.h
*dest	pointer to REAL	sim_types.h
*rect_ptr	pointer to REAL	sim_types.h

ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	Line segment intersects rectangle
FALSE	int	Line segment doesn't intersect rectangle

Calls	
Function	**Where Described**
min	Sec. 2.6.7.3
max	Sec. 2.6.7.3
intersect_rect	Sec. 2.14.3.6.2

Table 2.14-102: line_cross_rect CSU [8.14.3.6.1]

2.14.3.6.2 intersect_rect CSU

For two given rectangles, this CSU return TRUE if the rectangles intersect and returnsFALSE otherwise.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*r1	pointer to RECT	Sec. 2.14.3.9
*r2	pointer to RECT	Sec. 2.14.3.9

ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	rectangles intersected
FALSE	int	rectangles not intersected

Calls	
Function	**Where Described**
min	Sec. 2.6.7.3
max	Sec. 2.6.7.3

Table 2.14-103: intersect_rect CSU [8.14.3.6.2]

2.14.3.6.3 init_rect CSU

This CSU sets all elements in a rectangle to zero.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*rect_ptr	pointer to RECT	Sec. 2.14.3.9

Table 2.14-104: init_rect CSU [8.14.3.6.3]

2.14.3.6.4 inflate_rect CSU

This CSU increases the sides of a rectangle, inflating it, by the "amount" passed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*rect_ptr	pointer to RECT	Sec. 2.14.3.9
amount	REAL	sim_types.h

Table 2.14-105: inflate_rect CSU [8.14.3.6.4]

2.14.3.6.5 make_polygon CSU

This CSU creates a polygon.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
num_vertices	int	Standard
array_rep[MAX_POLGON_ VERTICES][2]	register REAL	sim_types.h
*poly_rep	pointer to register POLYGON	Sec. 2.14.3.9

Table 2.14-106: make_polygon CSU [8.14.3.6.5]

2.14.3.6.6 point_in_polygon CSU

This CSU determines if we are within a given polygon. A line segment is drawn from the current position to +infinity. If the line segment crosses an odd number of polygon line segments, then the current position is within the polygon.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*point	pointer to register REAL	sim-types.h
*polygon	pointer to register POLYGON	Sec. 2.14.3.9
ReturnValues		
Return Value	**Type**	**Meaning**
FALSE	int	not within polygon
(num_crossed % 2)	int	if odd, within polygon if even, not within polygon
Calls		
Function	**Where Described**	
point_in_rect	Sec. 2.	

Table 2.14-107: point_in_polygon CSU [8.14.3.6.6]

2.14.3.7 random.c CSC

/simnet/libsrc/libutil/random.c

This CSC contains random number utilities.

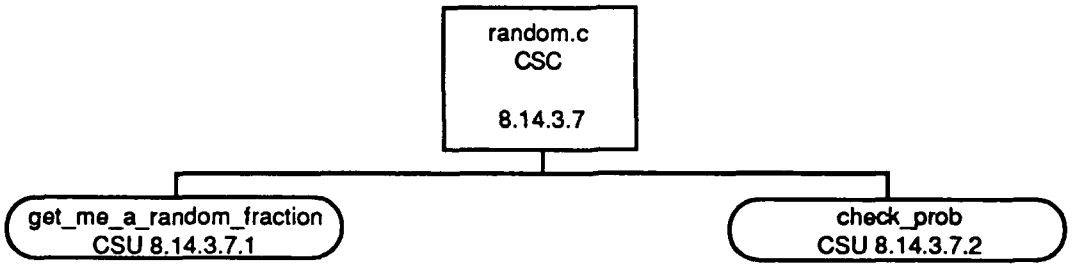


Figure 2.14-12: random.c CSC Structure

2.14.3.7.1 get_me_a_random_fraction CSU

This CSU calls rand to get a random (short) number and divides it (in REAL form) by 65535 to gat a random fraction.

ReturnValues		
Return Value	**Type**	**Meaning**
reandom_fraction	REAL	Random fraction

Table 2.14-108: get_me_a_random_fraction CSU [8.14.3.7.1]

2.14.3.7.2 check_prob CSU

This CSU returns TRUE if the fraction passed is equal to or grater than the random fraction generated, else it return FALSE.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
p	REAL	sim_types.h
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	p equal to or larger than random fraction
FALSE	int	p smaller than random fraction

Table 2.14-109: check_prob CSU [8.14.3.7.2]

2.14.3.8 cstring.c CSC

/simnet/libsrc/libutil/cstring.c

This CSC contains case insensitive string utilities.

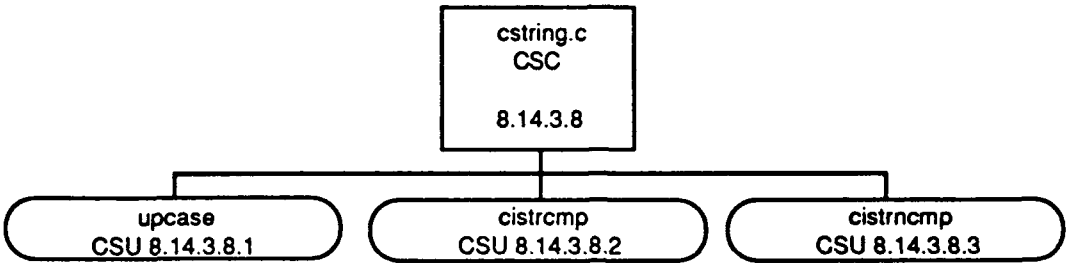


Figure 2.14-13: cstring.c CSC Structure

2.14.3.8.1 upcase CSU

This CSU compares characters converting lower case alphabetic characters to upper case. Other characters are not changed.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
c	register char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
c	char	Upper case or unconverted character

Table 2.14-110: upcase CSU [8.14.3.8.1]

2.14.3.8.2 cistrcmp CSU

This CSU performs a case independent string compare.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*a	pointer to register char	Standard
*b	pointer to register char	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
1	int	Characters compare
0	int	Characters different
Calls		
Function	**Where Described**	
upcase	Sec. 2.14.3.8.1	

Table 2.14-111: cistrcmp CSU [8.14.3.8.2]

2.14.3.8.3 cistrncmp CSU

This CSU performs a case independent string compare of "n" characters length.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*a	pointer to register char	Standard
*b	pointer to register char	Standard
n	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
1	int	Characters compare
0	int	Characters different
Calls		
Function	**Where Described**	
upcase	Sec. 2.14.3.8.1	

Table 2.14-112: cistrncmp CSU [8.14.3.8.3]

2.14.3.9 libutil.h CSU

/simnet/libsrc/libutil/libutil.h

This CSU contains the utility library macros (defined in Appendix A), constants (fractional values of PI, rectangle processing definitions, and the maximum number of polgon vertices), and two geometric structure definitions. The constants and structures are found in the following three tables.

Constant	Value
PI_OVER_2 /* Fractional values of PI */	1.5707963267948965000
PI_OVER_3	1.0471975511965976000
PI_OVER_4	0.7853981633974482700
PI_OVER_6	0.5235987755982987900
PI_OVER_8	0.3926990816987241400
UPPER_LEFT /* Rectangle definitions */	0x01
UPPER_RIGHT	0x02
LOWER_RIGHT	0x04
LOWER_LEFT	0x08
MAX_POLYGON_VERTICES	11

Table 2.14-113: libutil.h Constant Definitions

Item	Type	Where Type Defined
left	REAL	sim_types.h
top	REAL	sim_types.h
right	REAL	sim_types.h
bottom	REAL	sim_types.h

Table 2.14-114: RECT Structure Definition

Item	Type	Where Type Defined
boundaries	RECT	The previous typedef struct
num_vertices	int	Standard
vertices[MAX_POLYGON_VERTICES][2]	REAL	sim_types.h

Table 2.14-115: POLYGON Structure Definition

2.14.4 libcomm CSC

/simnet/libsrc/libcomm

This is a library of list, queue, and tree handling code. This is also where the performance monitoring code is defined.

2.14.4.1 bufpool.c CSC

/simnet/libsrc/libcomm/bufpool.c

This CSC contains CSUs dealing with buffer queues. In additon, a number of macros are defined. These are dewscribed in Appendix A.

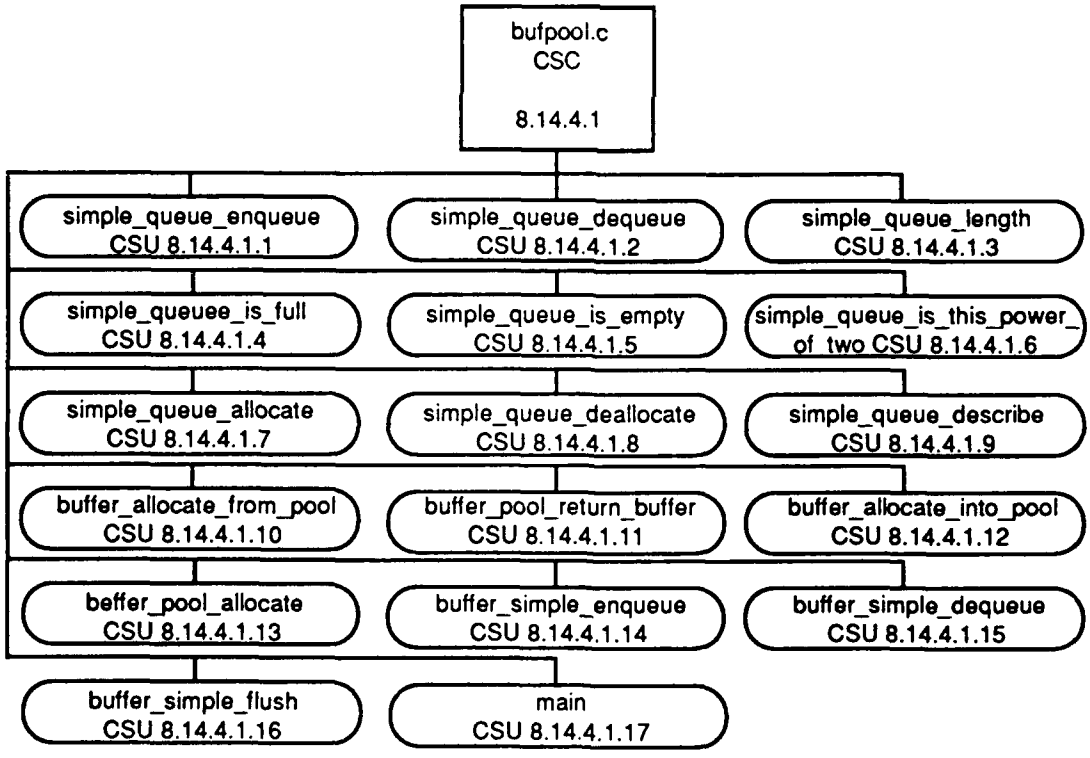


Figure 2.14-14: bufpool.c CSC Structure

2.14.4.1.1 simple_queue_enqueue CSU

This CSU enqueues an item to a buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to register SIMPLE_QUEUE	Sec. 2.14.4.3
item	POINTER	sim_stype.h
ReturnValues		
Return Value	**Type**	**Meaning**
ret	int	TRUE - item enqueued FALSE - buffer full
Calls		
Function	**Where Described**	
SQLOCK	Sec. 2.14.4.1	
SQFULL	Sec. 2.14.4.1	
SQNQ	Sec. 2.14.4.1	
SQUNLOCK	Sec. 2.14.4.1	

Table 2.14-116: simple_queue_enqueue CSU [8.14.4.1.1]

2.14.4.1.2 simple_queue_dequeue CSU

This CSU dequeues the buffer. It returns NULL if the buffer was empty and returns the contents of the buffer otherwise.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to register SIMPLE_QUEUE	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
NULL	POINTER	buffer was empty
ret	POINTER	contents of buffer
Calls		
Function	**Where Described**	
SQLOCK	Sec. 2.14.4.1	
SQFULL	Sec. 2.14.4.1	
SQNQ	Sec. 2.14.4.1	
SQUNLOCK	Sec. 2.14.4.1	

Table 2.14-117: simple_queue_dequeue CSU [8.14.4.1.2]

2.14.4.1.3 simple_queue_length CSU

This CSUteturns the size of the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to register SIMPLE_QUEUE	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
length	int	size of buffer
Calls		
Function	**Where Described**	
SQLOCK	Sec. 2.14.4.1	
SQLENGTH	Sec. 2.14.4.1	
SQUNLOCK	Sec. 2.14.4.1	

Table 2.14-118: simple_queue_length CSU [8.14.4.1.3]

2.14.4.1.4 simple_queue_is_full CSU

This CSU determines if the buffer is full.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to register SIMPLE_QUEUE	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
ret	int	queue is full
Calls		
Function	**Where Described**	
SQLOCK	Sec. 2.14.4.1	
SQFULL	Sec. 2.14.4.1	
SQUNLOCK	Sec. 2.14.4.1	

Table 2.14-119: simple_queue_is_full CSU [8.14.4.1.4]

2.14.4.1.5 simple_queue_is_empty CSU

This CSU determines if the buffer is empty.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to SIMPLE_QUEUE	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
ret	int	buffer is empty
Calls		
Function	**Where Described**	
SQLOCK	Sec. 2.14.4.1	
SQEMPTY	Sec. 2.14.4.1	
SQUNLOCK	Sec. 2.14.4.1	

Table 2.14-120: simple_queue_is_empty CSU [8.14.4.1.5]

2.14.4.1.6 simple_queue_is_this_power_of_two CSU

This CSU determines if the buffer size, in bits, is equal to a power of two.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
l	unsigned long	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
TRUE	int	number of bits is equal to a power of two
FALSE	int	number of bits not equal to a power of two

Table 2.14-121: simple_queue_is_this_power_of_two CSU [8.14.4.1.6]

2.14.4.1.7 simple_queue_allocate CSU

This CSU allocates the buffer space for the buffer of length size.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
size	unsigned long	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
sqp	pointer to SIMPLE_QUEUE	pointer to contents of buffer
NULL	pointer to SIMPLE_QUEUE	queue is empty

Calls	
Function	**Where Described**
simple_queue_is_this_power_of_two	Sec. 2.14.4.1.6
heap_allocate	Sec. 2.14.2.1.3

Table 2.14-122: simple_queue_allocate CSU [8.14.4.1.7]

2.14.4.1.8 simple_queue_deallocate CSU

This CSU deallocates the buffer space.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to SIMPLE_QUEUE	Sec. 2.14.4.3

Calls	
Function	**Where Described**
SQLOCK	Sec. 2.14.4.1
heap_deallocate	Sec. 2.2.14.2.1.2

Table 2.14-123: simple_queue_deallocate CSU [8.14.4.1.8]

2.14.4.1.9 simple_queue_describe CSU

This CSU orints statistics about the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to SIMPLE_QUEUE	Sec. 2.14.4.3

Calls	
Function	**Where Described**
SQLENGTH	Sec. 2.14.4.1
SQUNLOCK	Sec. 2.14.4.1

Table 2.14-124: simple_queue_describe CSU [8.14.4.1.9]

2.14.4.1.10 buffer_allocate_from_pool CSU

This CSU allocates a buffer from the pool of available buffers.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
buf_pool	pointer to BUFFER_POOL	Sec. 2.14.4.3

ReturnValues		
Return Value	**Type**	**Meaning**
p	BYTE_POINTER	pointer to buffer from pool
NULL	BYTE_POINTER	couldn't get buffer from pool
Calls		
Function	**Where Described**	
simple_queue_dequeue	Sec. 2.14.4.1.2	
char_from_buffer	Sec. 2.14.4.3	

Table 2.14-125: buffer_allocate_from_pool CSU [8.14.4.1.10]

2.14.4.1.11 buffer_pool_return_buffer CSU

This CSU enqueues a pool buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
bp	pointer to BUFFER	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
simple_queue_enqueue	Sec. 2.14.4.1.1	

Table 2.14-126: buffer_pool_return_buffer CSU [8.14.4.1.11]

2.14.4.1.12 buffer_allocate_into_pool CSU

This CSU allocates a buffer into the available buffer pool.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
buf_pool	pointer to BUFFER_POOL	Sec. 2.14.4.3
size	int	Standard
Calls		
Function	**Where Described**	
buffer_allocate	Sec. 2.14.4.2.12	
char_from_buffer	Sec. 2.14.4.3	
buffer_one_less	Sec. 2.14.4.2.29	
buffer_pool_return_buffer	Sec. 2.14.4.1.11	

Table 2.14-127: buffer_allocate_into_pool CSU [8.14.4.1.12]

2.14.4.1.13 buffer_pool_allocate CSU

This CSU allocates the entire buffer pool.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
size	int	Standard
number	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
bufpool	BUFFER_POOL	pointer to pool of buffers
Calls		
Function	**Where Described**	
simple_queue_allocate	Sec. 2.14.4.1.1	
buffer_allocate_into_pool	Sec. 2.14.4.1.12	

Table 2.14-128: buffer_pool_allocate CSU [8.14.4.1.13]

2.14.4.1.14 buffer_simple_enqueue CSU

This CSU enqueues the buffer and increments thje pointer to point to the next available location.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to SIMPLE_QUEUE	Sec. 2.14.4.3
p	BYTE_POINTER	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
successp	int	indicates success or failure
Calls		
Function	**Where Described**	
buffer_from_char	Sec. 2.14.4.3	
simple_queue_enque	Sec. 2.14.1.1	

Table 2.14-129: buffer_simple_enqueue CSU [8.14.4.1.14]

2.14.4.1.15 buffer_simple_dequeue CSU

This CSU dequeues the buffer and returns the contents of the buffer. If the buffer was empty, it returns NULL.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to SIMPLE_QUEUE	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
p	BYTE_POINTER	contents of buffer
NULL	BYTE_POINTER	buffer empty
Calls		
Function	**Where Described**	
simple_queue_dequeue	Sec. 2.14.4.1.1	
char_from_buffer	Sec. 2.14.4.3	

Table 2.14-130: buffer_simple_dequeue CSU [8.14.4.1.15]

2.14.4.1.16 buffer_simple_flush CSU

This CSU floushes the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*sqp	pointer to SIMPLE_QUEUE	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
buffer_simple_dequeue	Sec. 2.14.4.1.16	
buffer_deallocate	Sec. 2.14.4.3.15	

Table 2.14-131: buffer_simple_flush CSU [8.14.4.1.16]

2.14.4.1.17 main CSU

This is the top level routine for this module. It coordinates the allocation of the buffer pool.

Calls	
Function	**Where Described**
buffer_pool_allocate	Sec. 2..14.1.13
buffer_allocate_from_pool	Sec. 2.14.4.1.10

Table 2.14-132: main CSU [8.14.4.1.17]

2.14.4.2 queue_fix.c CSC

/simnet/libsrc/libcomm/queue_fix.c

This CSC contains CSUs dealing with locked queues. In additon, there are two macros whose expansion definition are conditional on MULTIPROCESSOR being defined. These are QLOCK(qp) and QUNLOCK(qp). These macros are described in Appendix A.

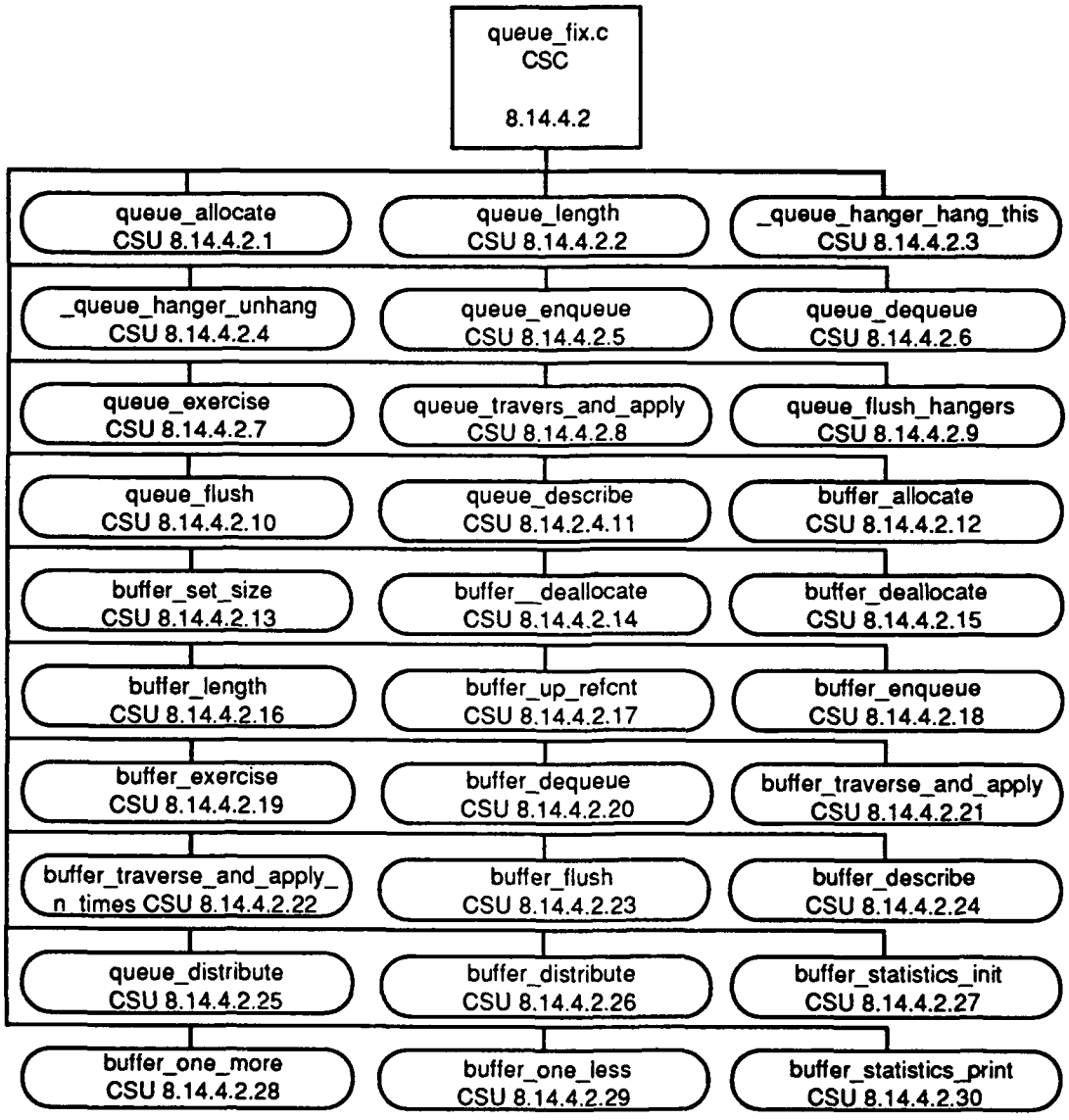


Figure 2.14-15: queue_fix.c CSC Structure

2.14.4.2.1 queue_allocate CSU

This CSU allocates the buffer.

ReturnValues		
Return Value	**Type**	**Meaning**
qp	pointer to QUEUE	
Calls		
Function	**Where Described**	
buffer_allocate	Sec. 2.14.4.2.12	
QUNLOCK	Sec. 2.14.4.3	
QLOCK	Sec. 2.14.4.3	

Table 2.14-133: queue_allocate CSU [8.14.4.2.1]

2.14.4.2.2 queue_length CSU

This CSU returns the length of the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to register QUEUE	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
ret	int	Length
Calls		
Function	**Where Described**	
QLOCK	Sec. 2.14.4.2 See Appendix A	
QUNLOCK	Sec. 2.14.4.2 See Appendix A	

Table 2.14-134: queue_length CSU [8.14.4.2.2]

2.14.4.2.3 _queue_hanger_hang_this CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*p	pointer to BUFFER	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
h	pointer to QELEMENT	
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	

Table 2.14-135: _queue_hanger_hang_this CSU [8.14.4.2.3]

2.14.4.2.4 _queue_hanger_unhang CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*p	pointer to BUFFER	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
p	pointer to BUFFER	
Calls		
Function	**Where Described**	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.14-136: _queue_hanger_unhang CSU [8.14.4.2.4]

2.14.4.2.5 queue_enqueue CSU

This CSU enqueues the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to register QUEUE	Sec. 2.14.4.3
*bufp	pointer to BUFFER	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
QLOCK	Sec. 2.14.4.3	
QUNLOCK	Sec. 2.14.4.3	
queue_hanger_hang_this	Sec. 2.14.4.2.7	

Table 2.14-137: queue_enqueue CSU [8.14.4.2.5]

2.14.4.2.6 queue_dequeue CSU

This CSU dequeues the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to register QUEUE	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
ret	pointer to BUFFER	pointer to contents of buffer
Calls		
Function	**Where Described**	
QLOCK	Sec. 2.14.4.3	
QUNLOCK	Sec. 2.14.4.3	
queue_hanger_unhang	Sec. 2.14.4.2.7	

Table 2.14-138: queue_dequeue CSU [8.14.4.2.6]

2.14.4.2.7 queue_excise CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to register QUEUE	Sec. 2.14.4.3
*p	pointer to register BUFFER	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
ret	pointer to BUFFER	
Calls		
Function	**Where Described**	
_queue_hanger_unhang	Sec. 2.14.4.2.4	
QUNLOCK	Sec. 2.14.4.3	

Table 2.14-139: queue_exercise CSU [8.14.4.2.7]

2.14.4.2.8 queue_traverse_and_apply CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to register QUEUE	Sec. 2.14.4.3
fcn	PFI	sim_types.h
arg	not defined	N/A
ReturnValues		
Return Value	**Type**	**Meaning**
count	int	1 if "hit" 0 if "miss"
Calls		
Function	**Where Described**	
QLOCK	Sec. 2.14.4.3	
QUNLOCK	Sec. 2.14.4.3	

Table 2.14-140: queue_traverse_and_apply CSU [8.14.4.2.8]

2.14.4.2.9 queue_flush_hangers CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qep	pointer to QELEMENT	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
queue_flush_hangers	Sec. 2.14.4.2.9	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.14-141: queue_flush_hangers CSU [8.14.4.2.9]

2.14.4.2.10 queue_flush CSU

This CSU flushes the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to register QUEUE	Sec. 2.14.4.3
deallocate_fcn	PFV	sim_types.h
arg	not defined	N/A
Calls		
Function	**Where Described**	
QLOCK	Sec. 2.14.4.3	
QUNLOCK	Sec. 2.14.4.3	
queue_flush_hangers	Sec. 2.14.4.2.9	

Table 2.14-142: queue_flush CSU [8.14.4.2.10]

2.14.4.2.11 queue_describe CSU

This CSU prints information about the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to QUEUE	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
QLOCK	Sec. 2.14.4.3	
QUNLOCK	Sec. 2.14.4.3	

Table 2.14-143: queue_describe CSU [8.14.4.2.11]

2.14.4.2.12 buffer_allocate CSU

This CSU allocates the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
size	int	Standard
ReturnValues		
Return Value	**Type**	**Meaning**
(char_from_buffer(bp)	BYTE_POINTER	
Calls		
Function	**Where Described**	
heap_allocate	Sec. 2.14.2.1.1	
buffer_one_more	Sec. 2.14.4.2.28	
char_from_buffer	Sec. 2.14.4.3	

Table 2.14-144: queue_describe CSU [8.14.4.2.12]

2.14.4.2.13 buffer_set_size CSU

This CSU sets the size of the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
p	BYTE_POINTER	Sec. 2.14.4.3
new_size	int	Standard
Calls		
Function	**Where Described**	
buffer_from_char	Sec. 2.14.4.3	

Table 2.14-145: buffer_set_size CSU [8.14.4.2.13]

2.14.4.2.14 buffer__deallocate CSU

This CSU deallocates the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*bp	pointer to register BUFFER	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
buffer_one_less	Sec. 2.14.4.2.29	
buffer_pool_return_buffer	Sec. 2.14.4.1.11	
heap_deallocate	Sec. 2.14.2.1.4	

Table 2.14-146: buffer_deallocate CSU [8.14.4.2.14]

2.14.4.2.15 buffer_deallocate CSU

This CSU deallocates the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
p	BYTE_POINTER	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
buffer_deallocate	Sec. 2.14.4.2.15	
buffer_from_char	Sec. 2.14.4.3	

Table 2.14-147: buffer_deallocate CSU [8.14.4.2.15]

2.14.4.2.16 buffer_length CSU

This CSU returns the length of the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
p	BYTE_POINTER	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
bp->length	int	length of buffer
Calls		
Function	**Where Described**	
buffer_from_char	Sec. 2.14.4.3	

Table 2.14-148: buffer_length CSU [8.14.4.2.16]

2.14.4.2.17 buffer_up_refcnt CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
p	BYTE_POINTER	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
buffer_from_char	Sec. 2.14.4.3	

Table 2.14-149: buffer_up_refcnt CSU [8.14.4.2.17]

2.14.4.2.18 buffer_enqueue CSU

This CSU enqueues the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to QUEUE	Sec. 2.14.4.3
p	BYTE_POINTER	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
buffer_from_char	Sec. 2.14.4.3	
queue_enque	Sec. 2.14.4.2.5	

Table 2.14-150: buffer_enqueue CSU [8.14.4.2.18]

2.14.4.2.19 buffer_excise CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to QUEUE	Sec. 2.14.4.3
p	BYTE_POINTER	Sec. 2.14.4.3
ReturnValues		
Return Value	**Type**	**Meaning**
foo	BYTE_POINTER	buffer excised
NULL	BYTE_POINTER	buffer not excised
Calls		
Function	**Where Described**	
buffer_from_char	Sec. 2.14.4.3	
queue_excise	Sec. 2.14.4.2.7	
char_from_buffer	Sec. 2.14.4.3	

Table 2.14-151: buffer_exercise CSU [8.14.4.2.19]

2.14.4.2.20 buffer_dequeue CSU

This CSU dequeues the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to QUEUE	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
queue_dequeue	Sec. 2.14.4.2.6	
char_from_buffer	Sec. 2.14.4.3	

Table 2.14-152: buffer_dequeue CSU [8.14.4.2.20]

2.14.4.2.21 buffer_traverse_and_apply CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to register QUEUE	Sec. 2.14.4.3
fcn	PFI	sim_types.h
arg1	not defined	N/A
arg2	not defined	N/A
Calls		
Function	**Where Described**	
queue_dequeue	Sec. 2.14.4.2.6	
char_from_buffer	Sec. 2.14.4.3	

Table 2.14-153: buffer_traverse_and_apply CSU [8.14.4.2.21]

2.14.4.2.22 buffer_traverse_and_apply_n_times CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to register QUEUE	Sec. 2.14.4.3
fcn	PFI	sim_types.h
arg	not defined	N/A
n	int	Standard
Calls		
Function	**Where Described**	
QLOCK	Sec. 2.14.4.3	
QUNLOCK	Sec. 2.14.4.3	
char_from_buffer	Sec. 2.14.4.3	

Table 2.14-154: buffer_traverse_and_apply_n_times CSU [8.14.4.2.22]

2.14.4.2.23 buffer_flush CSU

This CSU flushes the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to QUEUE	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
queue_flush	Sec. 2.14.4.2.10	

Table 2.14-155: buffer_flush CSU [8.14.4.2.23]

2.14.4.2.24 buffer_describe CSU

This CSU prints information about the buffer.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
p	BYTE_POINTER	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
buffer_from_char	Sec. 2.14.4.3	

Table 2.14-156: buffer_describe CSU [8.14.4.2.24]

2.14.4.2.25 queue_distribute CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to QUEUE	Sec. 2.14.4.3
*bp	pointer to BUFFER	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
queue_traverse_and_apply	Sec. 2.14.4.2.8	

Table 2.14-157: queue_distribute CSU [8.14.4.2.25]

2.14.4.2.26 buffer_distribute CSU

Parameters		
Parameters	**Type**	**Where Typedef Declared**
*qp	pointer to QUEUE	Sec. 2.14.4.3
p	BYTE_POINTER	Sec. 2.14.4.3
Calls		
Function	**Where Described**	
buffer_traverse_and_apply	Sec. 2.14.4.2.8	

Table 2.14-158: buffer_distribute CSU [8.14.4.2.26]

2.14.4.2.27 buffer_statistics_init CSU

Calls	
Function	**Where Described**
heap_allocate	Sec. 2.14.2.1.1

Table 2.14-159: buffer_statistics_init CSU [8.14.4.2.27]

2.14.4.2.28 buffer_one_more CSU

Calls	
Function	**Where Described**
buffer_statistics_init	Sec. 2.14.4.2.27

Table 2.14-160: buffer_one_more CSU [8.14.4.2.28]

2.14.4.2.29 buffer_one_less CSU

Calls	
Function	**Where Described**
buffer_statistics_init	Sec. 2.14.4.2.27

Table 2.14-161: buffer_one_less CSU [8.14.4.2.29]

2.14.4.2.30 buffer_statistics_print CSU

This CSU prints information about the buffer.

Calls	
Function	**Where Described**
buffer_statistics_init	Sec. 2.14.4.2.27

Table 2.14-162: buffer_statistics_print CSU [8.14.4.2.30]

2.14.4.3 libcomm.h CSU

/simnet/libsrc/libcomm/libcomm.h

This CSU contains constant definitions, macro definitions, and type definitions related to queues and functions. The macros are described in Appendix A.

The constant and structure definitions are contained in the following tables.

Constant	Value
BUFFER_LENGTH	1024

Table 2.14-163: BUFFER_LENGTH Constant Definition

The following typedef struct is a fixed-length queue whose length depends on whether MULTIPROCESSOR is defined at compilation time. If it is, the lock item is added, as shown in the following table. The SIMPLE_QUEUE structure is used to build buffer pools; BUFFER_POOL is type defined as SIMPLE_QUEUE.

Item	Type	Where Type Defined
#ifdef MULTIPROCESSOR		
lock	LOCK_TYPE	
#endif		
queue_length_mask	unsigned long	Standard
dequeue_this	unsigned long	Standard
enqueue_here	unsigned long	Standard
*queue	pointer to POINTER	sim_style.h

Table 2.14-164: SIMPLE_QUEUE Structure Definition

Item	Type	Where Type Defined
*free_pool	pointer to BUFFER_POOL	The previous typedef struct
refcnt	int	Standard
length	int	Standard

Table 2.14-165: BUFFER Structure Definition

The following typedef struct is tagged qelem.

Item	Type	Where Type Defined
*prev	pointer to qelem	This typedef struct
*next	pointer to qelem	This typedef struct
*data	pointer to BUFFER	The previous typedef struct

Table 2.14-166: QELEMENT Structure Definition

The following typedef struct is tagged queue. The structure's length depends on whether MULTIPROCESSOR is defined at compilation time. If it is, the lock item is added, as shown in the following table.

Item	Type	Where Type Defined
#ifdef MULTIPROCESSOR		
lock	LOCK_TYPE	
#endif		
*first	pointer to QELEMENT	The previous typedef struct
*last	pointer to QELEMENT	The previous typedef struct
length	int	Standard

Table 2.14-167: QUEUE Structure Definition

2.15 SUPPORT CSC

This CSC contains the CSUs used to test the SAF Simhost CSCI.

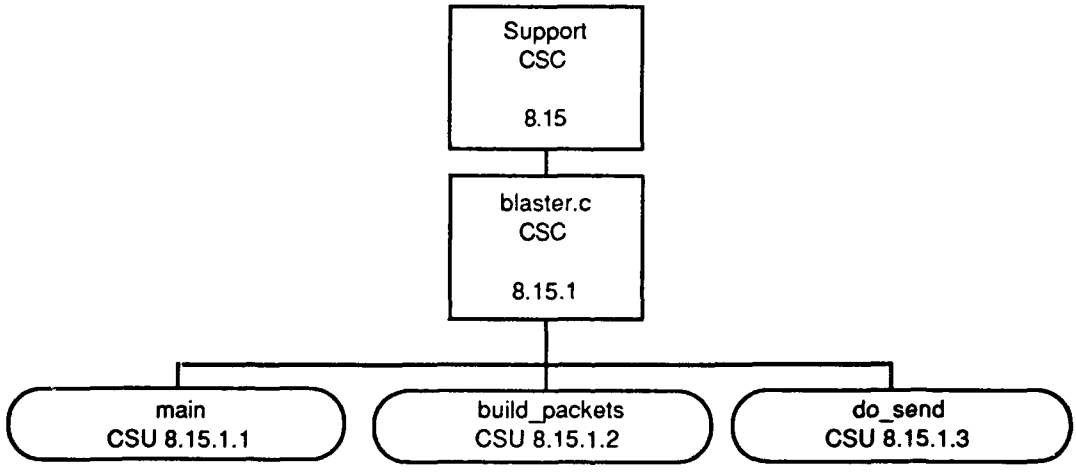


Figure 2.15-1: Support CSC Structure

2.15.1 blaster.c CSC

/siment/src/host/blaster.c

This CSC generates a program which creates multiple vehicles in locations scattered around the database. It is used to test load handling on other simhosts.

2.15.1.1 main CSU

This CSU is the base function for generating vehicles to test load handling.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
argc	int	Standard
*argv[ ]	pointer to char	Standard
Internal Variables		
Parameters	**Type**	**Where Typedef Declared**
net_handle	int	Standard
errbuf[80]	char	Standard
Calls		
Function	**Where Described**	
net_open	Sec. 2.20.2.10.1 MCC CSCI SDD	
tdb_init_cache	Sec. 2.21.7.27.1 MCC CSCI SDD	
tdb_init_patch_guards	Sec. 2.21.7.10.4 MCC CSCI SDD	
build_packets	Sec. 2.15.1.2	
do_send	Sec. 2.15.1.3	
net_close	Sec. 2.20.2.3.1 MCC CSCI SDD	

Table 2.15-1: main CSU [8.15.1.1]

2.15.1.2 build_packets CSU

This CSU generates an Association PDU, a Simulation PDU and a Vehicle Appearance Variant for a list of vehicles.

Calls	
Function	**Where Described**
PRO_SIM_APPEARANCE_SIZE	p_size.h
tdb_place_vehicle	libtdb
rmat_to_fmat	Sec. 2.14.3.5.15

Table 2.15-2: build_packet CSU [8.15.1.2]

2.15.1.3 do_send CSU

This CSU sends the Association and Simulation PDUs for a list of vehicles.

Parameters		
Parameters	**Type**	**Where Typedef Declared**
net_handler	int	Standard

Calls	
Function	**Where Described**
get_millisecond_time	Sec. 2.14.3.3.2
net_send	Sec. 2.20.2.15.1 MCC CSCI SDD
PRO_ASSOC_DATAGRAM_SIZE	p_size.h

Table 2.15-3: do_send CSU [8.15.1.3]

APPENDIX A

This appendix contains a list of defines, in alphabetical order (upper followed by lower case characters), along with their values and locations. Note that defines that equate directly to fixed values are not included.

Define	ADD_STRING(s,p,b)
Expansion	{ if (p>b) {sprintf(p,", "); p+=2;} sprintf(p,s); p+=strlen(p) }
Location	/simnet/src/host/saf_vehicle.c Sec. 2.6.1.1.10

Define	ALLOCATOR /* conditional on ALLOCATOR not being defined */
Expansion	malloc
Location	/simnet/libsrc/libreader/libreader.h Sec. 2.1.1.5

Define	BEGIN_ARG_TABLE(name)
Expansion	int name[ ][3] = {
Location	/simnet/libsrc/libcomm/libcomm.h Sec. 2.14.4.3

Define	C1(t0)
Expansion	f(a[)].arg.t0)
Location	/simnet/libsrc/libsched/invoke.c Sec. 2.2.1.2

Define	C2(t0, t1)
Expansion	f(a[0].arg.t0,a[1].arg.t1)
Location	/simnet/libsrc/libsched/invoke.c Sec. 2.2.1.2

Define	C3(t0, t1, t2)
Expansion	f(a[0].arg.t0,a[1].arg.t1,a[2].arg.t2)
Location	/simnet/libsrc/libsched/invoke.c Sec. 2.2.1.2

Define	C4(t0, t1, t2, t3)
Expansion	f(a[0].arg.t0,a[1].arg.t1,a[2].arg.t2,a[3].arg.t3)
Location	/simnet/libsrc/libsched/invoke.c Sec. 2.2.1.2

Define	CALL(f)
Expansion	{P_CALL, ((char *)f), NULL}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	CALL1(f,a)
Expansion	{P_CALL, ((char *)f), ((char *)a)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	CDR(list)
Expansion	((list == NULL_PTR) ? NULL_PTR : (list->cdr))
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	CDDR(list)
Expansion	(((list == NULL_PTR) || (list->cdr == NULL_PTR)) ? NULL_PTR : ((list-cdr)->cdr))
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	CITY_BLOCKDISTANCE2(a,b)
Expansion	(abs(*(a) - *(b)0 + abs((a)[1] - (b)[1]))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	CITY_BLOCKDISTANCE3(a,b)
Expansion	(abs(*(a) - *(b)0 + abs((a)[1] - (b)[1]) + abs((a) - (b)[2]))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	COMMAND_DONE
Expansion	{P_PARSE_FUNCTION, (char *)ParseCommandDone, NULL}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	CONSTANT(a, b, c)
Expansion	{((int)a), ((char *)b), ((char *)c)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	CONSTANT_TABLE(t)
Expansion	PARSE_TABLE t[ ] = {
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	CONVERT_FROM_SIXTEENTH_TIME(time)
Expansion	(time) >> 4
Location	/simnet/libsrc/libutil/time.c Sec. 2.14.3.3

Define	DEALLOCATE /* conditional on ALLOCATOR not being defined */
Expansion	free
Location	/simnet/libsrc/libreader/libreader.h Sec. 2.1.1.5

Define	DEBUG(args)
Expansion	{if (ge_debug_flags & D_GLOBAL) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_AIR(args)
Expansion	{if (ge_debug_flags & D_AIR) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_CM(args)
Expansion	{if (ge_debug_flags & D_CM) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_COLLISION(args)
Expansion	{if (ge_debug_flags & D_COLLISION) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_COMPOSITE(args)
Expansion	{if (ge_debug_flags & D_COMPOSITE) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_CONNECTION(x)
Expansion	{ if (g_debug_flags & (g_debug_flags & D_CONNECTION) printf x; }
Location	/simnet/libsrc/libudp/udp_conn.h Sec. 2.4.2.4

Define	DEBUG_DRIVER(args)
Expansion	{if (ge_debug_flags & D_DRIVER) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_EVENT(args)
Expansion	{if (ge_debug_flags & D_EVENT) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_GROUND(args)
Expansion	{if (ge_debug_flags & D_GROUND) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_GUNNER(args)
Expansion	{if (ge_debug_flags & D_GUNNER) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_INTERVIS(args)
Expansion	{if (ge_debug_flags & D_INTERVIS) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_LOADER(args)
Expansion	{if (ge_debug_flags & D_LOADER) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_MISSILE(args)
Expansion	{if (ge_debug_flags & D_MISSILE) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_NAVIGATOR(args)
Expansion	{if (ge_debug_flags & D_NAVIGATOR) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_OBSTACLE(args)
Expansion	{if (ge_debug_flags & D_OBSTACLE) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_PILOT(args)
Expansion	{if (ge_debug_flags & D_PILOT) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_REMOTE(args)
Expansion	{if (ge_debug_flags & D_REMOTE) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_REPORTER(args)
Expansion	{if (ge_debug_flags & D_REPORTER) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_STATION(args)
Expansion	{if (ge_debug_flags & D_STATION) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_TACTICAL(args)
Expansion	{if (ge_debug_flags & D_TACTICAL) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_TARGETING(args)
Expansion	{if (ge_debug_flags & D_TARGETING) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_TURRET(args)
Expansion	{if (ge_debug_flags & D_TURRET) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_VEHICLE(args)
Expansion	{if (ge_debug_flags & D_VEHICLE) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DEBUG_WEAPON(args)
Expansion	{if (ge_debug_flags & D_WEAPON) printf args;}
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	DECLARE_FOR_VEHICLES_DO_VARS
Expansion	short fvwngd_i, fvwngd_j; int fvwngd_x; int fvwngd_y; int fvwngd_xg, fvwngd_yg, fvwngd_w, fvwngd_h; short low_fvwngd_xoff, high_fvwngd_xoff; short low_fvwngd_yoff, high_fvwngd_yoff; GRID_ENTRY **fvwngd_gt; GRID_ENTRY *fvwngd_ge; short fvwngd_n; struct saf_object *s
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	DEFINE_TABLE(t)
Expansion	PARSE_TABLE t[ ] = {
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	DIST(A, B, C, D) /* Signed distance from (A,B) to (C,D) */
Expansion	(use_x ? (x_pos ? (C) - (A) : (A) - (C)) : (y_pos ? (D) - (B) : (B) - (D)))
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	DIV_GLOC(x) /* coarsens coord. to an int in {0, 1, ... , PV_GRID_NUM} */
Expansion	(((int)(x + EPSI)) / PV_GRID_SIDE)
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	DO_KEYWORD_TABLE(t)
Expansion	{P_PARSE_FUNCTION, (char *)ParseDoKeywordTable, ((char *)t)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	DT
Expansion	safobj->tickable->time_since_last_tick
Location	/simnet/src/host/flyingveh.h Sec. 2.6.4.7

Define	END_ARG_TABLE
Expansion	{0,0,0}};
Location	/simnet/libsrc/libcomm/libcomm.h Sec. 2.14.4.3

Define	END_CONSTANT_TABLE
Expansion	{0, NULL, NULL} }
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	END_DEFINE_TABLE
Expansion	{0, NULL, NULL} }
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	END_FIELD_TABLE
Expansion	{0, NULL, NULL} }
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	END_KEYWORD
Expansion	{P_END_KEYWORD, NULL, NULL}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	END_KEYWORD_SELECT
Expansion	{P_END_KEYWORD_SELECT, NULL, NULL}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	ENERGY(mass,speed)
Expansion	(0.5 * mass * speed * speed)
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	ERROR_ABBORT
Expansion	if (g_abort_on_error) abort()
Location	/simnet/src/host/flyingveh.h Sec. 2.6.4.7

Define	ERROR_OUT(args)
Expansion	printf args
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	EVEN_P(x)
Expansion	(!(x & 1))
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	FARTHER2(a,b,k)
Expansion	(((*(a) - *(b))*(*(a) - *(b)) + ((a)[1] - (b)[1])*((a)[1] - (b)[1])) > (k)*(k))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	FARTHER3(a,b,k)
Expansion	(((*(a) - *(b))*(*(a) - *(b)) + ((a)[1] - (b)[1])*((a)[1] - (b)[1]) + ((a)[2] - (b)[2])*((a)[2] - (b)[2])) > (k)*(k))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	FIELD(a, b, c)
Expansion	{((int)a), ((char *)b), ((char *)c)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	FIELD_TABLE(t)
Expansion	PARSE_TABLE t[ ] = {
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	FIRST(list)
Expansion	((list == NULL_PTR) ? NULL_PTR : (list->car))
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	FOR_VEHICLES_DO(match_type, form)

Expansion	{
	unsigned short id;
	struct saf_object **sp;
	struct saf_object *s;
	
	for (id=0,sp=g_safobj_table;id<MAX_VEH;id++,sp++)
	{
	if (*sp && type_ok((*sp)->saf_type,match_type))
	{
	s = *sp;
	form;
	}
	}
	}

Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	FOR_VEHICLES_WITHIN_N_GRIDS_DO(n,posx,posy,table,match_type,form)
Expansion	{ fvwngd_x =posx; fvwngd_y=posy; fvwngd_xn=n; if (coords_within_database(fvwngd_x , fvwngd_y)) { fvwngd_gt = grid_tables[tables]; fvwngd_xg = fvwngd_x/grid_table_size[table]; fvwngd_yg = fvwngd_y/grid_table_size[table]; fvwngd_w = grid_table_widths[table]; fvwngd_h = grid_table_heights[table]; low_fvwngd_xoff = max(-fvwngd_xg, -fvwngd_n); high_fvwegd_xoff = (fvwngd_w-fvwngd_xg,fvwngd_n); low_fvwngd_yoff = max(-fvwngd_yg, -fvwngd_n); high_fvwegd_yoff = min(fvwngd_h-fvwngd_yg,fvwngd_n); for (fvwngd_i=low_fvwngd_xoff; fvwngd_i<=high_fvwngd_xoff;fvwngd_i++) { for (fvwngd_j=low_fvwngd_yoff; fvwngd_j<=high+fvwngd_yoff;fvwngd_j++) { for (fvwngd_ge=fvwngd_gt[(fvwngd_xg+fvwngd_i)+ fvwngd_w*(fvwngd_yg+fvwngd_j)]; fvwngd_ge;fvwngd_ge=fvwngd_ge->next) { s = fvngd_ge->vehicle; if (s && type_ok(s->saf_type,match_type)) { form; } } } } } }
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	FORCE_OUT(args)
Expansion	{ printf args; fflush(stdout); }
Location	/simnet/src/host/debug.h Sec. 2.5.2.2

Define	GETCONSTANT(f)
Expansion	{P_PARSE_FUNCTION, (char *)ParseGetConstant, ((char *)f)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	GETDECIMAL(a)
Expansion	{P_PARSE_FUNCTION, (char *)ParseGetdecimal, ((char *)a)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	GETFIELDS(f)
Expansion	{P_PARSE_FUNCTION, (char *)ParseGrtFields, ((char *)f)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	GETHEX(a)
Expansion	{P_PARSE_FUNCTION, (char *)ParseGetHex, ((char *)a)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	GETOCTAL(a)
Expansion	{P_PARSE_FUNCTION, (char *)ParseGetOctal, ((char *)a)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	GETSTRING(a)
Expansion	{P_PARSE_FUNCTION, (char *)ParseGetString, ((char *)a)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	GET_ON_OR_OFF
Expansion	{P_PARSE_FUNCTION, (char *)ParseDoKeywordTable, ((char *)g_ParseOnOff)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	GET_YES_OR_NO
Expansion	{P_PARSE_FUNCTION, (char *)ParseDoKeywordTable, ((char *)g_ParseYesNo)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	I32MAX(x, y)
Expansion	(INT32) ((x > y) ? x : y)
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	I32MIN(x, y)
Expansion	(INT32) ((x < y) ? x : y)
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	INSIDE_BOX2(point,tolerance)
Expansion	((abs(point[X]) <= tolerance) && (abs(point[Y]) <= tolerance))
Location	/simnet/src/host/flyingveh.h Sec. 2.6.4.7

Define	IS_ARTY(x)
Expansion	(((x & (objectDomainMask | vehicleEnvironmentMask | vehicleFunctionMask)) == (objectDomainVehicle | vehicleEnvironmentGround | vehicleFunctionHowitzer)) | | ((x & (objectDomainMask | vehicleEnvironmentMask | vehicleFunctionMask)) == (objectDomainVehicle | vehicleEnvironmentGround | vehicleFunctionMortar)))
Location	/simnet/src/host/reporter.c Sec. 2.8.2.1

Define	IS_PC(x)
Expansion	((x & (objectDomainMask | vehicleEnvironmentMask | vehicleFunctionMask)) == (objectDomainVehicle | vehicleEnvironmentGround | vehicleFunctionPersonnelCarrier))
Location	/simnet/src/host/reporter.c Sec. 2.8.2.1

Define	IS_RESUPPLYING(a)
Expansion	((a)->logistics->supplier_id)
Location	/simnet/src/host/logistics.h Sec. 2.6.6.2

Define	IS_TRUCK(x)
Expansion	((x & (objectDomainMask | vehicleEnvironmentMask | vehicleFunctionMask)) == (objectDomainVehicle | vehicleEnvironmentGround | vehicleFunctionSupplyTruck))
Location	/simnet/src/host/reporter.c Sec. 2.8.2.1

Define	KEYWORD(f, c)
Expansion	{P_KEYWORD, ((char *)f), ((char *)c)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	KEYWORD_SELECT(a)
Expansion	{P_KEYWORD_SELECT, NULL, ((char *)a)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	LOOKUP_APPEARANCE(id)
Expansion	(OBJ_APPEARANCE(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_DIRECTION(id)
Expansion	(OBJ_DIRECTION(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_FORCEID(id)
Expansion	(OBJ_FORCEID(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_GUISES(id)
Expansion	(OBJ_GUISES(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_GUNELEVATION(id)
Expansion	(OBJ_GUNELEVATIONLOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_HULL_TO_WORLD(id)
Expansion	(OBJ_HULL_TO_WORLD(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_OBJECT_TYPE(id)
Expansion	(OBJ_OBJECT_TYPE(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_OWNER_PORT_NUMBER(id)
Expansion	(OBJ_OWNER_PORT_NUMBER(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_POSITION(id)
Expansion	(OBJ_POSITON(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_SAFOBJ(id)
Expansion	(g_safobj_table[(id)])
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_SIMPLE_RCVQ(id)
Expansion	(OBJ_SIMPLE_RCVQ(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_TIME_SINCE_LAST_TICK(id)
Expansion	(OBJ_TIME_SINCE_LAST_TICK(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_TURRETAZIMUTH(id)
Expansion	(OBJ_TURRETAZIMUTH(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_VEHICLE(id,type)
Expansion	((g_safobj_table[(id)] ? (type_ok(g_safobj_table[(id)]->saf_type, (type)) ? (g_safobj_table[(id)] : 0) : 0)
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	LOOKUP_VEHICLECLASS(id)
Expansion	(OBJ_VEHICLECLASS(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_VEHICLESTATUS(id)
Expansion	(OBJ_VEHICLESTATUS(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	LOOKUP_VELOCITY(id)
Expansion	(OBJ_VELOCITY(LOOKUP_SAFOBJ(id)))
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	MAX(x, y)
Expansion	(x > y) ? x : y
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	MIN(x, y)
Expansion	(x < y) ? x : y
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	MOD_QUEUE_SIZE(sqp,n)
Expansion	((n) & (sqp)->queue_length_mask)
Location	/simnet/libsrc/libcomm/bufpool.c Sec. 2.14.4.1

Define	MOMENTUM(mass,speed)
Expansion	(mass*speed)
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	NEXT_BLOCK(block)
Expansion	((BLOCK_HEADER*)(((char*)block) + block->size + sizeof)BLOCK_HEADER)))
Location	/simnet/libsrc/libheap/libheap.h Sec. 2.14.2.3

Define	OBJ_APPEARANCE(ptr)
Expansion	((ptr)->entity->appearance)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_DIRECTION(ptr)
Expansion	((ptr)->entity->direction)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_FORCEID(ptr)
Expansion	((ptr)->entity->forceID)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_GUISES(ptr)
Expansion	((ptr)->vehicle->guises)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_GUNELEVATION(ptr)
Expansion	((ptr)->vehicle->gunElevation)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_HULL_TO_WORLD(ptr)
Expansion	((ptr)->vehicle->hull_to_world)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_OBJECT_TYPE(ptr)
Expansion	((ptr)->vehicle->object_type)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_OWNER_PORT_NUMBER(ptr)
Expansion	((ptr)->entity->owner_port_number)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_POSITION(ptr)
Expansion	((ptr)->entity->position)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_SIMPLE_RCVQ(ptr)
Expansion	((ptr)->entity->simple_rcvq)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_TIME_SINCE_LAST_TICK(ptr)
Expansion	((ptr)->tickable->time_since_last_tick)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_TURRETAZIMUTH(ptr)
Expansion	((ptr)->vehicle->turretAzimuth)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_VEHICLECLASS(ptr)
Expansion	((ptr)->vehicle->vehicleClass)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_VEHICLEID(ptr)
Expansion	((ptr)->entity->vehicleID)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_VEHICLESTATUS(ptr)
Expansion	((ptr)->vehicle->vehicleStatus)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	OBJ_VELOCITY(ptr)
Expansion	((ptr)->entity->velocity)
Location	/simnet/src/host/safobj.h Sec. 2.9.1.1

Define	ODD_P(x)
Expansion	(x & 1)
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	OPTIONAL
Expansion	{P_PARSE_FUNCTION, (char *)ParseOptional, NULL}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	ORDER_FROM_USER
Expansion	((SAF_OBJECT *)NULL)
Location	/simnet//src/host/sbx.c Sec. 2.4.3.2

Define	PARSE_FUNCTION(f, a)
Expansion	{P_PARSE_FUNCTION, ((char *)f), ((char *)a)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	PARSE_KEYWORDS(t)
Expansion	{P_PARSE_KEYWORDS, ((char *)t), NULL}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	PATCH_COORD(x) /* Shift coordinates with (0,0) at lower left corner of patch */
Expansion	((x) - PATCH_ROUND_DOWN)
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	PATCH_ROUND_DOWN(x)
Expansion	(patch_size * (((int)(x)) / patch_size))
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	PATCH_ROUND_UP(x)
Expansion	(patch_size * (1 + (((int)(x)) /patch_size)))
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	POKE(d, a)
Expansion	PUTARG(sizeof (*a)) PUTARG(d) PUTARG(a) CALL(ParPoke)
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	POKE_B(d, a)
Expansion	PUTARG(1) PUTARG(d) PUTARG(a) CALL(ParPoke)
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	POKE_L(d, a)
Expansion	PUTARG(4) PUTARG(d) PUTARG(a) CALL(ParPoke)
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	POKE_W(d, a)
Expansion	PUTARG(2) PUTARG(d) PUTARG(a) CALL(ParPoke)
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	POINT_TO_LINE(x, y, m, b) /* gives "y" distance from point (x,y) to line */
Expansion	((y) - (m)*(x) - (b))
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	POSITION
Expansion	safobj->entity->position
Location	/simnet/src/host/flyingveh.h Sec. 2.6.4.7

Define	PRINT(s)
Expansion	{P_PRINT, ((char *)s), NULL}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	PRINT1(s, a)
Expansion	{P_PRINT, ((char *)s), ((char *)a)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	PRINTFIELDS(f, b)
Expansion	P_PRINTFIELD, ((char *)f), ((char *)b)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	PUTARG(v)
Expansion	P_PARSE_FUNCTION, (char *)ParsePutArg, ((char *)v)}
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	PV_GLOC_BIT(x,y)
Expansion	(DIV_GLOC(x) + DIV_GLOC(y)) * PV_GRID_NUM)
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	PV_GLOC_BIT2(x,y)
Expansion	((x) + (y) * PV_GRID_NUM)
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	PV_REF_GR_NUM /* no. of refined grid squares per side */
Expansion	(PV_GRID_NUM * REFINEMENT_LEVEL)
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	PV_REF_GR_SIDE /* side length of one refined grid square */
Expansion	((float)patch_size / PV_REF_GR_NUM)
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	QLOCK(sqp) /* only if MULTIPROCESSOR defined, otherwise ";" */
Expansion	lock_wait(&(qp->lock))
Location	/simnet/libsrc/libcomm/queue_fix.c Sec. 2.14.4.2

Define	QUNLOCK(sqp) /* only if MULTIPROCESSOR defined, otherwise ";" */
Expansion	lock_unlock(&(qp->lock))
Location	/simnet/libsrc/libcomm/queue_fix.c Sec. 2.14.4.2

Define	RANGE_CLIP(a,x,b)
Expansion	(((x) < (a)) ? (a) : (((x) > (b)) ? (b) : (x)))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	RETYPE_VEHICLE(id,type)
Expansion	(g_safobj_table[(id)]->saf_type = (type))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	S(n, x, y)

Expansion	switch (a[0].tag)
	{
	case A_INT: x ; break ;
	case A_DOUBLE: y ; break ;
	}
	break

Location	/simnet/libsrc/libsched/invoke.c Sec. 2.2.1.2

Define	SAF_COMPOSITE_P(safobj)
Expansion	(type_ok(safobj->saf_type, COMPOSITE))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	SAF_INACTIVE_REMOTE_VEHICLE_P(safobj)
Expansion	(type_ok(safobj->saf_type, INACTIVE_REMOTE_VEHICLE))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	SAF_LOCAL_P(safobj)
Expansion	(type_ok(safobj->saf_type, LOCALS))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	SAF_LOCAL_VEHICLE_P(safobj)
Expansion	(type_ok(safobj->saf_type, LOCAL_VEHICLE))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	SAF_MISSILE_P(safobj)
Expansion	(type_ok(safobj->saf_type, MOSSILE))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	SAF_REMOTE_VEHICLE_P(safobj)
Expansion	(type_ok(safobj->saf_type, REMOTE_VEHICLE))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	SAF_VEHICLE_P(safobj)
Expansion	(type_ok(safobj->saf_type, VEHICLE))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	SCALE_COORDS(x)
Expansion	((int)((x) / pv_ref_gr_size))
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	SECOND(list)
Expansion	(((list == NULL_PTR) || (list->cdr == NULL_PTR)) ? NULL_PTR : ((list->cdr)->car))
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	SQDQ(sqp)
Expansion	MOD_QUEUE_SIZE(sqp,(sqp)->dequeue_this)
Location	/simnet/libsrc/libcomm/bufpool.c Sec. 2.14.4.1

Define	SQEMPTY(sqp)
Expansion	(SQLENGTH(sqp) IS 0)
Location	/simnet/libsrc/libcomm/bufpool.c Sec. 2.14.4.1

Define	SQFULL(sqp)
Expansion	(SQLENGTH(sqp) IS (sqp->queue_length_mask +1)
Location	/simnet/libsrc/libcomm/bufpool.c Sec. 2.14.4.1

Define	SQLENGTH(sqp)
Expansion	(sqp->enqueue_here - sqp->deque_this)
Location	/simnet/libsrc/libcomm/bufpool.c Sec. 2.14.4.1

Define	SQLOCK(sqp) /* only if MULTIPROCESSOR defined, otherwise ";" */
Expansion	lock_wait(&(sqp->lock))
Location	/simnet/libsrc/libcomm/bufpool.c Sec. 2.14.4.1

Define	SQNQ(sqp)
Expansion	MOD_QUEUE_SIZE(sqp,(sqp)->enqueue_here)
Location	/simnet/libsrc/libcomm/bufpool.c Sec. 2.14.4.1

Define	SQUNLOCK(sqp) /* only if MULTIPROCESSOR defined, otherwise ";" */
Expansion	lock_unlock(&(sqp->lock))
Location	/simnet/libsrc/libcomm/bufpool.c Sec. 2.14.4.1

Define	STILL_IN_BOUNDS_X(x)
Expansion	(x_pos ? (x) < x_max_tdb : (x) >= 0)
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	STILL_IN_BOUNDS_Y(y)
Expansion	(y_pos ? (y) < y_max_tdb : (y) >= 0)
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	SWITCH(c, argcount, func)
Expansion	{(int) 'c', argcount, (int) func}.
Location	/simnet/libsrc/libcomm/libcomm.h Sec. 2.14.4.3

Define	TABLE_APPLIES_TO_FUNCTION(table,safobj)
Expansion	(((table) == TABLE_200) || (OBJ_FORCEID((safobj)) == targetForceID) || (((table) == TABLE_DIST_3500) && (OBJ_FORCEID((safobj)) == distinguishedForceID)) || (((table) == TABLE_OTHER_3500) && (OBJ_FORCEID((safobj)) == otherForceID)))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	TABLE_SIZE_FUNCTION(table)
Expansion	((table) ? 3500 : 200)
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

Define	VEC_ADD_2D(a, b, c)
Expansion	(c)->x = (a)->x + (b)->x; (c)->y = (a)->y + (b)->y
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	VEC_ANGLE_2D(a)
Expansion	(((a)->x == 0.0 && ((a)->y == 0.0)) ? 0.0 : atan2((a)->y, (a)->x))
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	VEC_LONGER2(a,b)
Expansion	(((*(a))*(*(a)) + ((a)[1])*((a)[1])) > ((*(b))*(*(b)) + ((b)[1])*((b)[1]))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	VEC_LONGER3(a,b)
Expansion	(((*(a))*(*(a)) + ((a)[1])*((a)[1]) + ((a)[2])*((a)[2])) > ((*(b))*(*(b)) + ((b)[1])*((b)[1]) + ((b)[2])*(b)[2]))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	VEC_SCALE_2D(a, b, c)
Expansion	(c)->x = (a)->x * b; (c)->y = (a)->y * b
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	VEC_SMALLER2(a,k)
Expansion	(((*(a))*(*(a)) + ((a)[1])*((a)[1])) < (k)*(k))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	VEC_SMALLER3(a,k)
Expansion	(((*(a))*(*(a)) + ((a)[1])*((a)[1]) + ((a)[2])*(a)[2])) < (k)*(k))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	VEC_SUB_2D(a, b, c)
Expansion	(c)->x = (a)->x - (b)->x; (c)->y=(a)->y - (b)->y
Location	/simnet/libsrc/libquad/libquad.h Sec. 2.12.1.18

Define	VEHICLEID
Expansion	safobj->entity->vehivleID
Location	/simnet/src/host/flyingveh.h Sec. 2.6.4.7

Define	XC(D)
Expansion	(use_x ? x_from + (x_pos ? (D) : -(D)) : x_from + (y_pos ? (D) : -(D)) * x_slope)
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	YC(D)
Expansion	(use_x ? y_from + (x_pos ? (D) : -(D)) * y_slope : y_from + (y_pos ? (D) : -(D)))
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	abs(x) /* return the absolute value of x */
Expansion	(((x) >= 0) ? (x) : (-(x)))
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	abs(x) /* return the absolute value of x */
Expansion	(((x) >= 0) ? (x) : (-(x)))
Location	/simnet/include/saf/src/abs.h Sec. 2.13.3.2

Define	allocate_airveh()
Expansion	(AIRVEH_VARS *)heap_allocate(sizeof(AIRVEH_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_area_cm()
Expansion	(AREA_CM *)heap_allocate(sizeof(AREA_CM))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	allocate_cis()
Expansion	(CIS *)heap_allocate(sizeof(CIS))
Location	/simnet/src/host/cis.h Sec. 2.10.1.2

Define	allocate_collision()
Expansion	(COLLISION_VARS *)heap_allocate(sizeof(COLLISION_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_commander()
Expansion	(COMMANDER_VARS *)heap_allocate(sizeof(COMMANDER_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_composite()
Expansion	(COMPOSITE_VARS *)heap_allocate(sizeof(COMPOSITE_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_control_measure()
Expansion	(CONTROL_MEASURE *)heap_allocate(sizeof(CONTROL_MEASURE))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	allocate_control_measure_list_item()
Expansion	(CONTROL_MEASURE_LIST *) heap_allocate(sizeof(CONTROL_MEASURE_LIST))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	allocate_damage()
Expansion	(DAMAGE_VARS *)heap_allocate(sizeof(DAMAGE_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_detection()
Expansion	(DETECTION_VARS *)heap_allocate(sizeof(DETECTION_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_driver()
Expansion	(DRIVER_VARS *)heap_allocate(sizeof(DRIVER_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_entity()
Expansion	(ENTITY_VARS *)heap_allocate(sizeof(ENTITY_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_groundveh()
Expansion	(GROUNDVEH_VARS *)heap_allocate(sizeof(GROUNDVEH_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_line_cm()
Expansion	(LINE_CM *)heap_allocate(sizeof(LINE_CM))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	allocate_logistics()
Expansion	(LOGISTICS_VARS *)heap_allocate(sizeof(LOGISTICS_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_missile()
Expansion	(MISSILE_VARS *)heap_allocate(sizeof(MISSILE_VARS))
Location	/simnet/src/host/missile.h Sec. 2.6.9.2

Define	allocate_navigator()
Expansion	(NAVIGATOR_VARS *)heap_allocate(sizeof(NAVIGATOR_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_overlay()
Expansion	(OVERLAY *)heap_allocate(sizeof(OVERLAY))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	allocate_overlay_list_item()
Expansion	(OVERLAY_LIST *)heap_allocate(sizeof(OVERLAY_LIST))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	allocate_pilot()
Expansion	(PILOT_VARS *)heap_allocate(sizeof(PILOT_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_point_cm()
Expansion	(POINT_CM *)heap_allocate(sizeof(POINT_CM))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	allocate_predicates()
Expansion	(PREDICATES *)heap_allocate(sizeof(PREDICATES))
Location	/simnet/src/host/predicates.h Sec. 2.10.1.4

Define	allocate_remote()
Expansion	(REMOTE_VARS *)heap_allocate(sizeof(REMOTE_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_rem_veh_id_list()
Expansion	(REM_VEH_ID_LIST *)heap_allocate(sizeof(REM_VEH_ID_LIST))
Location	/simnet/src/host/reporter.h Sec. 2.8.2.2

Define	allocate_reporter()
Expansion	(REPORTER_VARS *)heap_allocate(sizeof(REPORTER_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_route()
Expansion	(ROUTE *)heap_allocate(sizeof(ROUTE))
Location	/simnet/src/host/route.h Sec. 2.10.2.5

Define	allocate_routepoint()
Expansion	(ROUTEPOINT *)heap_allocate(sizeof(ROUTEPOINT))
Location	/simnet/src/host/route.h Sec. 2.10.2.5

Define	allocate_saf_vehicle()
Expansion	(SAF_VEHICLE_VARS *)heap_allocate(sizeof(SAF_VEHICLE_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_sbx_connection()
Expansion	(SBX_CONNECTION_VARS *) heap_allocate(sizeof(SBX_CONNECTION_VARS))
Location	/simnet/src/host/sbx.h Sec. 2.4.3.3

Define	allocate_shell_summary()
Expansion	(SHELL_SUMMARY *)heap_allocate(sizeof(SHELL_SUMMARY))
Location	/simnet/src/host/reporter.h Sec. 2.8.2.2

Define	allocate_spotter()
Expansion	(SPOTTER_VARS *)heap_allocate(sizeof(SPOTTER_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_targeting()
Expansion	(TARGETING_VARS *)heap_allocate(sizeof(TARGETING_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_tickable()
Expansion	(TICKABLE_VARS *)heap_allocate(sizeof(TICKABLE_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_turret()
Expansion	(TURRET_VARS *)heap_allocate(sizeof(TURRET_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_veh_cluster()
Expansion	(VEH_CLUSTER *)heap_allocate(sizeof(VEH_CLUSTER))
Location	/simnet/src/host/reporter.h Sec. 2.8.2.2

Define	allocate_vehicle()
Expansion	(VEHICLE_VARS *)heap_allocate(sizeof(VEHICLE_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_veh_id_list()
Expansion	(VEH_ID_LIST *)heap_allocate(sizeof(VEH_ID_LIST))
Location	/simnet/src/host/reporter.h Sec. 2.8.2.2

Define	allocate_weapon_systems()
Expansion	(WEAPON_SYSTEMS_VARS *)heap_allocate(sizeof (WEAPON_SYSTEMS_VARS))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	allocate_zone_cm()
Expansion	(ZONE_CM *)heap_allocate(sizeof (ZONE_CM))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	almost_eq(x, y, e)
Expansion	((abs((x) - (y)) <= (e)) ? TRUE : FALSE)
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	buffer_from_char(p)
Expansion	((BUFFER *)(p - sizeof(BUFFER)))
Location	/simnet/libsrc/libcomm/libcomm.h Sec. 2.14.4.3

Define	char_from_buffer(bp)
Expansion	((BYTE_POINTER)((BYTE_POINTER)bp + sizeof(BUFFER)))
Location	/simnet/libsrc/libcomm/libcomm.h Sec. 2.14.4.3

Define	coords_within_database(x,y)
Expansion	((appearance & (vehDestroyed | vehFlaming)) || ((appearance & vehMobilityDisabled) && ((appearance & vehFirepowerDisabled) || (simnet_class != vehicleClassTank))))
Location	/simnet/src/host/phantom.h Sec. 2.13.3.1

Define	copysign(x,y)
Expansion	(((x)>=0.0) ? (fabs(y)) : ((fabs(y)*-(1.0)))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	database_hitmodel_query(weapon, range, view)
Expansion	interpolate_curve((weapon[view + 2].array), range)
Location	/simnet/libsrc/libdatabase/libdatabase.h Sec. 2.6.8.9

Define	deallocate_airveh(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_area_cm(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	deallocate_cis(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/cis.h Sec. 2.10.1.2

Define	deallocate_collision(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_commander(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_composite(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_control_measure(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	deallocate_control_measure_list_item(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	deallocate_damage(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_detection(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_driver(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_entity(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_groundveh(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_line_cm(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	deallocate_logistics(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_missile(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/missile.h Sec. 2.6.9.2

Define	deallocate_navigator(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_overlay(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	deallocate_overlay_list_item(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	deallocate_pilot(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_point_cm(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	deallocate_predicates(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/predicates.h Sec. 2.10.1.4

Define	deallocate_rem_veh_id_list(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/reporter.h Sec. 2.8.2.2

Define	deallocate_reporter(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_route(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/route.h Sec. 2.10.2.5

Define	deallocate_routepoint(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/route.h Sec. 2.10.2.5

Define	deallocate_saf_vehicle(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_shell_summary(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/reporter.h Sec. 2.8.2.2

Define	deallocate_targeting(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_tickable(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_turret(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_veh_id_list(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/reporter.h Sec. 2.8.2.2

Define	deallocate_veh_cluster(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/reporter.h Sec. 2.8.2.2

Define	deallocate_vehicle(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_weapon_systems(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/veh_storage.h Sec. 2.9.1.2

Define	deallocate_zone_cm(x)
Expansion	(heap_deallocate(x))
Location	/simnet/src/host/cm.h Sec. 2.10.2.2

Define	ecmderr()
Expansion	{ printf("ecmderr\n");}
Location	/simnet/libsrc/libparser/kludge.c Sec. 2.5.1.10

Define	eq(x, y, e) /* (y-e) <= x <= (y+e) where e is an appropriate tolerance */
Expansion	((abs((x) - (y)) <= (e)) ? 1 : 0)
Location	/simnet/include/saf/src/tolerance.h Sec. 2.13.3.7

Define	extract_bit(bit_num, bitfield)
Expansion	(1 & ((bitfield)[(bit_num) / 8] >> ((bit_num) % 8)))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	gethost()
Expansion	{ printf("gethost\n");}
Location	/simnet/libsrc/libparser/kludge.c Sec. 2.5.1.10

Define	getnet()
Expansion	{ printf("getnett\n");}
Location	/simnet/libsrc/libparser/kludge.c Sec. 2.5.1.10

Define	hostname()
Expansion	{ printf("hostname\n");}
Location	/simnet/libsrc/libparser/kludge.c Sec. 2.5.1.10

Define	is_dead(appearance, simnet_class)
Expansion	((appearance & (vehDestroyed | vehFlaming)) || ((appearance & vehMobilityDisabled) && ((appearance & vehFirepowerDisabled) || (simnet_class != vehicleClassTank))))
Location	/simnet/src/host/phantom.h Sec. 2.13.3.1

Define	is_enemy(my_team,his_team)
Expansion	(((my_team == distinguishedForceID) && ((his_team == otherForceID) || (his_team == targetForceID))) || ((my_team == otherForceID) && (his_team == distinguishedForceID) || (his_team == targetForceID)))
Location	/simnet/src/host/phantom.h Sec. 2.13.3.1

Define	isbadhost()
Expansion	{ printf("isbadhost\n");}
Location	/simnet/libsrc/libparser/kludge.c Sec. 2.5.1.10

Define	isbadnett()
Expansion	{ printf("isbadnett\n");}
Location	/simnet/libsrc/libparser/kludge.c Sec. 2.5.1.10

Define	kph_to_mpf(s)
Expansion	((s) * (1000.0 / (3600.0 *15.0)))
Location	/simnet/src/host/phantom.h Sec. 2.13.3.1

Define	kph_to_mps(s)
Expansion	((s) *1000.0 / 3600.0)
Location	/simnet/src/host/phantom.h Sec. 2.13.3.1

Define	math_to_compass(rad)
Expansion	(angle_clip(PI_OVER_2-rad))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	max(a, b) /* Returns the maximum value of a and b */
Expansion	(((a) >= (b)) ? (a) : (b))
Location	/simnet/include/saf/src/minmax.h Sec. 2.13.3.5

Define	max(a, b) /* Returns the maximum value of a and b */
Expansion	(((a) >= (b)) ? (a) : (b))
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	min(a, b) /* Returns the minimum value of a and b */
Expansion	(((a) <= (b)) ? (a) : (b))
Location	/simnet/include/saf/src/minmax.h Sec. 2.13.3.5

Define	min(a, b) /* Returns the minimum value of a and b */
Expansion	(((a) <= (b)) ? (a) : (b))
Location	/simnet/libsrc/libpvis/pve.h Sec. 2.6.7.3

Define	min3(a, b, c)
Expansion	((a < b) ? ((a < c) ? a : c) : ((b < c) ? b : c))
Location	/simnet/libsrc/libsched/sched.c Sec. 2.2.1.5

Define	mpf_to_mps(s)
Expansion	((s) * 15.0)
Location	/simnet/src/host/phantom.h Sec. 2.13.3.1

Define	mps_to_knots(s)
Expansion	((s) * 1000.0 / 3600.0)
Location	/simnet/src/host/phantom.h Sec. 2.13.3.1

Define	mps_to_kph(s)
Expansion	((s) * 3600.0 / 1000.0)
Location	/simnet/src/host/phantom.h Sec. 2.13.3.1

Define	mps_to_mpf(s)
Expansion	((s) / 15.0)
Location	/simnet/src/host/phantom.h Sec. 2.13.3.1

Define	netfmt()
Expansion	{ printf("netfmt\n");}
Location	/simnet/libsrc/libparser/kludge.c Sec. 2.5.1.10

Define	p_arg0(x)
Expansion	(char *)((x)->p_command)
Location	/simnet/libsrc/libparser/libparser.h Sec. 2.5.1.11

Define	point_in_rect(pp, rp)
Expansion	(((pp[X]) >= (rp)->left) && ((pp[X]) <= (rp)->right) && ((pp[Y] >= (rp)bottom) && ((pp[Y]) <= (rp)->top))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	reflect_x_axis(y)
Expansion	{y = -y;}
Location	/simnet/libsrc/libquad/minimum_clip.c Sec. 2.12.1.13

Define	reflect_x_minus_y(x, y)
Expansion	{INT32 t; t = x; x= -y; y = -t;}
Location	/simnet/libsrc/libquad/minimum_clip.c Sec. 2.12.1.13

Define	rotate_90_c(x, y)
Expansion	{INT32 t; t = x; x= y; y = -t;}
Location	/simnet/libsrc/libquad/minimum_clip.c Sec. 2.12.1.13

Define	rotate_180_c(x, y)
Expansion	{x= -x; y = -y;}
Location	/simnet/libsrc/libquad/minimum_clip.c Sec. 2.12.1.13

Define	rotate_270_c(x, y)
Expansion	{INT32 t; t = x; x= -y; y = t;}
Location	/simnet/libsrc/libquad/minimum_clip.c Sec. 2.12.1.13

Define	s_atan2(x,y)
Expansion	((((x)==0.0)&&((y)==0.0)) ? 0.0 : atan2(x,y))
Location	/simnet/libsrc/libutil/libutil.h Sec. 2.14.3.9

Define	step(x, y, e) /* If |x-y| <= e then return z, else return x */
Expansion	((abs((x) - (z)) <= (e)) ? (z) : (x))
Location	/simnet/include/saf/src/tolerance.h Sec. 2.13.3.7

Define	symbols_compare(x,y)
Expansion	#ifndef NO_SYMBOLS ((int)(x)-(int)(y)) #else NO_SYMBOLS (strcmp((x),(y)))
Location	/simnet/libsrc/libreader/libreader.h Sec. 2.1.1.5

Define	symbols_match(x,y)
Expansion	#ifndef NO_SYMBOLS ((x)==(y)) #else NO_SYMBOLS (strcmp((x),(y))==0)
Location	/simnet/libsrc/libreader/libreader.h Sec. 2.1.1.5

Define	type_ok(v_type, match_type)
Expansion	((v_type) & (match_type))
Location	/simnet/src/host/iterator.h Sec. 2.9.3.2

APPENDIX B

This appendix consists of an alphabetical list of the CSUs in this document along with their directory paths and section numbers.

CSU	Path	Section
_queue_hanger_hang_this	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.3
_queue_hanger_unhang	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.4
abort_handler	/simnet/src/host/main.c	2.1.3.1.20
abs.h	/simnet/include/saf/src/abs.h	2.13.3.2
acquire_target	/simnet/src/host/gunner.c	2.6.9.5.1
action_drill_needed	/simnet/src/host/predicates.c	2.10.1.3.6
activate_new_cis	/simnet/src/host/cis.c	2.10.1.1.16
add_executing_unit	/simnet/src/host/cm.c	2.10.2.1.21
add_into_spotter_table	/simnet/src/host/spotter.c	2.8.2.3.9
add_last_hit	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.24
add_removed_vehicle	/simnet/src/host/reporter.c	2.8.2.1.29
add_shelling_to_cluster	/simnet/src/host/reporter.c	2.8.2.1.37
add_to_free_list	/simnet/libsrc/libheap/alloc.c	2.14.2.1.9
add_vehicle_to_cluster	/simnet/src/host/reporter.c	2.8.2.1.20
add_weapons_status	/simnet/src/host/weapons.c	2.6.9.8.12
adjust_behaviors	/simnet/src/host/cis.c	2.10.1.1.15
agl_to_abs_altitude	/simnet/src/host/pilot.c	2.6.4.2.9
air_raid_happening	/simnet/src/host/predicates.c	2.10.1.3.4
air_raid_over	/simnet/src/host/predicates.c	2.10.1.3.5
air_tick	/simnet/src/host/flyingveh.c	2.6.4.6.5
airveh_catastrophic_kill	/simnet/src/host/flyingveh.c	2.6.4.6.7
airveh_init	/simnet/src/host/flyingveh.c	2.6.4.6.1
airveh_mobility_kill	/simnet/src/host/flyingveh.c	2.6.4.6.6
airveh_show	/simnet/src/host/flyingveh.c	2.6.4.6.4
align_points	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.3
all_wide_segments_through_water	/simnet/libsrc/libquad/water_check.c	2.12.1.4.8
Alloc	/simnet/libsrc/libparser/par_unix.c	2.5.1.9.1
allocate_bitfield	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.1
allocate_points	/simnet/libsrc/libquad/terrain.c	2.12.1.1.5
allocate_safobj	/simnet/src/host/safobj.c	2.2.2.2
allocate_spotter_tables	/simnet/src/host/spotter.c	2.8.2.3.3
ammo_to_shell_type	/simnet/src/host/reporter.c	2.8.2.1.34
and_bitfield	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.9
angle_between_vectors	/simnet/libsrc/libutil/math.c	2.14.3.5.7
angle_clip	/simnet/libsrc/libutil/math.c	2.14.3.5.4
any_wide_segment_through_water	/simnet/libsrc/libquad/water_check.c	2.12.1.4.3
append_routepoint	/simnet/src/host/route.c	2.10.2.4.2
append_vector_2d_on_route	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.19
ApplicationGetSiteHost	/simnet/src/host/simnet.c	2.3.2.4
apply_to_blocks_of_size	/simnet/libsrc/libheap/alloc.c	2.14.2.1.19
arc_of_attention	/simnet/src/host/detection.c	2.6.5.1.9
are_we_there	/simnet/src/host/pilot.c	2.6.4.2.18
arty_type_string	/simnet/src/host/sbx.c	2.4.3.2.36
AssocGetSimAddress	/simnet/src/host/simnet.c	2.3.2.3
attackatwill_tick	/simnet/src/host/pilot.c	2.6.4.2.61
attackatwill_tick_new	/simnet/src/host/pilot.c	2.6.4.2.62
average	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.20
avoid_water_to_next_point	/simnet/src/host/route.c	2.10.2.4.11
Back1Char	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.12
BackChar	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.25
BEGIN_ARG_TABLE(SWITCH)	/simnet/src/host/main.c	2.1.3.1.18
BeginningOfLine	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.11

binarysearch_tag	/simnet/libsrc/libreader/tags.c	2.1.1.4.6
break_block	/simnet/libsrc/libheap/alloc.c	2.14.2.1.7
broadcast_appearance_data	/simnet/src/host/sbx.c	2.4.3.2.73
broadcast_echelon_data	/simnet/src/host/sbx.c	2.4.3.2.72
broadcast_pae_data	/simnet/src/host/sbx.c	2.4.3.2.74
broadcast_vehicle_is_gone	/simnet/src/host/sbx.c	2.4.3.2.75
buf_conn.h	/simnet/libsrc/libudp/buf_conn.h	2.4.2.6
buf_master_rudp_close	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.7
buf_master_rudp_open	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.4
buf_master_rudp_synch	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.5
buf_rudp_close	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.6
buf_rudp_discard_all_buffers	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.8
buf_rudp_disconnect	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.9
buf_rudp_flush	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.10
buf_rudp_open	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.3
buf_rudp_tick	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.13
buf_rudp_write_message	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.12
buff_rudp_read_message	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.11
buff_rudp_show	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.2
buffer__deallocate	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.14
buffer_allocate	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.12
buffer_allocate_from_pool	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.10
buffer_allocate_into_pool	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.12
buffer_deallocate	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.15
buffer_dequeue	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.20
buffer_describe	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.24
buffer_distribute	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.26
buffer_enqueue	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.18
buffer_exercise	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.19
buffer_flush	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.23
buffer_length	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.16
buffer_one_less	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.29
buffer_one_more	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.28
buffer_pool_allocate	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.13
buffer_pool_return_buffer	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.11
buffer_set_size	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.13
buffer_simple_dequeue	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.15
buffer_simple_enqueue	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.14
buffer_simple_flush	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.16
buffer_statistics_init	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.27
buffer_statistics_print	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.30
buffer_traverse_and_apply	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.21
buffer_traverse_and_apply_n_times	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.22
buffer_up_refcnt	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.17
build_packets	/simnet/src/host/blaster.c	2.15.1.2
building_thru	/simnet/libsrc/libquad/building_check.c	2.12.1.17.1
cancel_fncl	/simnet/libsrc/libsched/fncl.c	2.2.1.1.4
cancel_fncl_group	/simnet/libsrc/libsched/fncl.c	2.2.1.1.5
cancel_overlay	/simnet/src/host/dispatch.c	2.14.1.1.10
canopies	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.18
center_column	/simnet/libsrc/libquad/minimum_clip.c	2.12.1.13.7
change_fncl_period	/simnet/libsrc/libsched/fncl.c	2.2.1.1.6
change_movement_and_report	/simnet/src/host/cis.c	2.10.1.1.19
change_tick_rate	/simnet/src/host/tickable.c	2.2.3.5
ChangePrompt	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.1
check_box	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.31
check_edge_hit	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.11
check_edges	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.8

check_hits	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.21
check_object	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.14
check_predicates	/simnet/src/host/predicates.c	2.10.1.3.13
check_prob	/simnet/libsrc/libutil/random.c	2.14.3.7.2
check_tree_hits	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.29
checksum	/simnet/libsrc/libutil/checksum.c	2.14.3.2.1
choose_resupply_item	/simnet/src/host/logistics.c	2.6.6.1.7
choose_skirt_point	/simnet/src/host/collision.c	2.6.3.1.16
choose_target_and_weapon	/simnet/src/host/targeting.c	2.6.9.3.20
cis.h	/simnet/src/host/cis.h	2.10.1.2
cis_show	/simnet/src/host/cis.c	2.10.1.1.10
cistrcmp	/simnet/libsrc/libutil/cstring.c	2.14.3.8.2
cistrncmp	/simnet/libsrc/libutil/cstring.c	2.14.3.8.3
clear_bit	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.4
clear_bitfield	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.6
clear_cis_stack	/simnet/src/host/cis.c	2.10.1.1.14
clear_detection	/simnet/src/host/detection.c	2.6.5.1.2
clear_weapons_status	/simnet/src/host/weapons.c	2.6.9.8.11
ClearLine	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.19
clip_angle_negative	/simnet/libsrc/libutil/math.c	2.14.3.5.6
clip_angle_positive	/simnet/libsrc/libutil/math.c	2.14.3.5.5
clip_to_tdb	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.3
closest_leader	/simnet/src/host/formation.c	2.8.1.4.7
closest_to_a_leader	/simnet/src/host/formation.c	2.8.1.4.6
cluster_from_vehicle	/simnet/src/host/reporter.c	2.8.2.1.25
cluster_shell	/simnet/src/host/reporter.c	2.8.2.1.39
cluster_vehicle	/simnet/src/host/reporter.c	2.8.2.1.24
cm.h	/simnet/src/host/cm.h	2.10.2.2
cm_add_area	/simnet/src/host/cm.c	2.10.2.1.9
cm_add_line	/simnet/src/host/cm.c	2.10.2.1.11
cm_add_point	/simnet/src/host/cm.c	2.10.2.1.8
cm_add_route	/simnet/src/host/cm.c	2.10.2.1.12
cm_add_zone	/simnet/src/host/cm.c	2.10.2.1.10
cm_applies_to_units	/simnet/src/host/cm.c	2.10.2.1.16
cm_count_intersections	/simnet/src/host/navigator.c	2.10.2.3.17
cm_delete_cm	/simnet/src/host/cm.c	2.10.2.1.14
cm_delete_cm_from_overlay	/simnet/src/host/cm.c	2.10.2.1.15
cm_delete_overlay	/simnet/src/host/cm.c	2.10.2.1.13
cm_execute_overlay	/simnet/src/host/cm.c	2.10.2.1.20
cm_executed_in_list	/simnet/src/host/cm.c	2.10.2.1.18
cm_find_cm	/simnet/src/host/cm.c	2.10.2.1.7
cm_find_cm_list_item	/simnet/src/host/cm.c	2.10.2.1.6
cm_find_overlay	/simnet/src/host/cm.c	2.10.2.1.5
cm_find_overlay_list_item	/simnet/src/host/cm.c	2.10.2.1.4
cm_force_overlay_recalculation	/simnet/src/host/cm.c	2.10.2.1.23
cm_get_center_of_mass	/simnet/src/host/navigator.c	2.10.2.3.15
cm_get_cm_from_overlay	/simnet/src/host/cm.c	2.10.2.1.17
cm_intersection	/simnet/src/host/navigator.c	2.10.2.3.13
cm_point_inside_polygon	/simnet/src/host/navigator.c	2.10.2.3.16
cm_point_line_intersection	/simnet/src/host/navigator.c	2.10.2.3.18
cm_poly_intersection	/simnet/src/host/navigator.c	2.10.2.3.14
cm_recompute_cm_from_overlay	/simnet/src/host/cm.c	2.10.2.1.19
cm_to_passage_string	/simnet/src/host/cis.c	2.10.1.1.17
cmp_form_entries	/simnet/src/host/formation.c	2.8.1.4.11
cmp_tags	/simnet/libsrc/libreader/tags.c	2.1.1.4.4
collect	/simnet/libsrc/libheap/alloc.c	2.14.2.1.6
collect_perf_stat	/simnet/libsrc/libsched/perf.c	2.2.1.4.5
collision_dead_tick	/simnet/src/host/collision.c	2.6.3.1.7

collision_disengage	/simnet/src/host/collision.c	2.6.3.1.5
collision_remove_vehicle	/simnet/src/host/collision.c	2.6.3.1.3
collision_show	/simnet/src/host/collision.c	2.6.3.1.4
collision_tick	/simnet/src/host/collision.c	2.6.3.1.6
collision_vehicle_rammed	/simnet/src/host/collision.c	2.6.3.1.12
combined_velocity	/simnet/src/host/pilot.c	2.6.4.2.14
commander.h	/simnet/src/host/commander.h	2.8.1.2
commander_attack	/simnet/src/host/commander.c	2.8.1.1.27
commander_cancel_overlay	/simnet/src/host/commander.c	2.8.1.1.8
commander_change_altitude	/simnet/src/host/commander.c	2.8.1.1.26
commander_change_formation	/simnet/src/host/commander.c	2.8.1.1.16
commander_change_speed	/simnet/src/host/commander.c	2.8.1.1.14
commander_execute_overlay	/simnet/src/host/commander.c	2.8.1.1.7
commander_executing_order	/simnet/src/host/commander.c	2.8.1.1.6
commander_face_direction	/simnet/src/host/commander.c	2.8.1.1.21
commander_follow_vehicle	/simnet/src/host/commander.c	2.8.1.1.17
commander_get_mission_status	/simnet/src/host/commander.c	2.8.1.1.5
commander_goto_point	/simnet/src/host/commander.c	2.8.1.1.19
commander_halt	/simnet/src/host/commander.c	2.8.1.1.11
commander_inferior_changed_status	/simnet/src/host/commander.c	2.8.1.1.22
commander_land	/simnet/src/host/commander.c	2.8.1.1.25
commander_note_leader_state	/simnet/src/host/commander.c	2.8.1.1.23
commander_restore_leader_state	/simnet/src/host/commander.c	2.8.1.1.24
commander_resume_mission	/simnet/src/host/commander.c	2.8.1.1.20
commander_set_mission_direction	/simnet/src/host/commander.c	2.8.1.1.13
commander_set_mission_formation	/simnet/src/host/commander.c	2.8.1.1.15
commander_set_mission_speed	/simnet/src/host/commander.c	2.8.1.1.12
commander_show	/simnet/src/host/commander.c	2.2.1.1.4
commander_simulator_in_command	/simnet/src/host/commander.c	2.8.1.1.18
commander_state_string	/simnet/src/host/commander.c	2.2.1.1.3
commander_tick	/simnet/src/host/commander.c	2.8.1.1.9
CommandLog	/simnet/libsrc/libparser/par_hist.c	2.5.1.8.1
compare_targets	/simnet/src/host/targeting.c	2.6.9.3.14
composite_add_inferior_composite	/simnet/src/host/composite.c	2.8.1.3.5
composite_add_inferior_vehicle	/simnet/src/host/composite.c	2.8.1.3.7
composite_add_member_vehicle	/simnet/src/host/composite.c	2.8.1.3.9
composite_assume_formation	/simnet/src/host/composite.c	2.8.1.3.26
composite_assume_formation_internal	/simnet/src/host/composite.c	2.8.1.3.25
composite_attack	/simnet/src/host/composite.c	2.8.1.3.51
composite_cancel_overlay	/simnet/src/host/composite.c	2.8.1.3.42
composite_change_altitude	/simnet/src/host/composite.c	2.8.1.3.53
composite_change_formation	/simnet/src/host/composite.c	2.8.1.3.47
composite_change_speed	/simnet/src/host/composite.c	2.8.1.3.44
composite_check_unit_strength	/simnet/src/host/composite.c	2.8.1.3.17
composite_distinguished_member	/simnet/src/host/composite.c	2.8.1.3.16
composite_execute_overlay	/simnet/src/host/composite.c	2.8.1.3.40
composite_executing_order	/simnet/src/host/composite.c	2.8.1.3.35
composite_face_direction	/simnet/src/host/composite.c	2.8.1.3.50
composite_fake_resupply	/simnet/src/host/composite.c	2.8.1.3.32
composite_fill_in_appearance_data	/simnet/src/host/composite.c	2.8.1.3.14
composite_fill_in_echelon_data	/simnet/src/host/composite.c	2.8.1.3.12
composite_fill_in_position_data	/simnet/src/host/composite.c	2.8.1.3.13
composite_find_living_vehicle	/simnet/src/host/composite.c	2.8.1.3.15
composite_find_vehicle_for_stealth	/simnet/src/host/composite.c	2.8.1.3.29
composite_follow_vehicle	/simnet/src/host/composite.c	2.8.1.3.45
composite_generate_status_report	/simnet/src/host/composite.c	2.8.1.3.33

composite_get_composite_bumpers	/simnet/src/host/composite.c	2.8.1.3.22
composite_go_away	/simnet/src/host/composite.c	2.8.1.3.2
composite_goto_point	/simnet/src/host/composite.c	2.8.1.3.48
composite_halt	/simnet/src/host/composite.c	2.8.1.3.43
composite_indirect_fire	/simnet/src/host/composite.c	2.8.1.3.21
composite_inferior_changed_status	/simnet/src/host/composite.c	2.8.1.3.37
composite_land	/simnet/src/host/composite.c	2.8.1.3.52
composite_note_leader_state	/simnet/src/host/composite.c	2.8.1.3.36
composite_note_member_vehicle_has_died	/simnet/src/host/composite.c	2.8.1.3.38
composite_pro_sim	/simnet/src/host/composite.c	2.8.1.3.20
composite_readjust_overlay	/simnet/src/host/composite.c	2.8.1.3.41
composite_reassign_current_formation	/simnet/src/host/composite.c	2.8.1.3.27
composite_rejoin_unit	/simnet/src/host/composite.c	2.8.1.3.39
composite_remove_inferior_composite	/simnet/src/host/composite.c	2.8.1.3.6
composite_remove_inferior_vehicle	/simnet/src/host/composite.c	2.8.1.3.8
composite_remove_member_vehicle	/simnet/src/host/composite.c	2.8.1.3.10
composite_remove_vehicles	/simnet/src/host/composite.c	2.8.1.3.34
composite_resume_mission	/simnet/src/host/composite.c	2.8.1.3.49
composite_send_unit_strength_message	/simnet/src/host/composite.c	2.8.1.3.18
composite_set_recursive_ids	/simnet/src/host/composite.c	2.8.1.3.23
composite_set_sbx_ids	/simnet/src/host/composite.c	2.8.1.3.24
composite_set_superior	/simnet/src/host/composite.c	2.8.1.3.4
composite_set_targeting_parameters	/simnet/src/host/composite.c	2.8.1.3.31
composite_show	/simnet/src/host/composite.c	2.8.1.3.3
composite_simulator_in_command	/simnet/src/host/composite.c	2.8.1.3.46
composite_start_ticking	/simnet/src/host/composite.c	2.8.1.3.11
composite_teleport_to_station	/simnet/src/host/composite.c	2.8.1.3.28
composite_tick	/simnet/src/host/composite.c	2.8.1.3.19
compute_damage_keys	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.11
compute_enemy_weight	/simnet/src/host/detection.c	2.6.5.1.7
compute_explosion_point	/simnet/src/host/missile.c	2.6.9.1.9
compute_interest_direction	/simnet/src/host/detection.c	2.6.5.1.8
compute_mid	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.10
compute_situation	/simnet/src/host/pilot.c	2.6.4.2.20
compute_who_follows_whom	/simnet/src/host/formation.c	2.8.1.4.4
cons	/simnet/libsrc/libquad/list.c	2.12.1.7.1
copy_bitfield	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.5
copy_floating_point_list_real_point_list	/simnet/src/host/cm.c	2.10.2.1.1
copy_list	/simnet/libsrc/libquad/list.c	2.12.1.7.14
copy_matrix_row_to_vector	/simnet/libsrc/libutil/math.c	2.14.3.5.18
copy_stack_to_array	/simnet/libsrc/libreader/parser.y	2.1.1.1.5
copy_xy_on_tdb	/simnet/src/host/misc.c	2.14.1.2.5
CopyEarlierCommand	/simnet/libsrc/libparser/par_hist.c	2.5.1.8.2
count_bits	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.10
count_forces	/simnet/src/host/parser.c	2.5.2.1.13
count_hosts	/simnet/src/host/parser.c	2.5.2.1.15
count_intersections	/simnet/libsrc/libquad/intersection.c	2.12.1.14.4
count_sites	/simnet/src/host/parser.c	2.5.2.1.14
count_vehicles	/simnet/src/host/parser.c	2.5.2.1.12

countries_from_battle_scheme_and_force		
	/simnet/src/host/parser.c	2.5.2.1.34
country_to_string	/simnet/src/host/sbx.c	2.4.3.2.28
create_airveh	/simnet/src/host/flyingveh.c	2.6.4.6.2
create_bridges	/simnet/libsrc/libquad/bridges.c	2.12.1.3.1
create_buildings	/simnet/libsrc/libquad/trees.c	2.12.1.16.5
create_cis	/simnet/src/host/cis.c	2.10.1.1.8
create_collision	/simnet/src/host/collision.c	2.6.3.1.1
create_commander	/simnet/src/host/commander.c	2.8.1.1.1
create_composite	/simnet/src/host/composite.c	2.8.1.3.1
create_contours	/simnet/libsrc/libquad/trees.c	2.12.1.16.7
create_damage	/simnet/src/host/damage.c	2.6.2.1.1
create_detection	/simnet/src/host/detection.c	2.6.5.1.1
create_driver	/simnet/src/host/driver.c	2.6.3.2.1
create_entity	/simnet/src/host/entity.c	2.9.2.1.1
create_groundveh	/simnet/src/host/groundveh.c	2.6.3.4.1
create_lakes	/simnet/libsrc/libquad/bridges.c	2.12.1.3.3
create_logistics	/simnet/src/host/logistics.c	2.6.6.1.1
create_missile	/simnet/src/host/missile.c	2.6.9.1.3
create_navigator	/simnet/src/host/navigator.c	2.10.2.3.1
create_new_cluster	/simnet/src/host/reporter.c	2.8.2.1.19
create_new_shelling	/simnet/src/host/reporter.c	2.8.2.1.36
create_pilot	/simnet/src/host/pilot.c	2.6.4.2.2
create_predicates	/simnet/src/host/predicates.c	2.10.1.3.1
create_remote	/simnet/usr/host/remote.c	2.7.1.1
create_remote_vehicle	/simnet/usr/host/remote.c	2.7.1.17
create_reporter	/simnet/src/host/reporter.c	2.8.2.1.1
create_river_ints	/simnet/libsrc/libquad/rivers.c	2.12.1.2.3
create_rivers	/simnet/libsrc/libquad/rivers.c	2.12.1.2.1
create_road_ints	/simnet/libsrc/libquad/roads.c	2.12.1.15.3
create_roads	/simnet/libsrc/libquad/roads.c	2.12.1.15.1
create_routepoint	/simnet/src/host/route.c	2.10.2.4.3
create_saf_vehicle	/simnet/src/host/saf_vehicle.c	2.6.1.1.1
create_spotter	/simnet/src/host/spotter.c	2.8.2.3.1
create_targeting	/simnet/src/host/targeting.c	2.6.9.3.2
create_tickable	/simnet/src/host/tickable.c	2.2.3.1
create_tree_canopies	/simnet/libsrc/libquad/trees.c	2.12.1.16.3
create_trees	/simnet/libsrc/libquad/trees.c	2.12.1.16.1
create_turret	/simnet/src/host/turret.c	2.6.9.6.1
create_unit	/simnet/src/host/create.c	2.11.1.2
create_vehicle	/simnet/src/host/vehicle.c	2.9.2.2.1
create_weapon_systems	/simnet/src/host/weapons.c	2.6.9.8.2
crossing_location	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.9
damage_indirect_fire	/simnet/src/host/damage.c	2.6.2.1.6
damage_string	/simnet/src/host/damage.c	2.6.2.1.3
damage_vehicle_impact	/simnet/src/host/damage.c	2.6.2.1.4
damage_vehicle_rammed	/simnet/src/host/damage.c	2.6.2.1.5
database_detection_query	/simnet/libsrc/libdatabase/detection.c	2.6.8.2.2
database_df_damage_query	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.10
database_echelon_lobj_query	/simnet/libsrc/libdatabase/echelon.c	2.6.8.5.3
database_echelon_response_formation_query		
	/simnet/libsrc/libdatabase/echelon.c	2.6.8.5.4
database_echelon_template_query	/simnet/libsrc/libdatabase/echelon.c	2.6.8.5.2
database_formation_sub_form_query		
	/simnet/libsrc/libdatabase/formation.c	2.6.8.4.3
database_if_damage_query	/simnet/libsrc/libdatabase/if_damage.c	2.6.8.7.6
database_information_form_query	/simnet/libsrc/libdatabase/formation.c	2.6.8.4.2
database_init	/simnet/src/host/main.c	2.1.3.1.6

database_read	/simnet/libsrc/libdatabase/database.c	2.6.8.8.1
database_read_echelon	/simnet/libsrc/libdatabase/echelon.c	2.6.8.5.1
database_read_formation	/simnet/libsrc/libdatabase/formation.c	2.6.8.4.1
deallocate_bitfield	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.2
deallocate_safobj	/simnet/src/host/safobj.c	2.2.2.3
deallocate_spotter_tables	/simnet/src/host/spotter.c	2.8.2.3.4
debug.h	/simnet/src/host/debug.h	2.5.2.2
debugging_on	/simnet/src/host/main.c	2.1.3.1.7
deferred_fncl	/simnet/libsrc/libsched/fncl.c	2.2.1.1.1
delete	/simnet/libsrc/libquad/list.c	2.12.1.7.7
Delete1Backward	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.17
Delete1Forward	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.13
delete_last_item	/simnet/libsrc/libquad/list.c	2.12.1.7.12
DeleteChar	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.26
DeleteWord	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.22
describe_symbol_table	/simnet/libsrc/libreader/symbol.c	2.1.1.3.4
destroy_airveh	/simnet/src/host/flyingveh.c	2.6.4.6.3
destroy_area_cm	/simnet/src/host/cm.c	2.10.2.1.26
destroy_cis	/simnet/src/host/cis.c	2.10.1.1.9
destroy_cm	/simnet/src/host/cm.c	2.10.2.1.29
destroy_cm_data	/simnet/src/host/cm.c	2.10.2.1.24
destroy_cm_list	/simnet/src/host/cm.c	2.10.2.1.31
destroy_cm_list_item	/simnet/src/host/cm.c	2.10.2.1.30
destroy_collision	/simnet/src/host/collision.c	2.6.3.1.2
destroy_commander	/simnet/src/host/commander.c	2.8.1.1.2
destroy_damage	/simnet/src/host/damage.c	2.6.2.1.2
destroy_detection	/simnet/src/host/detection.c	2.6.5.1.3
destroy_driver	/simnet/src/host/driver.c	2.6.3.2.2
destroy_entity	/simnet/src/host/entity.c	2.9.2.1.2
destroy_grid_entry_list	/simnet/src/host/iterator.c	2.9.3.1.13
destroy_groundveh	/simnet/src/host/groundveh.c	2.6.3.4.2
destroy_line_cm	/simnet/src/host/cm.c	2.10.2.1.28
destroy_logistics	/simnet/src/host/logistics.c	2.6.6.1.2
destroy_missile	/simnet/src/host/missile.c	2.6.9.1.4
destroy_navigator	/simnet/src/host/navigator.c	2.10.2.3.2
destroy_overlay	/simnet/src/host/cm.c	2.10.2.1.32
destroy_overlay_list_item	/simnet/src/host/cm.c	2.10.2.1.33
destroy_pilot	/simnet/src/host/pilot.c	2.6.4.2.4
destroy_point_cm	/simnet/src/host/cm.c	2.10.2.1.25
destroy_predicates	/simnet/src/host/predicates.c	2.10.1.3.2
destroy_rem_veh_id_item	/simnet/src/host/reporter.c	2.8.2.1.15
destroy_removed_vehicles	/simnet/src/host/reporter.c	2.8.2.1.14
destroy_reporter	/simnet/src/host/reporter.c	2.8.2.1.2
destroy_route	/simnet/src/host/route.c	2.10.2.4.4
destroy_routepoint	/simnet/src/host/route.c	2.10.2.4.5
destroy_shell_summary	/simnet/src/host/reporter.c	2.8.2.1.17
destroy_shellings	/simnet/src/host/reporter.c	2.8.2.1.16
destroy_spotter	/simnet/src/host/spotter.c	2.8.2.3.2
destroy_targeting	/simnet/src/host/targeting.c	2.6.9.3.3
destroy_tickable	/simnet/src/host/tickable.c	2.2.3.2
destroy_turret	/simnet/src/host/turret.c	2.6.9.6.2
destroy_veh_id_item	/simnet/src/host/reporter.c	2.8.2.1.13
destroy_veh_id_list	/simnet/src/host/reporter.c	2.8.2.1.12
destroy_vehicle	/simnet/src/host/vehicle.c	2.9.2.2.2
destroy_vehicle_cluster	/simnet/src/host/reporter.c	2.8.2.1.11
destroy_vehicle_clusters	/simnet/src/host/reporter.c	2.8.2.1.10
destroy_weapon_systems	/simnet/src/host/weapons.c	2.6.9.8.3
destroy_zone_cm	/simnet/src/host/cm.c	2.10.2.1.27

detect_building_on_path_tick	/simnet/src/host/collision.c	2.6.3.1.10
detect_collision_tick	/simnet/src/host/collision.c	2.6.3.1.13
detect_imminent_collision_tick	/simnet/src/host/collision.c	2.6.3.1.8
detectable	/simnet/src/host/detection.c	2.6.5.1.10
detection_remove_vehicle	/simnet/src/host/detection.c	2.6.5.1.4
detection_save_object	/simnet/src/host/detection.c	2.6.5.1.12
detection_show	/simnet/src/host/detection.c	2.6.5.1.5
detection_show_type	/simnet/src/host/detection.c	2.6.5.1.6
detection_tick	/simnet/src/host/detection.c	2.6.5.1.11
diffraction_diff	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.30
diffv	/simnet/src/host/misc.c	2.14.1.2.9
dir_to_tan_point	/simnet/src/host/pilot.c	2.6.4.2.10
distance	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11 1
distance_around_path	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.16
Do_Hash	/simnet/libsrc/libreader/symbol.c	2.1.1.3.5
do_heap_collect	/simnet/libsrc/libheap/alloc.c	2.14.2.1.18
do_heap_statistics	/simnet/libsrc/libheap/alloc.c	2.14.2.1.17
do_heap_verify	/simnet/libsrc/libheap/alloc.c	2.14.2.1.16
do_send	/simnet/src/host/blaster.c	2.15.1.3
DoCommand	/simnet/libsrc/libparser/par_util.c	2.5.1.7.1
DoEscape	/simnet/libsrc/libparser/par_util.c	2.5.1.7.3
DoHelp	/simnet/libsrc/libparser/par_util.c	2.5.1.7.2
dotv	/simnet/src/host/misc.c	2.14.1.2.8
driver.h	/simnet/src/host/driver.h	2.6.3.3
driver_change_speed_cmd	/simnet/src/host/driver.c	2.6.3.2.33
driver_execute_overlay	/simnet/src/host/driver.c	2.6.3.2.17
driver_executing_immediate_command	/simnet/src/host/driver.c	2.6.3.2.15
driver_face_direction_cmd	/simnet/src/host/driver.c	2.6.3.2.30
driver_follow_leader	/simnet/src/host/driver.c	2.6.3.2.19
driver_follow_vehicle_cmd	/simnet/src/host/driver.c	2.6.3.2.28
driver_forget_about_forps	/simnet/src/host/driver.c	2.6.3.2.21
driver_goto_point_cmd	/simnet/src/host/driver.c	2.6.3.2.29
driver_halt_command	/simnet/src/host/driver.c	2.6.3.2.27
driver_mission_completed	/simnet/src/host/driver.c	2.6.3.2.16
driver_remove_vehicles	/simnet/src/host/driver.c	2.6.3.2.3
driver_resume_cmd	/simnet/src/host/driver.c	2.6.3.2.32
driver_resume_from_collision	/simnet/src/host/driver.c	2.6.3.2.26
driver_resume_mission_command	/simnet/src/host/driver.c	2.6.3.2.31
driver_set_direction	/simnet/src/host/driver.c	2.6.3.2.24
driver_set_direction_cmd	/simnet/src/host/driver.c	2.6.3.2.34
driver_set_leader_mis	/simnet/src/host/driver.c	2.6.3.2.18
driver_set_route	/simnet/src/host/driver.c	2.6.3.2.22
driver_set_route_direction_cmd	/simnet/src/host/driver.c	2.6.3.2.35
driver_set_routedirection	/simnet/src/host/driver.c	2.6.3.2.25
driver_set_speed	/simnet/src/host/driver.c	2.6.3.2.23
driver_show	/simnet/src/host/driver.c	2.6.3.2.43
driver_simulator_in_command	/simnet/src/host/driver.c	2.6.3.2.12
driver_stop_mission	/simnet/src/host/driver.c	2.6.3.2.20
driver_tick	/simnet/src/host/driver.c	2.6.3.2.4
eliminate_duplicates	/simnet/src/host/route.c	2.10.2.4.9
eliminate_vehicles	/simnet/src/host/saf.c	2.2.4.4
EmptyLine	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.10
enable_predicate	/simnet/src/host/predicates.c	2.10.1.3.11
enable_predicates	/simnet/src/host/predicates.c	2.10.1.3.12
EndOfLine	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.14
enemy_disposition	/simnet/src/host/predicates.c	2.10.1.3.8
enqueue_on_rcvq	/simnet/src/host/simnet.c	2.3.2.10

entity_fill_in_appearance_data	/simnet/src/host/entity.c	2.9.2.1.6
entity_fill_in_position_data	/simnet/src/host/entity.c	2.9.2.1.5
entity_show	/simnet/src/host/entity.c	2.9.2.1.4
exagerate_bridges	/simnet/src/host/route.c	2.10.2.4.10
execute_overlay	/simnet/src/host/dispatch.c	2.14.1.1.8
exit_all_sbx_conns	/simnet/src/host/sbx.c	2.4.3.2.2
exit_handler	/simnet/src/host/main.c	2.1.3.1.21
expand_points	/simnet/libsrc/libquad/building_check.c	2.12.1.17.5
expand_road_route	/simnet/src/host/route.c	2.10.2.4.6
extend_bridge	/simnet/libsrc/libquad/extend_crossings.c	2.12.1.10.3
extend_crossing	/simnet/libsrc/libquad/extend_crossings.c	2.12.1.10.1
extend_intersection	/simnet/libsrc/libquad/extend_crossings.c	2.12.1.10.2
extend_segment	/simnet/libsrc/libquad/extend_crossings.c	2.12.1.10.4
fake_resupply	/simnet/src/host/dispatch.c	2.14.1.1.12
FieldGet	/simnet/libsrc/libparser/par_fields.c	2.5.1.5.3
fill_generic_cis_report	/simnet/src/host/cis.c	2.10.1.1.18
fill_in_appearance_data	/simnet/src/host/dispatch.c	2.14.1.1.1
fill_in_echelon_data	/simnet/src/host/dispatch.c	2.14.1.1.2
fill_in_position_data	/simnet/src/host/dispatch.c	2.14.1.1.3
fill_sbx_opfor_header	/simnet/src/host/sbx.c	2.4.3.2.18
final_relax_points	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.12
find_a_quad	/simnet/libsrc/libquad/search.c	2.12.1.5.1
find_closer_crossing	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.7
find_closest_building	/simnet/libsrc/libquad/building_check.c	2.12.1.17.3
find_direction_at_crossing	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.3
find_extra_contour_information	/simnet/libsrc/libquad/trees.c	2.12.1.16.8
find_first_vector	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.5
find_next_point	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.8
find_river_bend_points	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.10
find_river_points	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.2
find_route	/simnet/libsrc/libquad/water_avoidance.c	2.12.1.8.1
find_route_core	/simnet/libsrc/libquad/water_avoidance.c	2.12.1.8.2
find_segment_cross_points	/simnet/libsrc/libquad/extend_crossings.c	2.12.1.10.5
find_suitable_crossing_route	/simnet/libsrc/libquad/water_avoidance.c	2.12.1.8.4
find_tag	/simnet/libsrc/libreader/tags.c	2.1.1.4.3
find_tag_sorted	/simnet/libsrc/libreader/tags.c	2.1.1.4.7
fire_missile_at_target	/simnet/src/host/missile.c	2.6.9.1.6
fire_weapon_at_target	/simnet/src/host/weapons.c	2.6.9.8.21
firestatus_to_string	/simnet/src/host/targeting.c	2.6.9.3.4
fix_coords	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.28
fly_missile	/simnet/src/host/missile.c	2.6.9.1.11
fly_round	/simnet/src/host/weapons.c	2.6.9.8.22
flyingveh.h	/simnet/src/host/flyingveh.h	2.6.4.7
fmat_to_rmat	/simnet/libsrc/libutil/math.c	2.14.3.5.16
follow_lake_around	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.8
follow_water_segment	/simnet/libsrc/libquad/water_avoidance.c	2.12.1.8.3
followvehicle	/simnet/src/host/driver.c	2.6.3.2.14
forceID_to_string	/simnet/src/host/entity.c	2.9.2.1.3
Forward1Char	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.15
ForwardChar	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.24
fragment_route	/simnet/src/host/route.c	2.10.2.4.12
Free	/simnet/libsrc/libparser/par_unix.c	2.5.1.9.2
free_crossings	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.20
free_function	/simnet/libsrc/libsched/maint.c	2.2.1.3.2
free_list	/simnet/libsrc/libquad/list.c	2.12.1.7.4
free_stack	/simnet/libsrc/libreader/parser.y	2.1.1.1.3
free_water_list	/simnet/libsrc/libquad/water_check.c	2.12.1.4.2
ft_float	/simnet/src/host/misc.c	2.14.1.2.12

ft_int	/simnet/src/host/misc.c	2.14.1.2.11
ft_symbol	/simnet/src/host/misc.c	2.14.1.2.13
ft_table	/simnet/src/host/misc.c	2.14.1.2.14
ft_untagged_table	/simnet/src/host/misc.c	2.14.1.2.15
fvec_copy	/simnet/libsrc/libutil/math.c	2.14.3.5.17
fvec_to_rvec	/simnet/libsrc/libutil/math.c	2.14.3.5.13
gasp	/simnet/src/host/main.c	2.1.3.1.1
generate_a_deactivate	/simnet/src/host/pro_sim.c	2.3.1.1
generate_in_place_route	/simnet/src/host/route.c	2.10.2.4.8
generate_indirect_fire_packet	/simnet/src/host/pro_sim.c	2.3.1.4
generate_route_from_route_msg	/simnet/src/host/route.c	2.10.2.4.7
generate_status_report	/simnet/src/host/dispatch.c	2.14.1.1.11
generate_weapon_hit	/simnet/src/host/weapons.c	2.6.9.8.24
generate_weapon_miss	/simnet/src/host/weapons.c	2.6.9.8.25
get_a_leader	/simnet/src/host/commander.c	2.8.1.1.10
get_args	/simnet/libsrc/libsched/fncl.c	2.2.1.1.3
get_building_indecies	/simnet/libsrc/libquad/building_check.c	2.12.1.17.4
get_control_message	/simnet/src/host/cm.c	2.10.2.1.3
get_formation_layout	/simnet/src/host/formation.c	2.8.1.4.1
get_guises	/simnet/src/host/saf_vehicle.c	2.6.1.1.8
get_i_and_t_from_normal	/simnet/src/host/impact.c	2.6.4.5.2
get_impact_and_trajectory	/simnet/src/host/impact.c	2.6.4.5.1
get_leader_state	/simnet/src/host/driver.c	2.6.3.2.13
get_local_address	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.6
get_me_a_random_fraction	/simnet/libsrc/libutil/random.c	2.14.3.7.1
get_mid_pt	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.9
get_millisecond_time	/simnet/libsrc/libutil/time.c	2.14.3.3.2
get_next_circle_point	/simnet/src/host/pilot.c	2.6.4.2.23
get_overlay	/simnet/src/host/cm.c	2.10.2.1.2
get_quad_nodes	/simnet/libsrc/libquad/search.c	2.12.1.5.2
get_quad_nodes_internals	/simnet/libsrc/libquad/search.c	2.12.1.5.3
get_quads_in_region	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.13
get_quads_in_rt_point_region	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.14
get_quads_in_vector_region	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.15
get_relative_vehicle_agility	/simnet/src/host/weapons.c	2.6.9.8.20
get_remote_address	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.7
get_sbx_from_port	/simnet/src/host/sbx.c	2.4.3.2.5
get_sender_of_last_packet	/simnet/src/host/simnet.c	2.3.2.15
get_station_info	/simnet/src/host/formation.c	2.8.1.4.10
get_string	/simnet/libsrc/libreader/symbol.c	2.1.1.3.3
get_symbol	/simnet/libsrc/libreader/symbol.c	2.1.1.3.2
get_symbol_value	/simnet/libsrc/libreader/symbol.c	2.1.1.3.6
get_turret	/simnet/src/host/formation.c	2.8.1.4.2
get_unit_direction	/simnet/src/host/driver.c	2.6.3.2.11
get_water_indices	/simnet/libsrc/libquad/water_check.c	2.12.1.4.1
globals.h	/simnet/src/host/globals.h	2.13.1.1
go_away	/simnet/src/host/dispatch.c	2.14.1.1.6
GobbleWord	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.16
grid_locword	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.27
ground_impact_to_sbx	/simnet/src/host/sbx.c	2.4.3.2.24
ground_level	/simnet/src/host/pilot.c	2.6.4.2.8
groundveh_close_enough	/simnet/src/host/groundveh.c	2.6.3.4.6
groundveh_mobility_kill	/simnet/src/host/groundveh.c	2.6.3.4.5
groundveh_move	/simnet/src/host/groundveh.c	2.6.3.4.3
groundveh_tick	/simnet/src/host/groundveh.c	2.6.3.4.4
gunner_round_fired	/simnet/src/host/gunner.c	2.6.9.5.6
gunner_tick	/simnet/src/host/gunner.c	2.6.9.5.7
has_this_packet_been_acked	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.14

hasty_attack_needed	/simnet/src/host/predicates.c	2.10.1.3.3
heading_to_compass	/simnet/src/host/reporter.c	2.8.2.1.5
heap_allocate	/simnet/libsrc/libheap/alloc.c	2.14.2.1.1
heap_allocate_distributed	/simnet/libsrc/libheap/alloc.c	2.14.2.1.3
heap_calloc	/simnet/libsrc/libheap/alloc.c	2.14.2.1.2
heap_create	/simnet/libsrc/libheap/create.c	2.14.2.2.2
heap_deallocate	/simnet/libsrc/libheap/alloc.c	2.14.2.1.4
heap_destroy	/simnet/libsrc/libheap/create.c	2.14.2.2.3
heap_hang	/simnet/libsrc/libheap/alloc.c	2.14.2.1.14
heap_inconsistent_action	/simnet/libsrc/libheap/alloc.c	2.14.2.1.13
heap_print_statistics	/simnet/libsrc/libheap/alloc.c	2.14.2.1.10
heap_verify	/simnet/libsrc/libheap/alloc.c	2.14.2.1.12
helo_tick	/simnet/src/host/helo.c	2.6.4.1.1
host_local_vehicle_predicate	/simnet/src/host/iterator.c	2.9.3.1.9
host_loopback_function	/simnet/src/host/simnet.c	2.3.2.6
hull_to_world_from_direction	/simnet/src/host/missile.c	2.6.9.1.1
hull_to_world_from_orientation	/simnet/src/host/flyingveh.c	2.6.4.6.8
icon_from_object_type	/simnet/src/host/map.c	2.4.3.1.9
identify_exercise	/simnet/src/host/parser.c	2.5.2.1.18
identify_version	/simnet/src/host/version.c	2.1.3.2.1
idle_tick	/simnet/src/host/pilot.c	2.6.4.2.60
impact_weapon	/simnet/src/host/weapons.c	2.6.9.8.23
in_object2	/simnet/libsrc/libpvis/pv_bv.c	2.6.7.2.2
indirect_fire_to_sbx	/simnet/src/host/sbx.c	2.4.3.2.23
induce_roll	/simnet/src/host/flyingveh.c	2.6.4.6.11
induce_tail_spin	/simnet/src/host/flyingveh.c	2.6.4.6.10
inflate_rect	/simnet/libsrc/libutil/geometry.c	2.14.3.6.4
init_clocks	/simnet/libsrc/libutil/time.c	2.14.3.3.1
init_free_list	/simnet/libsrc/libheap/alloc.c	2.14.2.1.5
init_global_symbols	/simnet/src/host/symbols.c	2.1.2.1.1
init_grid_entry_list	/simnet/src/host/iterator.c	2.9.3.1.11
init_grid_tables	/simnet/src/host/iterator.c	2.9.3.1.10
init_mappings	/simnet/src/host/map.c	2.4.3.1.1
init_pilot_state_machine	/simnet/src/host/pilot.c	2.6.4.2.3
init_rect	/simnet/libsrc/libutil/geometry.c	2.14.3.6.3
init_safobj_table	/simnet/src/host/safobj.c	2.2.2.1
init_static_matrices	/simnet/src/host/flyingveh.c	2.6.4.6.9
init_stuff	/simnet/src/host/main.c	2.1.3.1.3
init_symbol_table	/simnet/libsrc/libreader/symbol.c	2.1.1.3.1
init_target_items	/simnet/src/host/targeting.c	2.6.9.3.13
init_target_list	/simnet/src/host/targeting.c	2.6.9.3.6
init_terrain_stuff	/simnet/src/host/main.c	2.1.3.1.5
initialize_heap	/simnet/libsrc/libheap/create.c	2.14.2.2.1
initialize_weapon_priority_list	/simnet/src/host/weapons.c	2.6.9.8.13
InitParser	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.2
insert_function	/simnet/libsrc/libsched/maint.c	2.2.1.3.5
insert_grid_entry	/simnet/src/host/iterator.c	2.9.3.1.15
insert_periodic_function	/simnet/libsrc/libsched/maint.c	2.2.1.3.4
InsertChar	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.23
int_dist	/simnet/src/host/formation.c	2.8.1.4.5
interesting_packet_test	/simnet/src/host/simnet.c	2.3.2.11
interior_angle_between_vectors	/simnet/libsrc/libutil/math.c	2.14.3.5.8
interpolate_curve	/simnet/libsrc/libdatabase/interpolate.c	2.6.8.1.1
intersect_rect	/simnet/libsrc/libutil/geometry.c	2.14.3.6.2
intersection_direction	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.2
intervis_can_see_pt_to_pt	/simnet/src/host/intervis.c	2.6.5.2.1
intervis_get_high_ground	/simnet/src/host/intervis.c	2.6.5.2.5
intervis_get_view	/simnet/src/host/intervis.c	2.6.5.2.2

intervis_possibly_visible	/simnet/src/host/intervis.c	2.6.5.2.3
intervis_pt_to_pt	/simnet/src/host/intervis.c	2.6.5.2.4
invoke	/simnet/libsrc/libsched/invoke.c	2.2.1.2.1
invoke_functions_until	/simnet/libsrc/libsched/sched.c	2.2.1.5.1
is_probably_a_string	/simnet/libsrc/libreader/tags.c	2.1.1.4.1
isolate_connection	/simnet/src/host/main.c	2.1.3.1.14
issue_reports	/simnet/src/host/reporter.c	2.8.2.1.43
iterator.h	/simnet/src/host/iterator.h	2.9.3.2
KillForward	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.18
kludge.c	/simnet/libsrc/libparser/kludge.c	2.5.1.10
lakes_thru	/simnet/libsrc/libquad/water_check.c	2.12.1.4.12
last_item	/simnet/libsrc/libquad/list.c	2.12.1.7.10
last_item_butlast	/simnet/libsrc/libquad/list.c	2.12.1.7.11
left_bottom_reagion	/simnet/libsrc/libquad/minimum_clip.c	2.12.1.13.4
left_column	/simnet/libsrc/libquad/minimum_clip.c	2.12.1.13.2
left_edge	/simnet/libsrc/libquad/minimum_clip.c	2.12.1.13.5
LeftWord	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.20
length	/simnet/libsrc/libquad/list.c	2.12.1.7.16
lexer.l	/simnet/libsrc/libreader/lexer.l	2.1.1.2
libcomm.h	/simnet/libsrc/libcomm/libcomm.h	2.14.4.3
libdatabase.h	/simnet/libsrc/libdatabase/libdatabase.h	2.6.8.9
libheap.h	/simnet/libsrc/libheap/libheap.h	2.14.2.3
libparser.h	/simnet/libsrc/libparser/libparser.h	2.5.1.11
libquad.h	/simnet/libsrc/libquad/libquad.h	2.12.1.18
libreader.h	/simnet/libsrc/libreader/libreader.h	2.1.1.5
libsched.h	/simnet/libsrc/libsched/libsched.h	2.2.1.6
libutil.h	/simnet/libsrc/libutil/libutil.h	2.14.3.9
line_cross_rect	/simnet/libsrc/libutil/geometry.c	2.14.3.6.1
line_intersection	/simnet/libsrc/libquad/intersection.c	2.12.1.14.8
line_intersection_core	/simnet/libsrc/libquad/intersection.c	2.12.1.14.10
loader_tick	/simnet/usr/host/loader.c	2.6.9.10.1
logistics.h	/simnet/src/host/logistics.h	2.6.6.2
logistics_remove_vehicles	/simnet/src/host/logistics.c	2.6.6.1.3
logistics_supply_offer_canceled	/simnet/src/host/logistics.c	2.6.6.1.8
logistics_supply_offer_received	/simnet/src/host/logistics.c	2.6.6.1.9
logistics_tick	/simnet/src/host/logistics.c	2.6.6.1.10
longpt.h	/simnet/include/saf/src/longpt.h	2.13.3.3
LookForward	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.28
lookup_cis_communication_string	/simnet/src/host/cis.c	2.10.1.1.4
lookup_cis_default_speed	/simnet/src/host/cis.c	2.10.1.1.3
lookup_cis_enabled_predicates	/simnet/src/host/cis.c	2.10.1.1.6
lookup_cis_information	/simnet/src/host/cis.c	2.10.1.1.7
lookup_cis_movement_method	/simnet/src/host/cis.c	2.10.1.1.5
lookup_cis_resumable	/simnet/src/host/cis.c	2.10.1.1.2
lookup_cis_situational	/simnet/src/host/cis.c	2.10.1.1.1
lookup_vehicle_with_range_check	/simnet/src/host/parser.c	2.5.2.1.19
main	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.17
main	/simnet/src/host/blaster.c	2.15.1.1
main	/simnet/src/host/main.c	2.1.3.1.19
major_detection_increase	/simnet/src/host/detection.c	2.6.5.1.13
major_spotter_increase	/simnet/src/host/spotter.c	2.8.2.3.10
make_array	/simnet/libsrc/libreader/parser.y	2.1.1.1.4
make_create_list	/simnet/src/host/create.c	2.11.1.1
make_leader	/simnet/src/host/formation.c	2.8.1.4.8
make_me_non_blocking	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.10
make_polygon	/simnet/libsrc/libutil/geometry.c	2.14.3.6.5
make_vector_2d	/simnet/libsrc/libquad/vector_2d.c	2.12.1.9.1

make_vector_from_angle_magnitude	/simnet/libsrc/libutil/math.c	2.14.3.5.3
map_echelon_number_to_echelon_symbol	/simnet/src/host/map.c	2.4.3.1.4
map_echelon_symbol_to_echelon_number	/simnet/src/host/map.c	2.4.3.1.3
map_echelon_type_number_to_echelon_type_symbol	/simnet/src/host/map.c	2.4.3.1.6
map_echelon_type_symbol_to_echelon_type_number	/simnet/src/host/map.c	2.4.3.1.5
map_echelon_type_symbol_to_icon_number	/simnet/src/host/map.c	2.4.3.1.7
map_icon_symbol_to_icon_number	/simnet/src/host/map.c	2.4.3.1.2
map_role_sym_to_role_number	/simnet/src/host/targeting.c	2.6.9.3.1
MarkCursor	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.8
mass_stdc.h	/simnet/include/saf/src/mass_stdc.h	2.13.3.4
master_rudp_close	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.6
master_rudp_open	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.4
master_rudp_synch	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.8
master_udp_open	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.11
member	/simnet/libsrc/libquad/list.c	2.12.1.7.6
messages.h	/simnet/src/host/messages.h	2.4.1.1
minimum_clip	/simnet/libsrc/libquad/minimum_clip.c	2.12.1.13.1
minmax.h	/simnet/include/saf/src/minmax.h	2.13.3.5
missile.h	/simnet/src/host/missile.h	2.6.9.2
missile_deactivate	/simnet/src/host/missile.c	2.6.9.1.8
missile_ground_impact	/simnet/src/host/missile.c	2.6.9.1.7
missile_maybe_hit_target	/simnet/src/host/missile.c	2.6.9.1.10
missile_send_appearance	/simnet/src/host/missile.c	2.6.9.1.13
missile_set_desired_direction	/simnet/src/host/missile.c	2.6.9.1.12
missile_show	/simnet/src/host/missile.c	2.6.9.1.5
missile_state_to_string	/simnet/src/host/missile.c	2.6.9.1.2
mission_stationpoint	/simnet/src/host/driver.c	2.6.3.2.9
muzzle_position_in_world_coordinates	/simnet/src/host/weapons.c	2.6.9.8.18
navigator_entered_area	/simnet/src/host/navigator.c	2.10.2.3.9
navigator_entered_zone	/simnet/src/host/navigator.c	2.10.2.3.11
navigator_generate_report	/simnet/src/host/navigator.c	2.10.2.3.19
navigator_left_area	/simnet/src/host/navigator.c	2.10.2.3.10
navigator_left_zone	/simnet/src/host/navigator.c	2.10.2.3.12
navigator_passed_line	/simnet/src/host/navigator.c	2.10.2.3.8
navigator_passed_point	/simnet/src/host/navigator.c	2.10.2.3.7
navigator_reset_overlay	/simnet/src/host/navigator.c	2.10.2.3.5
navigator_set_overlay	/simnet/src/host/navigator.c	2.10.2.3.4
navigator_show	/simnet/src/host/navigator.c	2.10.2.3.3
navigator_tick	/simnet/src/host/navigator.c	2.10.2.3.6
nconc	/simnet/libsrc/libquad/list.c	2.12.1.7.9
ncons	/simnet/libsrc/libquad/list.c	2.12.1.7.2
need_decluster	/simnet/src/host/reporter.c	2.8.2.1.23
network_silent	/simnet/src/host/main.c	2.1.3.1.11
next_road_point	/simnet/libsrc/libquad/roads.c	2.12.1.15.5
next_route_point	/simnet/src/host/driver.c	2.6.3.2.7
NextChar	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.7
NextCommand	/simnet/libsrc/libparser/par_hist.c	2.5.1.8.4
noa_damp	/simnet/src/host/misc.c	2.14.1.2.7
normalize_and_rotate	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.4
nthcdr	/simnet/libsrc/libquad/list.c	2.12.1.7.8
objects	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.12

obstacles	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.13
offset_point	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.5
offset_points	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.4
open_line_intersection	/simnet/libsrc/libquad/intersection.c	2.12.1.14.9
open_to_connections	/simnet/src/host/sbx.c	2.4.3.2.1
or_bitfield	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.8
orbit_velocity	/simnet/src/host/pilot.c	2.6.4.2.12
p2_bottom	/simnet/libsrc/libquad/minimum_clip.c	2.12.1.13.6
p_follow_form_flip	/simnet/src/host/p_follower.c	2.6.4.3.2
p_follower_am_i_flying_coord	/simnet/src/host/p_follower.c	2.6.4.3.1
p_follower_arrive_at_same_time	/simnet/src/host/p_follower.c	2.6.4.3.5
p_follower_comp_reset_route	/simnet/src/host/p_follower.c	2.6.4.3.4
p_follower_find_pnt_in_turn	/simnet/src/host/p_follower.c	2.6.4.3.24
p_follower_flip_in_turn	/simnet/src/host/p_follower.c	2.6.4.3.23
p_follower_fly_in_coord_pos	/simnet/src/host/p_follower.c	2.6.4.3.3
p_follower_flying_independent	/simnet/src/host/p_follower.c	2.6.4.3.12
p_follower_followroute_tick	/simnet/src/host/p_follower.c	2.6.4.3.13
p_follower_gen_coord_goal_pnt	/simnet/src/host/p_follower.c	2.6.4.3.9
p_follower_get_wrld_offset	/simnet/src/host/p_follower.c	2.6.4.3.7
p_follower_goto_pnt_coord	/simnet/src/host/p_follower.c	2.6.4.3.17
p_follower_hoverattack_tick	/simnet/src/host/p_follower.c	2.6.4.3.16
p_follower_hoverhold	/simnet/src/host/p_follower.c	2.6.4.3.18
p_follower_hoverhold_tick	/simnet/src/host/p_follower.c	2.6.4.3.21
p_follower_land	/simnet/src/host/p_follower.c	2.6.4.3.19
p_follower_landhold_tick	/simnet/src/host/p_follower.c	2.6.4.3.20
p_follower_leader_passed_point	/simnet/src/host/p_follower.c	2.6.4.3.14
p_follower_new_rel_rt_pnt	/simnet/src/host/p_follower.c	2.6.4.3.11
p_follower_passed_point	/simnet/src/host/p_follower.c	2.6.4.3.15
p_follower_pnt_in_hard_turn	/simnet/src/host/p_follower.c	2.6.4.3.27
p_follower_pnt_in_no_turn	/simnet/src/host/p_follower.c	2.6.4.3.25
p_follower_pnt_in_shallow_turn	/simnet/src/host/p_follower.c	2.6.4.3.26
p_follower_set_desired_vel	/simnet/src/host/p_follower.c	2.6.4.3.6
p_follower_set_follow	/simnet/src/host/p_follower.c	2.6.4.3.10
p_follower_stop_coord	/simnet/src/host/p_follower.c	2.6.4.3.8
ParseCommandDone	/simnet/libsrc/libparser/par_base.c	2.5.1.1.4
ParseConvertNumber	/simnet/libsrc/libparser/par_lib.c	2.5.1.6.7
ParseDoKeywordTable	/simnet/libsrc/libparser/par_base.c	2.5.1.1.5
ParseError	/simnet/libsrc/libparser/par_util.c	2.5.1.7.7
ParseEscapeComplete	/simnet/libsrc/libparser/par_util.c	2.5.1.7.12
ParseFindEndList	/simnet/libsrc/libparser/par_util.c	2.5.1.7.9
ParseGetConstant	/simnet/libsrc/libparser/par_const.c	2.5.1.2.1
ParseGetDecimal	/simnet/libsrc/libparser/par_lib.c	2.5.1.6.1
ParseGetFields	/simnet/libsrc/libparser/par_fields.c	2.5.1.5.1
ParseGetHex	/simnet/libsrc/libparser/par_lib.c	2.5.1.6.2
ParseGetOctal	/simnet/libsrc/libparser/par_lib.c	2.5.1.6.3
ParseGetString	/simnet/libsrc/libparser/par_lib.c	2.5.1.6.4
ParseGetToken	/simnet/libsrc/libparser/par_util.c	2.5.1.7.6
ParseInput	/simnet/libsrc/libparser/par_base.c	2.5.1.1.1
ParseInSet	/simnet/libsrc/libparser/par_util.c	2.5.1.7.5
ParseKeyword	/simnet/libsrc/libparser/par_base.c	2.5.1.1.3
ParseList	/simnet/libsrc/libparser/par_base.c	2.5.1.1.2
ParseMatch	/simnet/libsrc/libparser/par_util.c	2.5.1.7.10
ParseMessage	/simnet/libsrc/libparser/par_util.c	2.5.1.7.8
ParseMustFree	/simnet/libsrc/libparser/par_util.c	2.5.1.7.4
ParseOptional	/simnet/libsrc/libparser/par_lib.c	2.5.1.6.6
ParsePrint	/simnet/libsrc/libparser/par_unix.c	2.5.1.9.3
ParsePutArg	/simnet/libsrc/libparser/par_lib.c	2.5.1.6.5
parser_create	/simnet/src/host/parser.c	2.5.2.1.3

parser_create_vehicle	/simnet/src/host/parser.c	2.5.2.1.33
parser_global_reset	/simnet/src/host/parser.c	2.5.2.1.35
parser_heap_collect	/simnet/src/host/parser.c	2.5.2.1.39
parser_heap_statistics	/simnet/src/host/parser.c	2.5.2.1.36
parser_heap_verify	/simnet/src/host/parser.c	2.5.2.1.37
parser_init	/simnet/src/host/parser.c	2.5.2.1.1
parser_restore_term	/simnet/src/host/parser.c	2.5.2.1.2
parser_send_string	/simnet/src/host/parser.c	2.5.2.1.41
parser_set_targeting_parameters	/simnet/src/host/parser.c	2.5.2.1.40
ParseTableFind	/simnet/libsrc/libparser/par_util.c	2.5.1.7.11
patches	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.6
perf_monitor_off	/simnet/libsrc/libsched/perf.c	2.2.1.4.2
perf_monitor_on	/simnet/libsrc/libsched/perf.c	2.2.1.4.1
periodic_fncl	/simnet/libsrc/libsched/fncl.c	2.2.1.1.2
phantom.h	/simnet/include/saf/src/phantom.h	2.13.3.1
phase_two_collision_check	/simnet/src/host/collision.c	2.6.3.1.15
pilot_cancel_immediate	/simnet/src/host/pilot.c	2.6.4.2.71
pilot_change_altitude_im	/simnet/src/host/pilot.c	2.6.4.2.66
pilot_change_speed_im	/simnet/src/host/pilot.c	2.6.4.2.68
pilot_check_state	/simnet/src/host/pilot.c	2.6.4.2.63
pilot_executing_immediate_command	/simnet/src/host/pilot.c	2.6.4.2.72
pilot_face_direction	/simnet/src/host/pilot.c	2.6.4.2.58
pilot_face_direction_im	/simnet/src/host/pilot.c	2.6.4.2.74
pilot_follow_leader	/simnet/src/host/pilot.c	2.6.4.2.80
pilot_follow_leader_tick	/simnet/src/host/p_follower.c	2.6.4.3.22
pilot_follow_vehicle	/simnet/src/host/pilot.c	2.6.4.2.56
pilot_follow_vehicle_im	/simnet/src/host/pilot.c	2.6.4.2.73
pilot_followroute_tick	/simnet/src/host/pilot.c	2.6.4.2.45
pilot_followvehicle	/simnet/src/host/pilot.c	2.6.4.2.57
pilot_get_altitude	/simnet/src/host/pilot.c	2.6.4.2.65
pilot_get_asm	/simnet/src/host/pilot.c	2.6.4.2.84
pilot_get_speed	/simnet/src/host/pilot.c	2.6.4.2.67
pilot_goto_endpoint	/simnet/src/host/pilot.c	2.6.4.2.28
pilot_goto_point_im	/simnet/src/host/pilot.c	2.6.4.2.69
pilot_gotopoint	/simnet/src/host/pilot.c	2.6.4.2.27
pilot_hold	/simnet/src/host/pilot.c	2.6.4.2.29
pilot_hold_im	/simnet/src/host/pilot.c	2.6.4.2.76
pilot_hoverattack	/simnet/src/host/pilot.c	2.6.4.2.51
pilot_hoverattack_approach	/simnet/src/host/pilot.c	2.6.4.2.48
pilot_hoverattack_complete	/simnet/src/host/pilot.c	2.6.4.2.54
pilot_hoverattack_im	/simnet/src/host/pilot.c	2.6.4.2.75
pilot_hoverattack_tick	/simnet/src/host/pilot.c	2.6.4.2.55
pilot_hoverhold	/simnet/src/host/pilot.c	2.6.4.2.32
pilot_hoverhold_tick	/simnet/src/host/pilot.c	2.6.4.2.33
pilot_init_followroute	/simnet/src/host/pilot.c	2.6.4.2.44
pilot_init_hoverattack	/simnet/src/host/pilot.c	2.6.4.2.46
pilot_init_hoverhold	/simnet/src/host/pilot.c	2.6.4.2.30
pilot_init_land	/simnet/src/host/pilot.c	2.6.4.2.39
pilot_init_orbithold	/simnet/src/host/pilot.c	2.6.4.2.34
pilot_is_facing_direction	/simnet/src/host/pilot.c	2.6.4.2.16
pilot_land_im	/simnet/src/host/pilot.c	2.6.4.2.70
pilot_landed_tick	/simnet/src/host/pilot.c	2.6.4.2.43
pilot_landing	/simnet/src/host/pilot.c	2.6.4.2.41
pilot_mission_completed	/simnet/src/host/pilot.c	2.6.4.2.77
pilot_on_same_route	/simnet/src/host/pilot.c	2.6.4.2.15
pilot_orbit_tick	/simnet/src/host/pilot.c	2.6.4.2.37
pilot_orbithold	/simnet/src/host/pilot.c	2.6.4.2.36

pilot_point_at_target	/simnet/src/host/pilot.c	2.6.4.2.50
pilot_racetrackhold	/simnet/src/host/pilot.c	2.6.4.2.38
pilot_remove_vehicle	/simnet/src/host/pilot.c	2.6.4.2.5
pilot_set_leader_mis	/simnet/src/host/pilot.c	2.6.4.2.79
pilot_set_route_mis	/simnet/src/host/pilot.c	2.6.4.2.81
pilot_set_speed_mis	/simnet/src/host/pilot.c	2.6.4.2.83
pilot_show	/simnet/src/host/pilot.c	2.6.4.2.7
pilot_show_machine	/simnet/src/host/pilot.c	2.6.4.2.6
pilot_start_gotopoint	/simnet/src/host/pilot.c	2.6.4.2.26
pilot_start_hoverattack	/simnet/src/host/pilot.c	2.6.4.2.49
pilot_start_hoverattack_approach	/simnet/src/host/pilot.c	2.6.4.2.47
pilot_start_hoverattack_complete	/simnet/src/host/pilot.c	2.6.4.2.53
pilot_start_hoverattack_egres	/simnet/src/host/pilot.c	2.6.4.2.52
pilot_start_hoverhold	/simnet/src/host/pilot.c	2.6.4.2.31
pilot_start_land	/simnet/src/host/pilot.c	2.6.4.2.40
pilot_start_landed	/simnet/src/host/pilot.c	2.6.4.2.42
pilot_start_orbithold	/simnet/src/host/pilot.c	2.6.4.2.35
pilot_start_takingoff	/simnet/src/host/pilot.c	2.6.4.2.24
pilot_state_to_string	/simnet/src/host/pilot.c	2.6.4.2.1
pilot_stationpoint	/simnet/src/host/pilot.c	2.6.4.2.78
pilot_stop_mission	/simnet/src/host/pilot.c	2.6.4.2.82
pilot_takingoff	/simnet/src/host/pilot.c	2.6.4.2.25
pilot_tick	/simnet/src/host/pilot.c	2.6.4.2.64
pilot_tick2	/simnet/src/host/pilot.c	2.6.4.2.59
plane_tick	/simnet/src/host/plane.c	2.6.4.4.1
point_in_building	/simnet/libsrc/libquad/building_check.c	2.12.1.17.2
point_in_bv	/simnet/libsrc/libpvis/pv_bv.c	2.6.7.2.1
point_in_polygon	/simnet/libsrc/libutil/geometry.c	2.14.3.6.6
point_inside_polygon	/simnet/libsrc/libquad/intersection.c	2.12.1.14.3
point_line_intersection	/simnet/libsrc/libquad/intersection.c	2.12.1.14.6
point_segment_intersection	/simnet/libsrc/libquad/intersection.c	2.12.1.14.7
point_weapon_at_target	/simnet/src/host/gunner.c	2.6.9.5.3
poll_request_to_string	/simnet/src/host/sbx.c	2.4.3.2.6
pop_cis	/simnet/src/host/cis.c	2.10.1.1.12
pop_resumable_cis	/simnet/src/host/cis.c	2.10.1.1.13
possible_intersection	/simnet/libsrc/libquad/intersection.c	2.12.1.14.2
predicate_impact_hook	/simnet/src/host/predicates.c	2.10.1.3.7
predicates_show	/simnet/src/host/predicates.c	2.10.1.3.9
predicates.h	/simnet/src/host/predicates.h	2.10.1.4
PreviousCommand	/simnet/libsrc/libparser/par_hist.c	2.5.1.8.3
print_block	/simnet/libsrc/libheap/alloc.c	2.14.2.1.15
print_bridge	/simnet/libsrc/libquad/bridges.c	2.12.1.3.2
print_building	/simnet/libsrc/libquad/trees.c	2.12.1.16.6
print_ch_pt_info	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.23
print_commands_from_sbx	/simnet/src/host/main.c	2.1.3.1.8
print_heap_table	/simnet/libsrc/libheap/alloc.c	2.14.2.1.11
print_lake	/simnet/libsrc/libquad/bridges.c	2.12.1.3.4
print_lists	/simnet/src/host/formation.c	2.8.1.4.9
print_matrix	/simnet/libsrc/libutil/math.c	2.14.3.5.1
print_message_position	/simnet/src/host/sbx.c	2.4.3.2.77
print_opfor_header	/simnet/src/host/sbx.c	2.4.3.2.76
print_performance_stats	/simnet/libsrc/libsched/perf.c	2.2.1.4.6
print_quad_node	/simnet/libsrc/libquad/print.c	2.12.1.6.1
print_reasons_and_clear	/simnet/src/host/parser.c	2.5.2.1.38
print_river_intersection	/simnet/libsrc/libquad/rivers.c	2.12.1.2.4
print_river_segment	/simnet/libsrc/libquad/rivers.c	2.12.1.2.2
print_road_intersection	/simnet/libsrc/libquad/roads.c	2.12.1.15.4
print_road_segment	/simnet/libsrc/libquad/roads.c	2.12.1.15.2

print_route	/simnet/src/host/route.c	2.10.2.4.17
print_route_list	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.22
print_routepoint	/simnet/src/host/route.c	2.10.2.4.16
print_sockaddr	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.8
print_tree_canopy	/simnet/libsrc/libquad/trees.c	2.12.1.16.4
print_tree	/simnet/libsrc/libquad/trees.c	2.12.1.16.2
print_vector	/simnet/libsrc/libutil/math.c	2.14.3.5.2
print_vector_list	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.23
print_vehicle_ids	/simnet/src/host/sbx.c	2.4.3.2.78
print_vehicle_marking	/simnet/src/host/vehicle.c	2.9.2.2.3
printdirection	/simnet/src/host/driver.c	2.6.3.2.40
printimmediate	/simnet/src/host/driver.c	2.6.3.2.42
printmission	/simnet/src/host/driver.c	2.6.3.2.39
printroutedirection	/simnet/src/host/driver.c	2.6.3.2.41
proc_switches	/simnet/libsrc/libutil/args.c	2.14.3.1.1
project_point	/simnet/src/host/pilot.c	2.6.4.2.21
protocol_data_process	/simnet/src/host/simnet.c	2.3.2.14
protocol_sim_process	/simnet/src/host/simnet.c	2.3.2.12
protocol_stealth_process	/simnet/src/host/simnet.c	2.3.2.13
prototypes.h	/simnet/src/host/prototypes.h	2.13.2.1
prune_to_point	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.6
push	/simnet/libsrc/libquad/list.c	2.12.1.7.3
push_cis	/simnet/src/host/cis.c	2.10.1.1.11
push_vector_2d_on_points	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.17
push_vector_2d_on_route	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.18
pve.h	/simnet/libsrc/libpvis/pve.h	2.6.7.3
pve_checkvis	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.1
pvis_call.h	/simnet/libsrc/libpvis/pvis_call.h	2.6.7.4
queue_allocate	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.1
queue_dequeue	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.6
queue_describe	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.11
queue_distribute	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.25
queue_enqueue	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.5
queue_exercise	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.7
queue_flush	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.10
queue_flush_hangers	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.9
queue_length	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.2
queue_traverse_and_apply	/simnet/libsrc/libcomm/queue_fix.c	2.14.4.2.8
r2	/simnet/src/host/misc.c	2.14.1.2.6
radiate_target	/simnet/src/host/weapons.c	2.6.9.8.19
randomize_vector	/simnet/src/host/impact.c	2.6.4.5.3
range_squared	/simnet/libsrc/libutil/math.c	2.14.3.5.10
rassoc_sym	/simnet/src/host/map.c	2.4.3.1.8
re_assign_routepoints	/simnet/src/host/route.c	2.10.2.4.13
read_feature	/simnet/libsrc/libquad/terrain.c	2.12.1.1.4
read_machine_file	/simnet/src/host/main.c	2.1.3.1.4
read_quadtree	/simnet/libsrc/libquad/terrain.c	2.12.1.1.3
read_quadtree_database	/simnet/libsrc/libquad/terrain.c	2.12.1.1.1
read_terrain_files	/simnet/libsrc/libquad/terrain.c	2.12.1.1.2
reader_read_file	/simnet/libsrc/libreader/parser.y	2.1.1.1.7
readjust_overlay	/simnet/src/host/dispatch.c	2.14.1.1.9
RedisplayLine	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.5
rel_conn.h	/simnet/libsrc/libudp/rel_conn.h	2.4.2.4
relax_points	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.10
relax_points_aux	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.11
remote_change_stealth_controlling_port	/simnet/usr/host/remote.c	2.7.1.16
remote_deactivate	/simnet/usr/host/remote.c	2.7.1.15

remote_fill_in_appearance_data	/simnet/usr/host/remote.c	2.7.1.6
remote_fill_in_echelon_data	/simnet/usr/host/remote.c	2.7.1.7
remote_fill_in_position_data	/simnet/usr/host/remote.c	2.7.1.8
remote_go_away	/simnet/usr/host/remote.c	2.7.1.3
remote_init_vars	/simnet/usr/host/remote.c	2.7.1.2
remote_new_appearance	/simnet/usr/host/remote.c	2.7.1.12
remote_next_road_point	/simnet/usr/host/remote.c	2.7.1.9
remote_show	/simnet/usr/host/remote.c	2.7.1.4
remote_start_being_watched	/simnet/usr/host/remote.c	2.7.1.10
remote_start_ticking	/simnet/usr/host/remote.c	2.7.1.5
remote_stop_being_watched	/simnet/usr/host/remote.c	2.7.1.11
remote_tick	/simnet/usr/host/remote.c	2.7.1.13
remove_clustered_vehicle	/simnet/src/host/reporter.c	2.8.2.1.26
remove_dead_clustered_vehicle	/simnet/src/host/reporter.c	2.8.2.1.31
remove_duplicates	/simnet/libsrc/libquad/list.c	2.12.1.7.5
remove_executing_unit	/simnet/src/host/cm.c	2.10.2.1.22
remove_from_free_list	/simnet/libsrc/libheap/alloc.c	2.14.2.1.8
remove_function	/simnet/libsrc/libsched/maint.c	2.2.1.3.3
remove_grid_entry	/simnet/src/host/iterator.c	2.9.3.1.14
remove_removed_vehicle	/simnet/src/host/reporter.c	2.8.2.1.30
remove_shelling_cluster	/simnet/src/host/reporter.c	2.8.2.1.38
remove_vehicles	/simnet/src/host/dispatch.c	2.14.1.1.5
report_cluster_vehicle_type	/simnet/src/host/reporter.c	2.8.2.1.45
report_error_from_tdb_once	/simnet/src/host/misc.c	2.14.1.2.4
report_hit	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.22
report_ivis_network	/simnet/src/host/reporter.c	2.8.2.1.44
report_last_hit	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.25
report_tree_block	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.17
reporter.h	/simnet/src/host/reporter.h	2.8.2.2
reporter_remove_vehicles	/simnet/src/host/reporter.c	2.8.2.1.3
reporter_show	/simnet/src/host/reporter.c	2.8.2.1.4
reporter_tick	/simnet/src/host/reporter.c	2.8.2.1.41
reset_predicates	/simnet/src/host/predicates.c	2.10.1.3.10
RestoreCursor	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.9
resuply_check_ok	/simnet/src/host/logistics.c	2.6.6.1.5
resupply_bld_ammo_needs	/simnet/src/host/logistics.c	2.6.6.1.6
reverse	/simnet/libsrc/libquad/list.c	2.12.1.7.15
RightWord	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.21
rmat_to_fmat	/simnet/libsrc/libutil/math.c	2.14.3.5.15
round_real_to_int	/simnet/libsrc/libutil/math.c	2.14.3.5.11
route.h	/simnet/src/host/route.h	2.10.2.5
routepoint_distance	/simnet/src/host/route.c	2.10.2.4.1
rt_point_to_vector	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.21
rudp_ack_received	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.15
rudp_close	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.5
rudp_discard_all_buffers	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.10
rudp_disconnect	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.9
rudp_init	/simnet/libsrc/libudp/buf_conn.c	2.4.2.5.1
rudp_init	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.1
rudp_open	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.3
rudp_post_message	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.12
rudp_read_message	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.11
rudp_retransmit	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.17
rudp_retransmit_buf	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.16
rudp_send	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.13
rudp_send_bare_ack	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.18
rudp_show	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.2
rudp_synch	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.7

rudp_tick	/simnet/libsrc/libudp/rel_conn.c	2.4.2.3.19
rvec_to_fvec	/simnet/libsrc/libutil/math.c	2.14.3.5.14
saf_complete_reset	/simnet/src/host/saf.c	2.2.4.1
saf_exit	/simnet/src/host/main.c	2.1.3.1.22
SAF_NET_SND_KLUDGE	/simnet/src/host/simnet.c	2.3.2.5
saf_print_help	/simnet/src/host/main.c	2.1.3.1.17
saf_remove_vehicle	/simnet/src/host/saf.c	2.2.4.3
saf_reset_for_workstation	/simnet/src/host/saf.c	2.2.4.2
saf_vehicle_cancel_overlay	/simnet/src/host/saf_vehicle.c	2.6.1.1.47
saf_vehicle_catastrophic_kill	/simnet/src/host/saf_vehicle.c	2.6.1.1.25
saf_vehicle_change_speed	/simnet/src/host/saf_vehicle.c	2.6.1.1.53
saf_vehicle_checkpoint_state	/simnet/src/host/saf_vehicle.c	2.6.1.1.16
saf_vehicle_collision_over	/simnet/src/host/saf_vehicle.c	2.6.1.1.42
saf_vehicle_doing_collision_stuff	/simnet/src/host/saf_vehicle.c	2.6.1.1.43
saf_vehicle_drain_supplies	/simnet/src/host/saf_vehicle.c	2.6.1.1.26
saf_vehicle_est_position	/simnet/src/host/saf_vehicle.c	2.6.1.1.58
saf_vehicle_execute_overlay	/simnet/src/host/saf_vehicle.c	2.6.1.1.46
saf_vehicle_face_direction	/simnet/src/host/saf_vehicle.c	2.6.1.1.57
saf_vehicle_fake_resupply	/simnet/src/host/saf_vehicle.c	2.6.1.1.33
saf_vehicle_fill_in_appearance_data	/simnet/src/host/saf_vehicle.c	2.6.1.1.13
saf_vehicle_fill_in_echelon_data	/simnet/src/host/saf_vehicle.c	2.6.1.1.14
saf_vehicle_fill_in_position_data	/simnet/src/host/saf_vehicle.c	2.6.1.1.15
saf_vehicle_firepower_kill	/simnet/src/host/saf_vehicle.c	2.6.1.1.22
saf_vehicle_follow_vehicle	/simnet/src/host/saf_vehicle.c	2.6.1.1.54
saf_vehicle_generate_status_report	/simnet/src/host/saf_vehicle.c	2.6.1.1.34
saf_vehicle_go_away	/simnet/src/host/saf_vehicle.c	2.6.1.1.2
saf_vehicle_goto_point	/simnet/src/host/saf_vehicle.c	2.6.1.1.55
saf_vehicle_halt	/simnet/src/host/saf_vehicle.c	2.6.1.1.52
saf_vehicle_indirect_fire	/simnet/src/host/saf_vehicle.c	2.6.1.1.38
saf_vehicle_mimic_vehicle	/simnet/src/host/saf_vehicle.c	2.6.1.1.18
saf_vehicle_mission_completed	/simnet/src/host/saf_vehicle.c	2.6.1.1.40
saf_vehicle_mobility_kill	/simnet/src/host/saf_vehicle.c	2.6.1.1.23
saf_vehicle_next_event_id	/simnet/src/host/saf_vehicle.c	2.6.1.1.28
saf_vehicle_out_of_ammo	/simnet/src/host/saf_vehicle.c	2.6.1.1.32
saf_vehicle_out_of_gas	/simnet/src/host/saf_vehicle.c	2.6.1.1.31
saf_vehicle_pro_sim	/simnet/src/host/saf_vehicle.c	2.6.1.1.39
saf_vehicle_reinit	/simnet/src/host/saf_vehicle.c	2.6.1.1.17
saf_vehicle_rejoin_unit	/simnet/src/host/saf_vehicle.c	2.6.1.1.45
saf_vehicle_remove_vehicles	/simnet/src/host/saf_vehicle.c	2.6.1.1.35
saf_vehicle_reset_station_keeper	/simnet/src/host/saf_vehicle.c	2.6.1.1.51
saf_vehicle_resume_mission	/simnet/src/host/saf_vehicle.c	2.6.1.1.56
saf_vehicle_send_appearance	/simnet/src/host/saf_vehicle.c	2.6.1.1.21
saf_vehicle_set_direction	/simnet/src/host/saf_vehicle.c	2.6.1.1.50
saf_vehicle_set_leader	/simnet/src/host/saf_vehicle.c	2.6.1.1.5
saf_vehicle_set_marking	/simnet/src/host/saf_vehicle.c	2.6.1.1.9
saf_vehicle_set_route	/simnet/src/host/saf_vehicle.c	2.6.1.1.48
saf_vehicle_set_speed	/simnet/src/host/saf_vehicle.c	2.6.1.1.49
saf_vehicle_set_superior	/simnet/src/host/saf_vehicle.c	2.6.1.1.4
saf_vehicle_set_targeting_parameters	/simnet/src/host/saf_vehicle.c	2.6.1.1.12
saf_vehicle_show	/simnet/src/host/saf_vehicle.c	2.6.1.1.11
saf_vehicle_simulator_in_command	/simnet/src/host/saf_vehicle.c	2.6.1.1.44
saf_vehicle_start_ticking	/simnet/src/host/saf_vehicle.c	2.6.1.1.3
saf_vehicle_stop_flaming	/simnet/src/host/saf_vehicle.c	2.6.1.1.24
saf_vehicle_stop_mimic	/simnet/src/host/saf_vehicle.c	2.6.1.1.19
saf_vehicle_sudden_stop	/simnet/src/host/saf_vehicle.c	2.6.1.1.27
saf_vehicle_supplies_needed	/simnet/src/host/saf_vehicle.c	2.6.1.1.29
saf_vehicle_supplies_provided	/simnet/src/host/saf_vehicle.c	2.6.1.1.30

saf_vehicle_teleport	/simnet/src/host/saf_vehicle.c	2.6.1.1.7
saf_vehicle_teleport_to_station	/simnet/src/host/saf_vehicle.c	2.6.1.1.6
saf_vehicle_tick	/simnet/src/host/saf_vehicle.c	2.6.1.1.20
saf_vehicle_useless	/simnet/src/host/saf_vehicle.c	2.6.1.1.41
saf_vehicle_vehicle_impact	/simnet/src/host/saf_vehicle.c	2.6.1.1.36
saf_vehicle_vehicle_rammed	/simnet/src/host/saf_vehicle.c	2.6.1.1.37
safobj.h	/simnet/src/host/safobj.h	2.9.1.1
sbx.h	/simnet/src/host/sbx.h	2.4.3.3
sbx_add_area	/simnet/src/host/sbx.c	2.4.3.2.46
sbx_add_line	/simnet/src/host/sbx.c	2.4.3.2.48
sbx_add_point	/simnet/src/host/sbx.c	2.4.3.2.45
sbx_add_route	/simnet/src/host/sbx.c	2.4.3.2.49
sbx_add_zone	/simnet/src/host/sbx.c	2.4.3.2.47
sbx_altitude	/simnet/src/host/sbx.c	2.4.3.2.65
sbx_attach_stealth	/simnet/src/host/sbx.c	2.4.3.2.70
sbx_attack	/simnet/src/host/sbx.c	2.4.3.2.69
sbx_change_formation	/simnet/src/host/sbx.c	2.4.3.2.54
sbx_change_speed	/simnet/src/host/sbx.c	2.4.3.2.53
sbx_connection_create_unit_msg	/simnet/src/host/sbx.c	2.4.3.2.29
sbx_connection_disconnect	/simnet/src/host/sbx.c	2.4.3.2.14
sbx_connection_exit	/simnet/src/host/sbx.c	2.4.3.2.11
sbx_connection_generate_arty_msg	/simnet/src/host/sbx.c	2.4.3.2.37
sbx_connection_generate_reset_vehicle	/simnet/src/host/sbx.c	2.4.3.2.31
sbx_connection_init	/simnet/src/host/sbx.c	2.4.3.2.9
sbx_connection_open	/simnet/src/host/sbx.c	2.4.3.2.10
sbx_connection_overloaded	/simnet/src/host/sbx.c	2.4.3.2.16
sbx_connection_process_ground_impact	/simnet/src/host/sbx.c	2.4.3.2.21
sbx_connection_process_indirect_fire	/simnet/src/host/sbx.c	2.4.3.2.22
sbx_connection_process_message	/simnet/src/host/sbx.c	2.4.3.2.26
sbx_connection_process_messages	/simnet/src/host/sbx.c	2.4.3.2.27
sbx_connection_process_vehicle_impact	/simnet/src/host/sbx.c	2.4.3.2.20
sbx_connection_query_sub_state	/simnet/src/host/sbx.c	2.4.3.2.38
sbx_connection_reset_msg	/simnet/src/host/sbx.c	2.4.3.2.30
sbx_connection_send_to_port	/simnet/src/host/sbx.c	2.4.3.2.19
sbx_connection_service_poll_msg	/simnet/src/host/sbx.c	2.4.3.2.33
sbx_connection_set_ivis_parameters	/simnet/src/host/sbx.c	2.4.3.2.40
sbx_connection_set_ivis_xmit_mode	/simnet/src/host/sbx.c	2.4.3.2.39
sbx_connection_show	/simnet/src/host/sbx.c	2.4.3.2.12
sbx_connection_stat_pulse	/simnet/src/host/sbx.c	2.4.3.2.35
sbx_connection_synch_received	/simnet/src/host/sbx.c	2.4.3.2.13
sbx_connection_tick	/simnet/src/host/sbx.c	2.4.3.2.17
sbx_connection_vehicle_reinit	/simnet/src/host/sbx.c	2.4.3.2.32
sbx_connection_wat_pulse	/simnet/src/host/sbx.c	2.4.3.2.34
sbx_connection_write_buffer	/simnet/src/host/sbx.c	2.4.3.2.15
sbx_delete_cm	/simnet/src/host/sbx.c	2.4.3.2.55
sbx_delete_overlay	/simnet/src/host/sbx.c	2.4.3.2.50
sbx_execute_overlay	/simnet/src/host/sbx.c	2.4.3.2.51
sbx_face_direction	/simnet/src/host/sbx.c	2.4.3.2.61
sbx_follow_vehicle	/simnet/src/host/sbx.c	2.4.3.2.57
sbx_goto_point	/simnet/src/host/sbx.c	2.4.3.2.59

sbx_halt	/simnet/src/host/sbx.c	2.4.3.2.52
sbx_hold	/simnet/src/host/sbx.c	2.4.3.2.67
sbx_land	/simnet/src/host/sbx.c	2.4.3.2.63
sbx_printf	/simnet/src/host/sbx.c	2.4.3.2.8
sbx_rejoin_unit	/simnet/src/host/sbx.c	2.4.3.2.56
sbx_resume_mission	/simnet/src/host/sbx.c	2.4.3.2.60
sbx_resupply	/simnet/src/host/sbx.c	2.4.3.2.71
sbx_set_all_known_vehicles	/simnet/src/host/sbx.c	2.4.3.2.42
sbx_set_specific_known_vehicles	/simnet/src/host/sbx.c	2.4.3.2.43
sbx_set_targeting_parameters	/simnet/src/host/sbx.c	2.4.3.2.62
sbx_set_top_level_known_vehicles	/simnet/src/host/sbx.c	2.4.3.2.44
sbx_simulator_in_command	/simnet/src/host/sbx.c	2.4.3.2.58
sbx_swap_known_vehicles	/simnet/src/host/sbx.c	2.4.3.2.41
scan_weapon	/simnet/src/host/gunner.c	2.6.9.5.4
ScanBackwards	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.29
ScanForward	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.27
scheduler_init	/simnet/libsrc/libsched/maint.c	2.2.1.3.1
seg_intersection	/simnet/libsrc/libquad/intersection.c	2.12.1.14.1
segment_inside_polygon	/simnet/libsrc/libquad/intersection.c	2.12.1.14.5
segment_through_lake	/simnet/libsrc/libquad/water_check.c	2.12.1.4.7
segment_through_river	/simnet/libsrc/libquad/water_check.c	2.12.1.4.6
segment_through_water	/simnet/libsrc/libquad/water_check.c	2.12.1.4.5
select_weapon_priority_list	/simnet/src/host/targeting.c	2.6.9.3.18
send_contact_report	/simnet/src/host/reporter.c	2.8.2.1.46
send_shelling_report	/simnet/src/host/reporter.c	2.8.2.1.48
send_spot_report	/simnet/src/host/reporter.c	2.8.2.1.47
send_stealth_gone_msg	/simnet/usr/host/remote.c	2.7.1.14
set_abort_on_error	/simnet/src/host/parser.c	2.5.2.1.5
set_bit	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.3
set_bitfield	/simnet/libsrc/libutil/bitfield.c	2.14.3.4.7
set_command_printing	/simnet/src/host/parser.c	2.5.2.1.4
set_critical_performance_level	/simnet/libsrc/libsched/perf.c	2.2.1.4.3
set_exercise_id	/simnet/src/host/main.c	2.1.3.1.13
set_ground_impact_mode	/simnet/src/host/parser.c	2.5.2.1.6
set_header_printing	/simnet/src/host/parser.c	2.5.2.1.8
set_indirect_fire_mode	/simnet/src/host/parser.c	2.5.2.1.7
set_monitor_period	/simnet/src/host/parser.c	2.5.2.1.9
set_number_of_fake_remotes	/simnet/src/host/main.c	2.1.3.1.16
set_pv_parms	/simnet/src/host/intervis.c	2.6.5.2.6
set_simnet_files	/simnet/src/host/main.c	2.1.3.1.12
set_soil_type	/simnet/src/host/misc.c	2.14.1.2.1
set_speed_dir	/simnet/src/host/driver.c	2.6.3.2.6
set_symbol_value	/simnet/libsrc/libreader/symbol.c	2.1.1.3.7
set_target_list	/simnet/src/host/targeting.c	2.6.9.3.7
set_target_machine	/simnet/src/host/main.c	2.1.3.1.15
set_targeting_parameters	/simnet/src/host/dispatch.c	2.14.1.1.7
set_tdb_to_cache	/simnet/src/host/main.c	2.1.3.1.10
set_terrain_dbase	/simnet/src/host/main.c	2.1.3.1.9
set_xor	/simnet/libsrc/libquad/list.c	2.12.1.7.13
SetStopPoint	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.6
shell_report_needed	/simnet/src/host/reporter.c	2.8.2.1.40
shell_type_to_string	/simnet/src/host/reporter.c	2.8.2.1.33
shoot_target	/simnet/src/host/gunner.c	2.6.9.5.5
show	/simnet/src/host/dispatch.c	2.14.1.1.4
show_applies_to	/simnet/src/host/cm.c	2.10.2.1.43
show_area_cm	/simnet/src/host/cm.c	2.10.2.1.40
show_connection	/simnet/src/host/parser.c	2.5.2.1.16
show_connection_all_sbx	/simnet/src/host/sbx.c	2.4.3.2.4

show_control_measure	/simnet/src/host/cm.c	2.10.2.1.37
show_control_measures	/simnet/src/host/cm.c	2.10.2.1.36
show_curves	/simnet/libsrc/libdatabase/database.c	2.6.8.8.3
show_detection_database	/simnet/libsrc/libdatabase/detection.c	2.6.8.2.1
show_df_damage_database	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.1
show_df_damage_entry	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.6
show_df_damage_object	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.4
show_df_damage_side	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.5
show_df_damage_veh	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.2
show_df_damage_weapon	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.3
show_hitmodel_database	/simnet/libsrc/libdatabase/hitmodel.c	2.6.8.3.1
show_if_damage_database	/simnet/libsrc/libdatabase/if_damage.c	2.6.8.7.1
show_if_damage_entry	/simnet/libsrc/libdatabase/if_damage.c	2.6.8.7.5
show_if_damage_veh	/simnet/libsrc/libdatabase/if_damage.c	2.6.8.7.2
show_if_damage_weapon	/simnet/libsrc/libdatabase/if_damage.c	2.6.8.7.3
show_if_ranged_damage_entry	/simnet/libsrc/libdatabase/if_damage.c	2.6.8.7.4
show_line_cm	/simnet/src/host/cm.c	2.10.2.1.42
show_overlay	/simnet/src/host/cm.c	2.10.2.1.35
show_overlays	/simnet/src/host/cm.c	2.10.2.1.34
show_overlays_all_sbx	/simnet/src/host/sbx.c	2.4.3.2.7
show_point_cm	/simnet/src/host/cm.c	2.10.2.1.39
show_point_list	/simnet/src/host/cm.c	2.10.2.1.38
show_removed_vehicles	/simnet/src/host/reporter.c	2.8.2.1.28
show_route	/simnet/src/host/route.c	2.10.2.4.14
show_route_points	/simnet/src/host/route.c	2.10.2.4.15
show_sbx_overlays	/simnet/src/host/parser.c	2.5.2.1.17
show_shelling_clusters	/simnet/src/host/reporter.c	2.8.2.1.35
show_table_record	/simnet/libsrc/libdatabase/database.c	2.6.8.8.2
show_vehicle	/simnet/src/host/parser.c	2.5.2.1.20
show_vehicle_cluster	/simnet/src/host/reporter.c	2.8.2.1.8
show_vehicle_clusters	/simnet/src/host/reporter.c	2.8.2.1.7
show_vehicle_id_list	/simnet/src/host/reporter.c	2.8.2.1.9
show_vehicle_ids	/simnet/src/host/parser.c	2.5.2.1.11
show_zone_cm	/simnet/src/host/cm.c	2.10.2.1.41
sigh	/simnet/src/host/main.c	2.1.3.1.2
sim_undef.h	/simnet/include/saf/src/sim_undef.h	2.13.3.6
simnet_exit	/simnet/src/host/simnet.c	2.3.2.7
simnet_getstats	/simnet/src/host/simnet.c	2.3.2.8
simnet_tick	/simnet/src/host/simnet.c	2.3.2.2
simnet_zerostats	/simnet/src/host/simnet.c	2.3.2.9
simple_queue_allocate	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.7
simple_queue_deallocate	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.8
simple_queue_dequeue	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.2
simple_queue_describe	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.9
simple_queue_enqueue	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.1
simple_queue_is_empty	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.5
simple_queue_is_full	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.4
simple_queue_is_this_power_of_two	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.6
simple_queue_length	/simnet/libsrc/libcomm/bufpool.c	2.14.4.1.3
skirt_lake	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.7
skirt_river_bend	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.9
skirt_water	/simnet/libsrc/libquad/skirt_water.c	2.12.1.12.1
sockaddr_init	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.9
sort_damage_table	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.9
sort_form_db	/simnet/src/host/formation.c	2.8.1.4.12
sort_tag_table	/simnet/libsrc/libreader/tags.c	2.1.1.4.5
sort_water	/simnet/libsrc/libquad/water_check.c	2.12.1.4.10

splice_out_vehicle_cluster	/simnet/src/host/reporter.c	2.8.2.1.21
spotter.h	/simnet/src/host/spotter.h	2.8.2.4
spotter_init_table	/simnet/src/host/spotter.c	2.8.2.3.5
spotter_remove_vehicle	/simnet/src/host/spotter.c	2.8.2.3.6
spotter_save_weight	/simnet/src/host/spotter.c	2.8.2.3.11
spotter_show	/simnet/src/host/spotter.c	2.8.2.3.12
spotter_show_table	/simnet/src/host/spotter.c	2.8.2.3.13
spotter_sum_inferior_composite_tables	/simnet/src/host/spotter.c	2.8.2.3.8
spotter_tick	/simnet/src/host/spotter.c	2.8.2.3.7
spread_increment	/simnet/src/host/formation.c	2.8.1.4.3
stack_push	/simnet/libsrc/libreader/parser.y	2.1.1.1.1
stack_push_array	/simnet/libsrc/libreader/parser.y	2.1.1.1.2
start_avoiding_collision	/simnet/src/host/collision.c	2.6.3.1.9
start_avoiding_house	/simnet/src/host/collision.c	2.6.3.1.11
start_disengaging	/simnet/src/host/collision.c	2.6.3.1.14
start_resupply_of_to	/simnet/src/host/logistics.c	2.6.6.1.4
start_simnet	/simnet/src/host/simnet.c	2.3.2.1
start_ticking	/simnet/src/host/tickable.c	2.2.3.3
startup	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.2
stationpoint	/simnet/src/host/driver.c	2.6.3.2.10
stealth_attach_to	/simnet/src/host/parser.c	2.5.2.1.31
stealth_set_mimic_off	/simnet/src/host/parser.c	2.5.2.1.29
stealth_set_mimic_on	/simnet/src/host/parser.c	2.5.2.1.28
stealth_set_symbols_draw_tick	/simnet/src/host/parser.c	2.5.2.1.27
stealth_site_host_pair	/simnet/src/host/parser.c	2.5.2.1.30
stealth_teleport_to	/simnet/src/host/parser.c	2.5.2.1.32
stop	/simnet/src/host/driver.c	2.6.3.2.5
stop_ticking	/simnet/src/host/tickable.c	2.2.3.4
store_hit	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.19
string_for_altitude_type	/simnet/src/host/sbx.c	2.4.3.2.64
string_for_attack_type	/simnet/src/host/sbx.c	2.4.3.2.68
string_for_hold_type	/simnet/src/host/sbx.c	2.4.3.2.66
stringcopy	/simnet/libsrc/libparser/par_copy.c	2.5.1.3.1
symbols.h	/simnet/src/host/symbols.h	2.1.2.2
tactical_state.h	/simnet/src/host/tactical_state.h	2.8.3.1
tag_error	/simnet/libsrc/libreader/tags.c	2.1.1.4.2
target_in_position	/simnet/src/host/targeting.c	2.6.9.3.17
TARGET_ITEM	/simnet/src/host/targeting.c	2.6.9.3.12
target_priority	/simnet/src/host/targeting.c	2.6.9.3.15
target_type_ok	/simnet/src/host/targeting.c	2.6.9.3.16
targeting.h	/simnet/src/host/targeting.h	2.6.9.4
targeting_set_fire_at_pointair	/simnet/src/host/targeting.c	2.6.9.3.8
targeting_set_fire_at_will	/simnet/src/host/targeting.c	2.6.9.3.10
targeting_set_hold_fire	/simnet/src/host/targeting.c	2.6.9.3.9
targeting_set_parameters	/simnet/src/host/targeting.c	2.6.9.3.11
targeting_show	/simnet/src/host/targeting.c	2.6.9.3.5
targeting_tick	/simnet/src/host/targeting.c	2.6.9.3.21
tdb_get_gl	/simnet/src/host/misc.c	2.14.1.2.2
tdb_get_zl	/simnet/src/host/misc.c	2.14.1.2.3
terrain	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.7
test_clutter	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.26
test_maxima	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.5
test_mins	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.4
tickable_note_start_tick	/simnet/src/host/tickable.c	2.2.3.6
toggle_debugging	/simnet/src/host/parser.c	2.5.2.1.10
tolerance.h	/simnet/include/saf/src/tolerance.h	2.13.3.7
top_left_corner	/simnet/libsrc/libquad/minimum_clip.c	2.12.1.13.3

track_target	/simnet/src/host/gunner.c	2.6.9.5.2
treelines	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.16
trees	/simnet/libsrc/libpvis/pve_checkvis.c	2.6.7.1.15
turret.h	/simnet/src/host/turret.h	2.6.9.7
turret_firepower_kill	/simnet/src/host/turret.c	2.6.9.6.10
turret_interest_dir	/simnet/src/host/turret.c	2.6.9.6.9
turret_muzzle_position_in_world_coordinates	/simnet/src/host/turret.c	2.6.9.6.7
turret_point_at_target	/simnet/src/host/turret.c	2.6.9.6.5
turret_scan	/simnet/src/host/turret.c	2.6.9.6.6
turret_set_scan_parms	/simnet/src/host/turret.c	2.6.9.6.8
turret_show	/simnet/src/host/turret.c	2.6.9.6.3
turret_slew	/simnet/src/host/turret.c	2.6.9.6.4
TypeFields	/simnet/libsrc/libparser/par_fields.c	2.5.1.5.2
uc	/simnet/libsrc/libparser/par_util.c	2.5.1.7.13
udp_close	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.2
udp_conn.h	/simnet/libsrc/libudp/udp_conn.h	2.4.2.2
udp_open	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.1
udp_read	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.4
udp_show	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.3
udp_write	/simnet/libsrc/libudp/udp_berkeley.c	2.4.2.1.5
unit_send_pae_data	/simnet/src/host/composite.c	2.8.1.3.30
upcase	/simnet/libsrc/libutil/cstring.c	2.14.3.8.1
update_cluster_com	/simnet/src/host/reporter.c	2.8.2.1.22
update_cluster_heading	/simnet/src/host/reporter.c	2.8.2.1.6
update_clusters	/simnet/src/host/reporter.c	2.8.2.1.32
update_grid_entry_list	/simnet/src/host/iterator.c	2.9.3.1.12
update_reporter_vehicles	/simnet/src/host/reporter.c	2.8.2.1.42
update_target_list	/simnet/src/host/targeting.c	2.6.9.3.19
UpdateLine	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.3
UpdateMoveCursor	/simnet/libsrc/libparser/par_edit.c	2.5.1.4.4
valid_quad	/simnet/libsrc/libquad/search.c	2.12.1.5.4
vec2_add	/simnet/libsrc/libutil/math.c	2.14.3.5.22
vec2_copy	/simnet/libsrc/libutil/math.c	2.14.3.5.29
vec2_cross	/simnet/libsrc/libutil/math.c	2.14.3.5.25
vec2_dot	/simnet/libsrc/libutil/math.c	2.14.3.5.24
vec2_init	/simnet/libsrc/libutil/math.c	2.14.3.5.20
vec2_mag	/simnet/libsrc/libutil/math.c	2.14.3.5.26
vec2_mag2	/simnet/libsrc/libutil/math.c	2.14.3.5.27
vec2_norm	/simnet/libsrc/libutil/math.c	2.14.3.5.31
vec2_print	/simnet/libsrc/libutil/math.c	2.14.3.5.19
vec2_range_squared	/simnet/libsrc/libutil/math.c	2.14.3.5.30
vec2_rot90	/simnet/libsrc/libutil/math.c	2.14.3.5.32
vec2_rot90minus	/simnet/libsrc/libutil/math.c	2.14.3.5.33
vec2_rotate	/simnet/src/host/pilot.c	2.6.4.2.22
vec2_scale	/simnet/libsrc/libutil/math.c	2.14.3.5.28
vec2_set	/simnet/libsrc/libutil/math.c	2.14.3.5.21
vec2_sub	/simnet/libsrc/libutil/math.c	2.14.3.5.23
vec2_veh2world	/simnet/src/host/driver.c	2.6.3.2.37
vec2_world2veh	/simnet/src/host/driver.c	2.6.3.2.38
vec_set	/simnet/src/host/driver.c	2.6.3.2.36
vector_distance_2d	/simnet/libsrc/libquad/vector_2d.c	2.12.1.9.4
vector_is_first	/simnet/libsrc/libquad/water_utilities.c	2.12.1.11.6
vector_normalize_2d	/simnet/libsrc/libquad/vector_2d.c	2.12.1.9.2
vector_rotate_2d	/simnet/libsrc/libquad/vector_2d.c	2.12.1.9.3
vector_z_rotate	/simnet/libsrc/libutil/math.c	2.14.3.5.12
veh_storage.h	/simnet/src/host/veh_storage.h	2.9.1.2
vehicle_bong	/simnet/src/host/pro_sim.c	2.3.1.5

vehicle_bong_do	/simnet/src/host/parser.c	2.5.2.1.26
vehicle_catastrophe	/simnet/src/host/parser.c	2.5.2.1.21
vehicle_component_name	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.8
vehicle_deactivate	/simnet/src/host/vehicle.c	2.9.2.2.8
vehicle_defuel	/simnet/src/host/parser.c	2.5.2.1.24
vehicle_facing_point	/simnet/src/host/pilot.c	2.6.4.2.17
vehicle_fake_resupply	/simnet/src/host/parser.c	2.5.2.1.23
vehicle_fill_in_appearance_data	/simnet/src/host/vehicle.c	2.9.2.2.6
vehicle_fill_in_position_data	/simnet/src/host/vehicle.c	2.9.2.2.5
vehicle_impact_to_sbx	/simnet/src/host/sbx.c	2.4.3.2.25
vehicle_iterator_next	/simnet/src/host/iterator.c	2.9.3.1.2
vehicle_iterator_once_next	/simnet/src/host/iterator.c	2.9.3.1.3
vehicle_iterator_reset	/simnet/src/host/iterator.c	2.9.3.1.1
vehicle_kill	/simnet/src/host/pro_sim.c	2.3.1.2
vehicle_kill_remote	/simnet/src/host/pro_sim.c	2.3.1.3
vehicle_list_append	/simnet/src/host/reporter.c	2.8.2.1.18
vehicle_manager_count	/simnet/src/host/iterator.c	2.9.3.1.5
vehicle_manager_count_force	/simnet/src/host/iterator.c	2.9.3.1.6
vehicle_manager_count_hosts	/simnet/src/host/iterator.c	2.9.3.1.8
vehicle_manager_count_sites	/simnet/src/host/iterator.c	2.9.3.1.7
vehicle_manager_print	/simnet/src/host/iterator.c	2.9.3.1.4
vehicle_ping	/simnet/src/host/pro_sim.c	2.3.1.6
vehicle_ping_do	/simnet/src/host/parser.c	2.5.2.1.25
vehicle_planar_distance_squared	/simnet/src/host/vehicle.c	2.9.2.2.7
vehicle_reappeared	/simnet/src/host/reporter.c	2.8.2.1.27
vehicle_resupply	/simnet/src/host/parser.c	2.5.2.1.22
vehicle_show	/simnet/src/host/vehicle.c	2.9.2.2.4
vehicle_status_string	/simnet/src/host/saf_vehicle.c	2.6.1.1.10
vehicle_wall_name	/simnet/libsrc/libdatabase/df_damage.c	2.6.8.6.7
vel2point	/simnet/src/host/driver.c	2.6.3.2.8
wait_until	/simnet/libsrc/libsched/sched.c	2.2.1.5.2
warn_about_point_space	/simnet/libsrc/libquad/terrain.c	2.12.1.1.6
water_check	/simnet/libsrc/libquad/water_check.c	2.12.1.4.4
water_segments_thru	/simnet/libsrc/libquad/water_check.c	2.12.1.4.11
water_thru	/simnet/libsrc/libquad/water_check.c	2.12.1.4.9
waypoint_vel	/simnet/src/host/pilot.c	2.6.4.2.11
weapon_configure	/simnet/src/host/weapons.c	2.6.9.8.4
weapon_deselect	/simnet/src/host/weapons.c	2.6.9.8.15
weapon_load	/simnet/src/host/weapons.c	2.6.9.8.16
weapon_priority_list_show	/simnet/src/host/weapons.c	2.6.9.8.7
weapon_select	/simnet/src/host/weapons.c	2.6.9.8.14
weapon_show	/simnet/src/host/weapons.c	2.6.9.8.6
weapon_state_to_string	/simnet/src/host/weapons.c	2.6.9.8.1
weapon_systems_checkpoint	/simnet/src/host/weapons.c	2.6.9.8.9
weapon_systems_rearm	/simnet/src/host/weapons.c	2.6.9.8.8
weapon_systems_reinit	/simnet/src/host/weapons.c	2.6.9.8.10
weapon_systems_show	/simnet/src/host/weapons.c	2.6.9.8.5
weapon_unload	/simnet/src/host/weapons.c	2.6.9.8.17
weapons.h	/simnet/src/host/weapons.h	2.6.9.9
which_event_ring	/simnet/libsrc/libsched/maint.c	2.2.1.3.6
which_side	/simnet/libsrc/libutil/math.c	2.14.3.5.9
within_delta	/simnet/src/host/misc.c	2.14.1.2.10
write_buffer_all_sbx	/simnet/src/host/sbx.c	2.4.3.2.3
xy_dir_and_range	/simnet/src/host/pilot.c	2.6.4.2.19
yyerror	/simnet/libsrc/libreader/parser.y	2.1.1.1.6
z_velocity	/simnet/src/host/pilot.c	2.6.4.2.13
zero_perf_stats	/simnet/libsrc/libsched/perf.c	2.2.1.4.4

APPENDIX C

This appendix contains a function called-by tree. The left-hand column is a list of the functions called. The right-hand column is a listing of the calling function(s). The file in which the calling function is defined is listed in parenthesis.

Called Function	Calling Function (filename)
AddSubscription	AssocSubscribeWithMask (subscribe.c)
AppearanceDiscrepancyExceedsThresholds	simnet_send_appearance (SimulationPDU.c)
ApplicationGetSiteHost	AssocReadParams (params.c)
AssocAddToBucket	AssocAddTransaction (origin.c)
	AssocCacheResponse (respondent.c)
AssocAddToEndOfTimeList	AssocAddTransaction (origin.c)
	AssocCacheResponse (respondent.c)
AssocAddTransaction	AssocSendTransact (transact.c)
AssocBucketLookup	AssocFindTransaction (origin.c)
	AssocFindResponse (respondent.c)
AssocCacheResponse	AssocSendResponse (transact.c)
AssocClose	simnet_exit (simnet.c)
AssocCreateFreeList	AssocOpen (open.c)
AssocCreateMCAWithMask	AssocSendTransact (transact.c)
	AssocSendDatagram (send.c)
	AssocSendResponse (transact.c)
	AssocSubscribeWithMask (subscribe.c)
	AssocSendAggregate (aggregate.c)
	AssocUnsubscribeWithMask (subscribe.c)
AssocCurrentlySubscribed	PointToPointReceivePDU (receive.c)
AssocCurrentlySubscribedWithMask	AssocCurrentlySubscribed (subscribe.c)
AssocDeleteCachedResponse	AssocTimeOutOldResponses (respondent.c)
AssocDeleteFromBucket	AssocDeleteCachedResponse (respondent.c)
	AssocDeleteTransaction (origin.c)
AssocDeleteFromTimeList	AssocDeleteCachedResponse (respondent.c)
	AssocDeleteTransaction (origin.c)
AssocDeleteTransaction	AssocProcessResponsePDU (proc_rsp.c)
	UpdateTransactions (tick.c)
AssocError	simnet_tick (simnet.c)
	start_simnet (simnet.c)
AssocFindResponse	AssocProcessRequestPDU (proc_req.c)
AssocFindTransaction	AssocProcessResponsePDU (proc_rsp.c)
AssocFreeDescriptor	AssocDeleteCachedResponse (respondent.c)
	AssocDeleteTransaction (origin.c)
	AssocSendResponse (transact.c)
AssocGetDescriptor	AssocSendTransact (transact.c)
	AssocSendResponse (transact.c)
AssocGetLastAddress	get_sender_of_last_packet (simnet.c)
AssocGetSimAddress	PointToPointOpen (init.c)
AssocGrowFreeList	AssocGetDescriptor (free_list.c)
AssocInitResponses	AssocOpen (open.c)
AssocInitTransactions	AssocOpen (open.c)
AssocMoveToEndOfTimeList	AssocRescheduleTransaction (origin.c)
AssocOpen	PointToPointOpen (init.c)
AssocPadBuffer	AssocSendTransact (transact.c)
	AssocSendDatagram (send.c)
	AssocSendResponse (transact.c)

Called Function	Calling Function (filename)
AssocPadBuffer (cont'd)	AssocSendAggregate (aggregate.c)
AssocProcessRequestPDU	AssocReceivePDU (receive.c)
AssocProcessResponsePDU	AssocReceivePDU (receive.c)
AssocReadParams	AssocOpen (open.c)
AssocReceivePDU	PointToPointReceivePDU (receive.c)
AssocRescheduleTransaction	UpdateTransactions (tick.c)
AssocSendAggregate	AssocTickAssocLayer (tick.c)
AssocSendDatagram	ivis_send_contact_report (IvisPDU.c)
	ivis_send_shell_report (IvisPDU.c)
	ivis_send_spot_report (IvisPDU.c)
	PointToPointSendPDU (send.c)
	simnet_send_appearance (SimulationPDU.c)
	simnet_send_collision (SimulationPDU.c)
	simnet_send_deactivate (SimulationPDU.c)
	simnet_send_fire (SimulationPDU.c)
	simnet_send_impact (SimulationPDU.c)
	simnet_send_indirect_fire (SimulationPDU.c)
	simnet_send_radiate (SimulationPDU.c)
	simnet_send_resupply_cancel (SimulationPDU.c)
	simnet_send_resupply_received (SimulationPDU.c)
	simnet_send_resupply_request (SimulationPDU.c)
	simnet_send_status_change (DataCollectionPDU.c)
AssocSendResponse	simnet_tick (simnet.c)
AssocSendTransact	simnet_send_collision (SimulationPDU.c)
	simnet_send_impact (SimulationPDU.c)
AssocSubscribe	start_simnet (simnet.c)
	PointToPointOpen (init.c)
AssocSubscribeWithMask	AssocSubscribe (subscribe.c)
AssocTickAssocLayer	simnet_tick (simnet.c)
AssocTimeOutOldResponses	AssocGetDescriptor (free_list.c)
	AssocTickAssocLayer (tick.c)
AssocUnsubscribeWithMask	AssocClose (close.c)
BeginningOfLine	CopyEarlierCommand (par_hist.c)
ClearLine	ParseInput (par_base.c)
CommandLog	ParseList (par_base.c)
CopyEarlierCommand	PreviousCommand (par_hist.c)
Delete1Backward	DoHelp (par_util.c)
DeleteChar	KillForward (par_edit.c)
DeleteSubscription	AssocUnsubscribeWithMask (subscribe.c)
Do_Hash	get_symbol (symbol.c)
EmptyLine	ParseInput (par_base.c)
EndOfLine	ParseInput (par_base.c)
FieldGet	TypeFields (par_fields.c)
ForwardChar	RightWord (par_edit.c)
Free	CommandLog (par_hist.c)
InitParser	tty_setup_parser_break_set (geyer.c)
InsertChar	CopyEarlierCommand (par_hist.c)
KillForward	CopyEarlierCommand (par_hist.c)
LookForward	GobbleWord (par_edit.c)
MarkCursor	ParseGetToken (par_util.c)
NextChar	ParseEscapeComplete (par_util.c)
ParseConvertNumber	ParseGetDecimal (par_lib.c)

Called Function	Calling Function (filename)
ParseError	ParseKeyword (par_base.c)
ParseEscapeComplete	ParseKeyword (par_base.c)
ParseFindEndList	ParseKeyword (par_base.c)
ParseGetToken	ParseList (par_base.c)
ParseInSet	ParseGetToken (par_util.c)
ParseInput	DoEscape (par_util.c)
ParseKeyword	ParseList (par_base.c)
ParseList	ParseKeyword (par_base.c)
ParseMatch	ParseKeyword (par_base.c)
ParseMessage	ParseList (par_base.c)
ParseMustFree	ParseGetString (par_lib.c)
ParsePrint	ForwardChar (par_edit.c)
ParseTableFind	ParseGetFields (par_fields.c)
PointToPointOpen	start_simnet (simnet.c)
PointToPointReceivePDU	simnet_tick (simnet.c)
PointToPointSendPDU	stealth_send_attach_to_vehicle (StealthPDU.c)
	stealth_send_teleport_to (StealthPDU.c)
PrepareDiscrepancyThresholds	create_missile (missile.c)
	create_saf_vehicle (saf_vehicle.c)
RedisplayLine	ParseInput (par_base.c)
RestoreCursor	ParseEscapeComplete (par_util.c)
SAF_NET_SND_KLUDGE	AssocProcessRequestPDU (proc_req.c)
	AssocSendAggregate (aggregate.c)
	AssocSendDatagram (send.c)
	AssocSendResponse (transact.c)
	AssocSendTransact (transact.c)
	UpdateTransactions (tick.c)
ScanBackwards	DeleteWord (par_edit.c)
ScanForward	RightWord (par_edit.c)
SetChannelDefaults	AssocOpen (open.c)
SetStopPoint	DoEscape (par_util.c)
TypeFields	ParseList (par_base.c)
UpdateLine	RedisplayLine (par_edit.c)
	parser_invoke_fcn_on_char (geyer.c)
UpdateMoveCursor	UpdateLine (par_edit.c)
UpdateTransactions	AssocTickAssocLayer (tick.c)
abort	_exit (_exit.s)
	air_tick (flyingveh.c)
	airveh_catastrophic_kill (flyingveh.c)
	bse (breakset.c)
	buffer_allocate (queue_fix.c)
	driver_follow_leader (driver.c)
	driver_follow_vehicle_cmd (driver.c)
	driver_set_leader_mis (driver.c)
	find_new_host_mask (id_hash.c)
	get_args (fncl.c)
	idle_tick (pilot.c)
	init_mappings (map.c)
	init_pilot_state_machine (pilot.c)
	invoke (invoke.c)
	p_follower_followroute_tick (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)

Called Function	Calling Function (filename)
abort (cont.)	p_follower_landhold_tick (p_follower.c)
	pilot_follow_leader (pilot.c)
	pilot_follow_leader_tick (p_follower.c)
	pilot_followroute_tick (pilot.c)
	pilot_goto_point_im (pilot.c)
	pilot_hold (pilot.c)
	pilot_hoverattack_tick (pilot.c)
	pilot_hoverhold_tick (pilot.c)
	pilot_init_followroute (pilot.c)
	pilot_landhold_tick (pilot.c)
	pilot_landing (pilot.c)
	pilot_orbit_tick (pilot.c)
	pilot_orbithold (pilot.c)
	pilot_set_leader_mis (pilot.c)
	pilot_start_landed (pilot.c)
	pilot_start_landing (pilot.c)
	pilot_start_takingoff (pilot.c)
	pilot_tick2 (pilot.c)
	remove_function (maint.c)
	saf_vehicle_set_speed (saf_vehicle.c)
	simple_queue_allocate (bufpool.c)
	which_event_ring (maint.c)
abs	next_road_point (roads.c)
accepted	_seterr_reply (rpc_prot.c)
access	fgetpwent (ht_getpwent.c)
	tzload (ctime.c)
access_147	net_access (net_acce.c)
access_cmc	net_access (net_acce.c)
acos	interior_angle_between_vectors (math.c)
	missile_show (missile.c)
acquire_target	gunner_tick (gunner.c)
action_drill_needed	check_predicates (predicates.c)
activate_new_cis	navigator_entered_area (navigator.c)
	navigator_entered_zone (navigator.c)
	navigator_passed_line (navigator.c)
	cm_execute_overlay (cm.c)
	navigator_passed_point (navigator.c)
add_executing_unit	cm_execute_overlay (cm.c)
add_into_spotter_table	spotter_sum_inferior_composite_tables (spotter.c)
add_last_hit	sort_trim_hits (pve_checkvis.c)
add_removed_vehicle	remove_clustered_vehicle (reporter.c)
add_shelling_to_cluster	cluster_shell (reporter.c)
add_vehicle_to_cluster	cluster_vehicle (reporter.c)
add_weapons_status	composite_generate_status_report (composite.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
additem	dbm_store (ndbm.c)
addtominuslist	yp_getgrent (yp_getgrent.c)
addtominuslist	yp_getpwent (yp_getpwent.c)
adjust_behaviors	push_cis (cis.c)
	pop_cis (cis.c)
	pop_resumable_cis (cis.c)
air_raid_happening	check_predicates (predicates.c)

Called Function	Calling Function (filename)
air_raid_over	check_predicates (predicates.c)
air_tick	saf_vehicle_tick (saf_vehicle.c)
airveh_catastrophic_kill	saf_vehicle_catastrophic_kill (saf_vehicle.c)
airveh_init	create_airveh (flyingveh.c)
airveh_show	saf_vehicle_show (saf_vehicle.c)
alarm	syslog (syslog.c)
allocate_bitfield	create_reporter (reporter.c)
	allocate_spotter_tables (spotter.c)
	sbx_connection_init (sbx.c)
	sbx_connection_init (sbx.c)
allocate_points	create_bridges (bridges.c)
	create_buildings (trees.c)
	create_contours (trees.c)
	create_lakes (bridges.c)
	create_rivers (rivers.c)
	create_roads (roads.c)
	create_trees (trees.c)
allocate_safobj	create_remote (remote.c)
	create_composite (composite.c)
	create_saf_vehicle (saf_vehicle.c)
allocate_spotter_tables	create_spotter (spotter.c)
ammo_to_shell_type	cluster_shell (reporter.c)
angle_between_vectors	groundveh_move (groundveh.c)
angle_clip	turret_point_at_target (turret.c)
	turret_scan (turret.c)
	turret_slew (turret.c)
	vehicle_fill_in_position_data (vehicle.c)
any	_gethtent (ht_getnamadr.c)
	_host_interpret (ht_gethent.c)
	ht_getnetent (ht_gnetent.c)
	ht_getprotoent (ht_getpent.c)
	ht_getservent (ht_gservent.c)
	interpret (ht_grpcent.c)
	interpret (yp_gethostent.c)
	interpret (yp_getpent.c)
	interpret (yp_getservent.c)
	interpret (yp_gnetent.c)
append_routepoint	expand_road_route (route.c)
	generate_in_place_route (route.c)
	generate_route_from_route_msg (route.c)
are_we_there	idle_tick (pilot.c)
arty_type_string	sbx_connection_generate_arty_msg (sbx.c)
asctime	ctime (ctime.c)
asin	turret_point_at_target (turret.c)
	helo_tick (helo.c)
atan	missile_show (missile.c)
atan2	create_new_cluster (reporter.c)
	create_saf_vehicle (saf_vehicle.c)
	entity_fill_in_position_data (entity.c)
	groundveh_move (groundveh.c)
	helo_tick (helo.c)
	pilot_start_landed (pilot.c)

Called Function	Calling Function (filename)
atan2 (cont'd)	remote_tick (remote.c)
	saf_vehicle_teleport (saf_vehicle.c)
	turret_point_at_target (turret.c)
	update_cluster_heading (reporter.c)
	vehicle_fill_in_position_data (vehicle.c)
atof	yylex (lexer.c)
	_innum (doscan.c)
toi	interpret (yp_getrpcent.c)
attackatwill_tick	pilot_tick (pilot.c)
auth_errmsg	clnt_sperror (clnt_perror.c)
authnone_create	clntudp_bufcreate (clnt_udp.c)
average	sort_trim_hits (pve_checkvis.c)
bcmp	yp_match (yp_match.c)
bcopy	AssocPadBuffer (send.c)
	AssocSendAggregate (aggregate.c)
	AssocSendDatagram (send.c)
	AssocSendResponse (transact.c)
	AssocSendTransact (transact.c)
	buf_rudp_write_message (buf_conn.c)
	commander_note_leader_state (commander.c)
	commander_restore_leader_state (commander.c)
	composite_fill_in_appearance_data (composite.c)
	composite_get_composite_bumpers (composite.c)
	copy_bitfield (bitfield.c)
	dns_gethostbyaddr (dns_getnamadr.c)
	do_mode_cmd_cmc (net_mode.c)
	fill_ivis_report_tags (IvisPDU.c)
	get_args (fncl.c)
	get_control_measure (cm.c)
	get_overlay (cm.c)
	getanswer (dns_getnamadr.c)
	interesting_packet_test (simnet.c)
	ivis_send_contact_report (IvisPDU.c)
	ivis_send_shell_report (IvisPDU.c)
	ivis_send_spot_report (IvisPDU.c)
	net_get_parameters (net_open.c)
	net_get_rcv (net_recv.c)
	next_road_point (roads.c)
	open_cmc (net_open.c)
	PointToPointSendPDU (send.c)
	remote_tick (remote.c)
	res_mkquery (res_mkquery.c)
	res_querydomain (res_query.c)
	rudp_post_message (rel_conn.c)
	SAF_NET_SND_KLUDGE (simnet.c)
	sbx_connection_reset_msg (sbx.c)
	send_cmc (net_osend.c)
	send_cmc_8023 (net_send.c)
	send_contact_report (reporter.c)
	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
	simnet_send_appearance (SimulationPDU.c)

Called Function	Calling Function (filename)
bcopy (cont.)	simnet_send_resupply_received (SimulationPDU.c)
	simnet_send_resupply_request (SimulationPDU.c)
	simnet_tick (simnet.c)
	vehicle_fill_in_appearance_data (vehicle.c)
	xdrmem_getbytes (xdr_mem.c)
	xdrmem_getlong (xdr_mem.c)
	xdrmem_putbytes (xdr_mem.c)
	xdrmem_putlong (xdr_mem.c)
binarysearch_tag	find_tag_sorted (tags.c)
bind	udp_open (udp_berkeley.c)
	bindresvport (bindresvport.c)
bindresvport	clntudp_bufcreate (clnt_udp.c)
broadcast_appearance_data	saf_vehicle_catastrophic_kill (saf_vehicle.c)
	saf_vehicle_mobility_kill (saf_vehicle.c)
	remote_tick (remote.c)
	saf_vehicle_firepower_kill (saf_vehicle.c)
broadcast_pae_data	remote_tick (remote.c)
	unit_send_pae_data (composite.c)
broadcast_veh_is_gone	remote_deactivate (remote.c)
bse	print_break_set (breakset.c)
	tty_invoke_fcn_on_char (breakset.c)
	map_key (breakset.c)
	parser_invoke_fcn_on_char (geyer.c)
buf_rudp_close	sbx_connection_exit (sbx.c)
buf_rudp_discard_all_buffers	buf_rudp_disconnect (buf_conn.c)
	sbx_connection_reset_msg (sbx.c)
	buf_rudp_read_message (buf_conn.c)
buf_rudp_disconnect	sbx_connection_disconnect (sbx.c)
buf_rudp_flush	buf_rudp_write_message (buf_conn.c)
	buf_rudp_tick (buf_conn.c)
	sbx_connection_reset_msg (sbx.c)
	sbx_connection_wat_pulse (sbx.c)
buf_rudp_init	sbx_connection_init (sbx.c)
buf_rudp_open	sbx_connection_open (sbx.c)
buf_rudp_read_message	sbx_connection_tick (sbx.c)
buf_rudp_show	sbx_connection_show (sbx.c)
buf_rudp_tick	sbx_connection_wat_pulse (sbx.c)
	sbx_connection_reset_msg (sbx.c)
	sbx_connection_tick (sbx.c)
buf_rudp_write_message	buf_rudp_tick (buf_conn.c)
buffer__deallocate	buffer_deallocate (queue_fix.c)
buffer_allocate	activate_new_cis (cis.c)
	broadcast_appearance_data (sbx.c)
	broadcast_pae_data (sbx.c)
	broadcast_veh_is_gone (sbx.c)
	buffer_allocate_into_pool (bufpool.c)
	change_movement_and_report (cis.c)
	composite_generate_status_report (composite.c)
	create_missile (missile.c)
	create_saf_vehicle (saf_vehicle.c)
	gasp (main.c)
	ivis_send_contact_report (IvisPDU.c)

Called Function	Calling Function (filename)
buffer_allocate (cont.)	ivis_send_spot_report (IvisPDU.c)
	queue_allocate (queue_fix.c)
	remote_tick (remote.c)
	rudp_post_message (rel_conn.c)
	saf_vehicle_checkpoint_state (saf_vehicle.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
	sbx_connection_process_ground_impact (sbx.c)
	sbx_connection_process_indirect_fire (sbx.c)
	sbx_connection_process_vehicle_impact (sbx.c)
	sbx_connection_reset_msg (sbx.c)
	sbx_connection_stat_pulse (sbx.c)
	sbx_connection_synch_received (sbx.c)
	sbx_connection_wat_pulse (sbx.c)
	sbx_printf (sbx.c)
	send_contact_report (reporter.c)
	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
	send_stealth_gone_msg (remote.c)
	simnet_send_appearance (SimulationPDU.c)
	simnet_send_collision (SimulationPDU.c)
	simnet_send_deactivate (SimulationPDU.c)
	simnet_send_fire (SimulationPDU.c)
	simnet_send_impact (SimulationPDU.c)
	simnet_send_indirect_fire (SimulationPDU.c)
	simnet_send_radiate (SimulationPDU.c)
	simnet_send_resupply_cancel (SimulationPDU.c)
	simnet_send_resupply_received (SimulationPDU.c)
	simnet_send_resupply_request (SimulationPDU.c)
	simnet_send_status_change (DataCollectionPDU.c)
	simnet_tick (simnet.c)
	stealth_send_attach_to_vehicle (StealthPDU.c)
	stealth_send_teleport_to (StealthPDU.c)
buffer_allocate_from_pool	simnet_tick (simnet.c)
buffer_allocate_into_pool	buffer_pool_allocate (bufpool.c)
buffer_deallocate	activate_new_cis (cis.c)
	broadcast_appearance_data (sbx.c)
	broadcast_pae_data (sbx.c)
	broadcast_veh_is_gone (sbx.c)
	buf_rudp_tick (buf_conn.c)
	buffer_simple_flush (bufpool.c)
	change_movement_and_report (cis.c)
	composite_generate_status_report (composite.c)
	composite_tick (composite.c)
	destroy_logistics (logistics.c)
	destroy_missile (missile.c)
	gasp (main.c)
	ivis_send_contact_report (IvisPDU.c)
	ivis_send_shell_report (IvisPDU.c)
	ivis_send_spot_report (IvisPDU.c)
	remote_deactivate (remote.c)
	remote_go_away (remote.c)
	remote_new_appearance (remote.c)

Called Function	Calling Function (filename)
buffer_deallocate (cont'd)	remote_tick (remote.c)
	rudp_ack_received (rel_conn.c)
	rudp_post_message (rel_conn.c)
	saf_vehicle_checkpoint_state (saf_vehicle.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
	saf_vehicle_go_away (saf_vehicle.c)
	saf_vehicle_tick (saf_vehicle.c)
	sbx_connection_process_ground_impact (sbx.c)
	sbx_connection_process_indirect_fire (sbx.c)
	sbx_connection_process_vehicle_impact (sbx.c)
	sbx_connection_stat_pulse (sbx.c)
	sbx_connection_synch_received (sbx.c)
	sbx_connection_wat_pulse (sbx.c)
	sbx_printf (sbx.c)
	send_contact_report (reporter.c)
	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
	send_stealth_gone_msg (remote.c)
	simnet_send_appearance (SimulationPDU.c)
	simnet_send_collision (SimulationPDU.c)
	simnet_send_deactivate (SimulationPDU.c)
	simnet_send_fire (SimulationPDU.c)
	simnet_send_impact (SimulationPDU.c)
	simnet_send_indirect_fire (SimulationPDU.c)
	simnet_send_radiate (SimulationPDU.c)
	simnet_send_resupply_cancel (SimulationPDU.c)
	simnet_send_resupply_received (SimulationPDU.c)
	simnet_send_resupply_request (SimulationPDU.c)
	simnet_send_status_change (DataCollectionPDU.c)
	simnet_tick (simnet.c)
	stealth_send_attach_to_vehicle (StealthPDU.c)
	stealth_send_teleport_to (StealthPDU.c)
buffer_dequeue	buf_rudp_tick (buf_conn.c)
	rudp_ack_received (rel_conn.c)
buffer_enqueue	sbx_connection_write_buffer (sbx.c)
	rudp_post_message (rel_conn.c)
buffer_flush	rudp_discard_all_buffers (rel_conn.c)
	buf_rudp_discard_all_buffers (buf_conn.c)
buffer_length	rudp_retransmit_buf (rel_conn.c)
	buf_rudp_tick (buf_conn.c)
buffer_one_less	buffer__deallocate (queue_fix.c)
	buffer_allocate_into_pool (bufpool.c)
buffer_one_more	buffer_allocate (queue_fix.c)
	buffer_allocate_from_pool (bufpool.c)
buffer_pool_allocate	start_simnet (simnet.c)
buffer_pool_return_buffer	buffer__deallocate (queue_fix.c)
	buffer_allocate_into_pool (bufpool.c)
buffer_simple_dequeue	buffer_simple_flush (bufpool.c)
	composite_tick (composite.c)
	remote_tick (remote.c)
	saf_vehicle_tick (saf_vehicle.c)
buffer_simple_enqueue	enqueue_on_rcvq (simnet.c)

Called Function	Calling Function (filename)
buffer_simple_enqueue (cont'd)	protocol_sim_process (simnet.c)
buffer_simple_flush	remote_tick (remote.c)
	destroy_entity (entity.c)
buffer_statistics_init	buffer_one_less (queue_fix.c)
	buffer_one_more (queue_fix.c)
	buffer_statistics_print (queue_fix.c)
buffer_traverse_and_apply	rudp_ack_received (rel_conn.c)
buffer_traverse_and_apply_n_times	rudp_retransmit (rel_conn.c)
buffer_up_refcnt	remote_new_appearance (remote.c)
buildings_thru	detect_building_on_path_tick (collision.c)
bzero	allocate_bitfield (bitfield.c)
	allocate_safobj (safobj.c)
	bindresvport (bindresvport.c)
	broadcast_appearance_data (sbx.c)
	broadcast_pae_data (sbx.c)
	calloc (calloc.c)
	clear_bitfield (bitfield.c)
	clntudp_call (clnt_udp.c)
	count_hosts (parser.c)
	count_sites (parser.c)
	create_composite (composite.c)
	create_missile (missile.c)
	create_saf_vehicle (saf_vehicle.c)
	create_vehicle (vehicle.c)
	do_mode_cmd_cmc (net_mode.c)
	find_new_host_mask (id_hash.c)
	get_formation_layout (formation.c)
	get_symbol (symbol.c)
	init_grid_tables (iterator.c)
	init_saf_to_simnet_id_table (id_hash.c)
	init_safobj_table (safobj.c)
	init_symbol_table (symbol.c)
	map_buffers (net_stuf.c)
	open_cmc (net_open.c)
	open_io_connections (sbx.c)
	parse_servfile (vis.c)
	res_send (res_send.c)
	saf_vehicle_checkpoint_state (saf_vehicle.c)
	sbx_connection_stat_pulse (sbx.c)
	simnet_send_fire (SimulationPDU.c)
	weapon_configure (weapons.c)
	zero_perf_stats (perf.c)
cache_and_file_terminate	exit_handler (main.c)
cache_init	tdb_init_cache (tdb_init.c)
cachectl	map_buffers (net_stuf.c)
calloc	_findiop (findiop.c)
	_f_morefiles (findiop.c)
	f_prealloc (findiop.c)
	heap_calloc (alloc.c)
cancel_fncl	sbx_connection_disconnect (sbx.c)
	stop_ticking (tickable.c)
	sbx_connection_reset_msg (sbx.c)

Called Function	Calling Function (filename)
cancel_fncl (cont'd)	cancel_fncl_group (fncl.c)
cancel_fncl_group	composite_go_away (composite.c)
	remote_go_away (remote.c)
	saf_vehicle_go_away (saf_vehicle.c)
canopies	patch (pve_checkvis.c)
center_column	minimum_clip (minimum_clip.c)
change_fncl_period	change_tick_rate (tickable.c)
change_movement_and_report	adjust_behaviors (cis.c)
change_tick_rate	remote_start_being_watched (remote.c)
change_tick_rate (cont'd)	saf_vehicle_tick (saf_vehicle.c)
	remote_tick (remote.c)
	remote_stop_being_watched (remote.c)
check_binder_up	_yp_dobind (yp_bind.c)
check_binding	_yp_dobind (yp_bind.c)
check_box	treelines (pve_checkvis.c)
	canopies (pve_checkvis.c)
	objects (pve_checkvis.c)
check_edge_hit	check_edges (pve_checkvis.c)
heck_edges	canopies (pve_checkvis.c)
	terrain (pve_checkvis.c)
check_hits	patch (pve_checkvis.c)
check_object	objects (pve_checkvis.c)
check_pmap_up	get_binder_port (yp_bind.c)
check_predicates	commander_tick (commander.c)
check_prob	get_i_and_t_from_normal (impact.c)
	fire_weapon_at_target (weapons.c)
	target_type_ok (targeting.c)
check_tree_hit	treelines (pve_checkvis.c)
	trees (pve_checkvis.c)
choose_resupply_item	logistics_tick (logistics.c)
choose_skirt_point	start_avoiding_house (collision.c)
choose_target_and_weapon	targeting_tick (targeting.c)
is_show	composite_show (composite.c)
cleanup	ht_fgetgrent (ht_getgrent.c)
clear_bit	reporter_remove_vehicles (reporter.c)
	spotter_remove_vehicles (spotter.c)
clear_bitfield	create_reporter (reporter.c)
	sbx_connection_init (sbx.c)
	sbx_set_specific_known_vehicles (sbx.c)
	sbx_set_top_level_known_vehicles (sbx.c)
	spotter_init_table (spotter.c)
clear_cis_stack	cm_execute_overlay (cm.c)
clear_detection	create_detection (detection.c)
clear_monitor_variables	print_reasons_and_clear (parser.c)
clear_weapons_status	composite_generate_status_report (composite.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
clip_angle_positive	turret_slew (turret.c)
	stealth_send_teleport_to (StealthPDU.c)
clip_to_tdb	startup (pve_checkvis.c)
clnt_spcreateerror	clnt_pcreateerror (clnt_perror.c)
lnt_sperrno	clnt_perrno (clnt_perror.c)
	clnt_spcreateerror (clnt_perror.c)

Called Function	Calling Function (filename)
lnt_sperrno (cont'd)	clnt_sperror (clnt_perror.c)
clnt_sperror	clnt_perror (clnt_perror.c)
clntudp_bufcreate	talk2_binder (yp_bind.c)
	pmap_getport (pmap_getport.c)
	clntudp_create (clnt_udp.c)
close	_res_close (res_send.c)
	cache_and_file_terminate (cache_alloc.c)
	clntudp_destroy (clnt_udp.c)
	closelog (syslog.c)
	fclose (flsbuf.c)
close (cont'd)	get_type (net_open.c)
	open_cmc (net_open.c)
	pmap_getport (pmap_getport.c)
	read_quadtree (terrain.c)
	res_send (res_send.c)
	syslog (syslog.c)
	tdb_get_db_format (version.c)
	tzload (ctime.c)
	udp_close (udp_berkeley.c)
closest_leader	compute_who_follows_whom (formation.c)
closest_to_a_leader	compute_who_follows_whom (formation.c)
cluster_from_vehicle	action_drill_needed (predicates.c)
	hasty_attack_needed (predicates.c)
cluster_shell	composite_indirect_fire (composite.c)
cluster_vehicle	update_reporter_vehicles (reporter.c)
	update_clusters (reporter.c)
cm_add_area	sbx_add_area (sbx.c)
cm_add_line	sbx_add_line (sbx.c)
cm_add_point	sbx_add_point (sbx.c)
cm_add_route	sbx_add_route (sbx.c)
cm_add_zone	sbx_add_zone (sbx.c)
cm_applies_to_unit	cm_recompute_cm_from_overlay (cm.c)
	cm_get_cm_from_overlay (cm.c)
cm_count_intersections	cm_point_inside_polygon (navigator.c)
cm_delete_cm	sbx_delete_cm (sbx.c)
cm_delete_cm_from_overlay	cm_delete_cm (cm.c)
	cm_delete_overlay (cm.c)
cm_delete_overlay	sbx_delete_overlay (sbx.c)
	sbx_connection_reset_msg (sbx.c)
cm_execute_overlay	sbx_execute_overlay (sbx.c)
cm_find_cm	cm_execute_overlay (cm.c)
cm_find_cm_list_item	cm_delete_cm_from_overlay (cm.c)
	cm_find_cm (cm.c)
cm_find_overlay	cm_delete_cm (cm.c)
	cm_execute_overlay (cm.c)
cm_find_overlay_list_item	cm_find_overlay (cm.c)
	cm_delete_overlay (cm.c)
cm_force_overlay_recalculation	cm_add_line (cm.c)
	cm_delete_cm (cm.c)
	cm_add_zone (cm.c)
	cm_add_point (cm.c)
	cm_add_area (cm.c)

Called Function	Calling Function (filename)
cm_get_center_of_mass	change_movement_and_report (cis.c)
cm_get_cm_from_overlay	navigator_set_overlay (navigator.c)
cm_intersection	navigator_tick (navigator.c)
cm_point_inside_polygon	navigator_tick (navigator.c)
cm_point_line_intersection	navigator_tick (navigator.c)
cm_poly_intersection	navigator_tick (navigator.c)
cm_recompute_cm_from_overlay	navigator_reset_overlay (navigator.c)
cm_to_passage_string	fill_generic_cis_report (cis.c)
collect_perf_stat	invoke_functions_until (sched.c)
collision_dead_tick	saf_vehicle_tick (saf_vehicle.c)
collision_disengaging	saf_vehicle_doing_collision_stuff (saf_vehicle.c)
collision_remove_vehicles	saf_vehicle_remove_vehicles (saf_vehicle.c)
collision_show	saf_vehicle_show (saf_vehicle.c)
collision_tick	saf_vehicle_tick (saf_vehicle.c)
collision_vehicle_rammed	saf_vehicle_vehicle_rammed (saf_vehicle.c)
combined_velocity	p_follower_hoverhold (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
	p_follower_landhold_tick (p_follower.c)
	p_follower_set_desired_vel (p_follower.c)
	pilot_goto_endpoint (pilot.c)
	pilot_gotopoint (pilot.c)
	pilot_orbithold (pilot.c)
commander_attack	composite_attack (composite.c)
commander_cancel_overlay	composite_cancel_overlay (composite.c)
commander_change_altitude	composite_change_altitude (composite.c)
commander_change_formation	composite_change_formation (composite.c)
commander_change_speed	composite_change_speed (composite.c)
commander_execute_overlay	composite_execute_overlay (composite.c)
commander_executing_order	composite_executing_order (composite.c)
commander_face_direction	composite_face_direction (composite.c)
commander_follow_vehicle	composite_follow_vehicle (composite.c)
commander_get_mission_status	composite_generate_status_report (composite.c)
commander_goto_point	change_movement_and_report (cis.c)
	composite_goto_point (composite.c)
commander_halt	change_movement_and_report (cis.c)
	composite_halt (composite.c)
commander_inferior_changed_status	composite_inferior_changed_status (composite.c)
commander_land	composite_land (composite.c)
commander_note_leader_state	composite_note_leader_state (composite.c)
	composite_assume_formation (composite.c)
commander_restore_leader_state	composite_reassign_current_formation (composite.c)
	composite_assume_formation (composite.c)
commander_resume_mission	composite_resume_mission (composite.c)
commander_set_mission_direction	change_movement_and_report (cis.c)
commander_set_mission_formation	adjust_behaviors (cis.c)
commander_set_mission_speed	adjust_behaviors (cis.c)
commander_show	composite_show (composite.c)
commander_simulator_in_command	composite_simulator_in_command (composite.c)
commander_state_string	commander_show (commander.c)
commander_tick	composite_tick (composite.c)
composite_add_inferior_composite	composite_set_superior (composite.c)
composite_add_inferior_vehicle	saf_vehicle_set_superior (saf_vehicle.c)

Called Function	Calling Function (filename)
composite_add_member_vehicle	composite_add_member_vehicle (composite.c)
	composite_add_inferior_vehicle (composite.c)
composite_assume_formation	commander_set_mission_formation (commander.c)
	commander_resume_mission (commander.c)
	create_unit (create.c)
	commander_change_formation (commander.c)
composite_assume_formation_internal	composite_reassign_current_formation (composite.c)
	composite_assume_formation (composite.c)
composite_attack	sbx_attack (sbx.c)
composite_cancel_overlay	composite_rejoin_unit (composite.c)
composite_change_altitude	sbx_altitude (sbx.c)
composite_change_formation	sbx_change_formation (sbx.c)
composite_change_speed	sbx_change_speed (sbx.c)
composite_check_unit_strength	composite_tick (composite.c)
composite_distinguished_member	enemy_disposition (predicates.c)
	composite_tick (composite.c)
composite_execute_overlay	execute_overlay (dispatch.c)
composite_executing_order	saf_vehicle_useless (saf_vehicle.c)
composite_face_direction	sbx_face_direction (sbx.c)
composite_fake_resupply	fake_resupply (dispatch.c)
composite_fill_in_appearance_data	fill_in_appearance_data (dispatch.c)
composite_fill_in_echelon_data	fill_in_echelon_data (dispatch.c)
composite_fill_in_position_data	fill_in_position_data (dispatch.c)
composite_find_living_vehicle	composite_distinguished_member (composite.c)
	composite_find_vehicle_for_stealth (composite.c)
composite_find_vehicle_for_stealth	sbx_attach_stealth (sbx.c)
composite_follow_vehicle	sbx_follow_vehicle (sbx.c)
composite_generate_status_report	generate_status_report (dispatch.c)
composite_get_composite_bumpers	composite_get_composite_bumpers (composite.c)
	create_unit (create.c)
composite_go_away	go_away (dispatch.c)
composite_goto_point	sbx_goto_point (sbx.c)
composite_halt	collision_tick (collision.c)
	sbx_hold (sbx.c)
	sbx_halt (sbx.c)
composite_indirect_fire	composite_pro_sim (composite.c)
composite_inferior_changed_status	commander_attack (commander.c)
	commander_cancel_overlay (commander.c)
	commander_change_formation (commander.c)
	commander_execute_overlay (commander.c)
	commander_face_direction (commander.c)
	commander_follow_vehicle (commander.c)
	commander_goto_point (commander.c)
	commander_halt (commander.c)
	commander_inferior_changed_status (commander.c)
	commander_land (commander.c)
	commander_resume_mission (commander.c)
	commander_set_mission_formation (commander.c)
	commander_simulator_in_command (commander.c)
	groundveh_tick (groundveh.c)
	saf_vehicle_cancel_overlay (saf_vehicle.c)
	saf_vehicle_catastrophic_kill (saf_vehicle.c)

Called Function	Calling Function (filename)
composite_inferior_changed_status (cont.)	saf_vehicle_execute_overlay (saf_vehicle.c)
	saf_vehicle_face_direction (saf_vehicle.c)
	saf_vehicle_fake_resupply (saf_vehicle.c)
	saf_vehicle_follow_vehicle (saf_vehicle.c)
	saf_vehicle_goto_point (saf_vehicle.c)
	saf_vehicle_halt (saf_vehicle.c)
	saf_vehicle_mobility_kill (saf_vehicle.c)
	saf_vehicle_out_of_gas (saf_vehicle.c)
	saf_vehicle_resume_mission (saf_vehicle.c)
composite_land	sbx_land (sbx.c)
composite_note_leader_state	commander_attack (commander.c)
	commander_cancel_overlay (commander.c)
	commander_change_formation (commander.c)
composite_note_leader_state (cont'd)	commander_execute_overlay (commander.c)
	commander_face_direction (commander.c)
	commander_follow_vehicle (commander.c)
	commander_goto_point (commander.c)
	commander_halt (commander.c)
	commander_land (commander.c)
	commander_note_leader_state (commander.c)
	commander_resume_mission (commander.c)
	commander_set_mission_formation (commander.c)
	commander_simulator_in_command (commander.c)
	get_a_leader (commander.c)
	groundveh_tick (groundveh.c)
	saf_vehicle_cancel_overlay (saf_vehicle.c)
	saf_vehicle_catastrophic_kill (saf_vehicle.c)
	saf_vehicle_execute_overlay (saf_vehicle.c)
	saf_vehicle_face_direction (saf_vehicle.c)
	saf_vehicle_fake_resupply (saf_vehicle.c)
	saf_vehicle_follow_vehicle (saf_vehicle.c)
	saf_vehicle_goto_point (saf_vehicle.c)
	saf_vehicle_halt (saf_vehicle.c)
	saf_vehicle_mobility_kill (saf_vehicle.c)
	saf_vehicle_out_of_gas (saf_vehicle.c)
	saf_vehicle_resume_mission (saf_vehicle.c)
composite_note_member_vehicle_has_died	saf_vehicle_catastrophic_kill (saf_vehicle.c)
composite_pro_sim	composite_tick (composite.c)
composite_readjust_overlay	readjust_overlay (dispatch.c)
composite_reassign_current_formation	get_a_leader (commander.c)
	commander_inferior_changed_status (commander.c)
composite_rejoin_unit	sbx_rejoin_unit (sbx.c)
composite_remove_inferior_composite	composite_set_superior (composite.c)
composite_remove_inferior_vehicle	saf_vehicle_set_superior (saf_vehicle.c)
composite_remove_member_vehicle	composite_remove_inferior_composite (composite.c)
	composite_remove_inferior_vehicle (composite.c)
composite_remove_vehicles	remove_vehicles (dispatch.c)
composite_resume_mission	sbx_resume_mission (sbx.c)
composite_send_unit_strength_message	composite_check_unit_strength (composite.c)
composite_set_recursive_ids	composite_set_recursive_ids (composite.c)
	composite_set_sbx_ids (composite.c)
composite_set_sbx_ids	create_unit (create.c)

Called Function	Calling Function (filename)
composite_set_superior	create_unit (create.c)
composite_set_targeting_parameters	set_targeting_parameters (dispatch.c)
composite_show	show (dispatch.c)
composite_simulator_in_command	sbx_simulator_in_command (sbx.c)
composite_start_ticking	create_unit (create.c)
composite_teleport_to_station	create_unit (create.c)
compute_damage_keys	database_df_damage_query (df_damage.c)
compute_enemy_weight	detection_tick (detection.c)
compute_explosion_point	missile_maybe_hit_target (missile.c)
compute_interest_direction	detection_tick (detection.c)
compute_mid	treelines (pve_checkvis.c)
compute_situation	pilot_start_gotopoint (pilot.c)
	pilot_goto_endpoint (pilot.c)
	pilot_gotopoint (pilot.c)
compute_who_follows_whom	get_formation_layout (formation.c)
connect	openlog (syslog.c)
	res_send (res_send.c)
cons	push (list.c)
copy_bitfield	update_reporter_vehicles (reporter.c)
copy_float_point_list_to_real_point_list	cm_add_line (cm.c)
	cm_add_zone (cm.c)
	cm_add_area (cm.c)
copy_matrix_row_to_vector	plane_tick (plane.c)
	helo_tick (helo.c)
	remote_tick (remote.c)
	groundveh_move (groundveh.c)
copy_stack_to_array	copy_stack_to_array (parser.y)
	make_array (parser.y)
copy_xy_on_tdb	z_velocity (pilot.c)
	ground_level (pilot.c)
cos	create_entity (entity.c)
	create_missile (missile.c)
	driver_face_direction_cmd (driver.c)
	helo_tick (helo.c)
	hull_to_world_from_orientation (flyingveh.c)
	mat_rot_init (mat_r_init.c)
	pilot_face_direction (pilot.c)
	plane_tick (plane.c)
	PrepareDiscrepancyThresholds (thresh.c)
	turret_muzzle_position_in_world_coordinates (turret.c)
	turret_set_scan_parms (turret.c)
count_intersections	point_inside_polygon (intersection.c)
countries_from_battle_scheme_and_force	parser_create_vehicle (parser.c)
country_to_string	sbx_connection_create_unit_msg (sbx.c)
create_airveh	create_saf_vehicle (saf_vehicle.c)
create_bridges	read_terrain_files (terrain.c)
create_buildings	read_terrain_files (terrain.c)
create_cis	activate_new_cis (cis.c)
	check_predicates (predicates.c)
create_collision	create_saf_vehicle (saf_vehicle.c)
create_commander	create_composite (composite.c)

Called Function	Calling Function (filename)
create_composite	create_unit (create.c)
create_contours	read_terrain_files (terrain.c)
create_damage	create_saf_vehicle (saf_vehicle.c)
create_detection	create_saf_vehicle (saf_vehicle.c)
create_driver	create_saf_vehicle (saf_vehicle.c)
create_entity	create_remote (remote.c)
	create_composite (composite.c)
	create_saf_vehicle (saf_vehicle.c)
create_groundveh	create_saf_vehicle (saf_vehicle.c)
create_lakes	read_terrain_files (terrain.c)
create_logistics	create_saf_vehicle (saf_vehicle.c)
create_missile	weapon_configure (weapons.c)
create_navigator	create_composite (composite.c)
create_new_cluster	cluster_vehicle (reporter.c)
create_new_shelling	cluster_shell (reporter.c)
create_pilot	create_saf_vehicle (saf_vehicle.c)
create_predicates	create_composite (composite.c)
create_remote	create_remote_vehicle (remote.c)
create_remote_vehicle	protocol_sim_process (simnet.c)
	protocol_stealth_process (simnet.c)
create_reporter	create_composite (composite.c)
create_river_ints	read_terrain_files (terrain.c)
create_rivers	read_terrain_files (terrain.c)
create_road_ints	read_terrain_files (terrain.c)
create_roads	read_terrain_files (terrain.c)
create_routepoint	expand_road_route (route.c)
	generate_in_place_route (route.c)
	generate_route_from_route_msg (route.c)
create_saf_vehicle	create_unit (create.c)
create_spotter	create_composite (composite.c)
create_targeting	create_saf_vehicle (saf_vehicle.c)
create_tickable	create_remote (remote.c)
	create_composite (composite.c)
	create_saf_vehicle (saf_vehicle.c)
create_tree_canopies	read_terrain_files (terrain.c)
create_trees	read_terrain_files (terrain.c)
create_turret	create_saf_vehicle (saf_vehicle.c)
create_unit	sbx_connection_create_unit_msg (sbx.c)
	parser_create_vehicle (parser.c)
create_vehicle	create_remote (remote.c)
	create_saf_vehicle (saf_vehicle.c)
create_weapon_systems	create_saf_vehicle (saf_vehicle.c)
ctime	syslog (syslog.c)
cvt	_doprnt (doprnt.c)
damage_indirect_fire	saf_vehicle_indirect_fire (saf_vehicle.c)
damage_string	damage_indirect_fire (damage.c)
	damage_vehicle_impact (damage.c)
damage_vehicle_impact	saf_vehicle_vehicle_impact (saf_vehicle.c)
damage_vehicle_rammed	saf_vehicle_vehicle_rammed (saf_vehicle.c)
database_detection_query	detectable (detection.c)
database_df_damage_query	damage_vehicle_impact (damage.c)
database_if_damage_query	damage_indirect_fire (damage.c)

Called Function	Calling Function (filename)
database_init	init_stuff (main.c)
database_read	database_init (main.c)
dbm_access	dbm_fetch (ndbm.c)
dbm_close	yp_endhostent (yp_gethostent.c)
dbm_nextkey	dbm_firstkey (ndbm.c)
dcalchash	dbm_fetch (ndbm.c)
deallocate_bitfield	destroy_reporter (reporter.c)
	deallocate_spotter_tables (spotter.c)
deallocate_safobj	composite_go_away (composite.c)
	remote_go_away (remote.c)
	saf_vehicle_go_away (saf_vehicle.c)
deallocate_spotter_tables	destroy_spotter (spotter.c)
deferred_fncl	saf_vehicle_catastrophic_kill (saf_vehicle.c)
	sbx_connection_reset_msg (sbx.c)
	sbx_connection_service_poll_msg (sbx.c)
	sbx_connection_stat_pulse (sbx.c)
	sbx_connection_synch_received (sbx.c)
	sbx_connection_wat_pulse (sbx.c)
delitem	dbm_delete (ndbm.c)
dequeue_terrain_patch	rotate_queue (cache_queue.c)
	terrain_cache_inquire (cache_query.c)
destroy_airveh	saf_vehicle_go_away (saf_vehicle.c)
destroy_area_cm	destroy_cm_data (cm.c)
destroy_cis	activate_new_cis (cis.c)
	clear_cis_stack (cis.c)
	composite_go_away (composite.c)
	pop_cis (cis.c)
	pop_resumable_cis (cis.c)
destroy_cm	cm_delete_cm_from_overlay (cm.c)
destroy_cm_data	cm_delete_cm_from_overlay (cm.c)
destroy_cm_list	navigator_reset_overlay (navigator.c)
	navigator_set_overlay (navigator.c)
destroy_cm_list_item	destroy_cm_list (cm.c)
	cm_delete_cm_from_overlay (cm.c)
destroy_collision	saf_vehicle_go_away (saf_vehicle.c)
destroy_damage	saf_vehicle_go_away (saf_vehicle.c)
destroy_detection	saf_vehicle_go_away (saf_vehicle.c)
destroy_driver	saf_vehicle_go_away (saf_vehicle.c)
destroy_entity	composite_go_away (composite.c)
	remote_go_away (remote.c)
	saf_vehicle_go_away (saf_vehicle.c)
destroy_grid_entry_list	remote_deactivate (remote.c)
	remote_go_away (remote.c)
	saf_vehicle_go_away (saf_vehicle.c)
destroy_groundveh	saf_vehicle_go_away (saf_vehicle.c)
destroy_line_cm	destroy_cm_data (cm.c)
destroy_logistics	saf_vehicle_go_away (saf_vehicle.c)
destroy_missile	destroy_weapon_systems (weapons.c)
destroy_overlay	cm_delete_overlay (cm.c)
destroy_overlay_list_item	cm_delete_overlay (cm.c)
destroy_pilot	saf_vehicle_go_away (saf_vehicle.c)
destroy_point_cm	destroy_cm_data (cm.c)

Called Function	Calling Function (filename)
destroy_predicates	composite_go_away (composite.c)
destroy_rem_veh_id_item	remove_removed_vehicle (reporter.c)
	destroy_removed_vehicles (reporter.c)
destroy_removed_vehicles	destroy_reporter (reporter.c)
destroy_reporter	composite_go_away (composite.c)
destroy_route	destroy_cm_data (cm.c)
	cm_add_route (cm.c)
destroy_routepoint	destroy_route (route.c)
destroy_shell_summary	destroy_shellings (reporter.c)
destroy_shellings	destroy_reporter (reporter.c)
destroy_spotter	composite_go_away (composite.c)
destroy_targeting	saf_vehicle_go_away (saf_vehicle.c)
destroy_tickable	composite_go_away (composite.c)
	remote_go_away (remote.c)
	saf_vehicle_go_away (saf_vehicle.c)
destroy_turret	saf_vehicle_go_away (saf_vehicle.c)
destroy_veh_id_item	remove_clustered_vehicle (reporter.c)
	destroy_veh_id_list (reporter.c)
	remove_dead_clustered_vehicles (reporter.c)
destroy_veh_id_list	update_clusters (reporter.c)
	destroy_overlay (cm.c)
	destroy_vehicle_cluster (reporter.c)
destroy_vehicle	remote_go_away (remote.c)
	saf_vehicle_go_away (saf_vehicle.c)
destroy_vehicle_cluster	destroy_vehicle_clusters (reporter.c)
destroy_vehicle_clusters	destroy_reporter (reporter.c)
destroy_weapon_systems	saf_vehicle_go_away (saf_vehicle.c)
destroy_zone_cm	destroy_cm_data (cm.c)
detachnode	yp_match (yp_match.c)
detect_building_on_path_tick	collision_tick (collision.c)
detect_collision_tick	collision_dead_tick (collision.c)
	collision_tick (collision.c)
detect_imminent_collision_tick	collision_tick (collision.c)
detectable	detection_tick (detection.c)
detection_remove_vehicles	saf_vehicle_remove_vehicles (saf_vehicle.c)
detection_save_weight	targeting_tick (targeting.c)
detection_show	saf_vehicle_show (saf_vehicle.c)
detection_show_type	detection_show (detection.c)
etection_tick	targeting_tick (targeting.c)
detzcode	tzload (ctime.c)
diffraction_diff	pve_checkvis (pve_checkvis.c)
diffv	within_delta (misc.c)
distance	cm_point_line_intersection (navigator.c)
	find_closest_building (building_check.c)
dn_comp	res_mkquery (res_mkquery.c)
dn_expand	getanswer (dns_getnamadr.c)
dn_find	dn_comp (res_comp.c)
dn_skipname	getanswer (dns_getnamadr.c)
do_heap_collect	parser_heap_collect (parser.c)
do_heap_statistics	parser_heap_statistics (parser.c)
do_heap_verify	parser_heap_verify (parser.c)
do_mode_cmd	net_norm (net_mode.c)

Called Function	Calling Function (filename)
do_mode_cmd_147	do_mode_cmd (net_mode.c)
do_mode_cmd_cmc	do_mode_cmd (net_mode.c)
doit	yp_setnetgrent (yp_gnetgr.c)
	yp_innetgr (yp_innetgr.c)
dotv	within_delta (misc.c)
driver_change_speed_cmd	saf_vehicle_change_speed (saf_vehicle.c)
driver_execute_overlay	saf_vehicle_execute_overlay (saf_vehicle.c)
driver_executing_immediate_command	saf_vehicle_useless (saf_vehicle.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
driver_face_direction_cmd	saf_vehicle_face_direction (saf_vehicle.c)
driver_follow_leader	saf_vehicle_set_leader (saf_vehicle.c)
driver_follow_vehicle_cmd	saf_vehicle_follow_vehicle (saf_vehicle.c)
driver_forget_about_forps	saf_vehicle_reset_station_keeper (saf_vehicle.c)
driver_goto_point_cmd	saf_vehicle_goto_point (saf_vehicle.c)
driver_halt_cmd	saf_vehicle_halt (saf_vehicle.c)
driver_remove_vehicles	saf_vehicle_remove_vehicles (saf_vehicle.c)
driver_resume_from_collision	saf_vehicle_collision_over (saf_vehicle.c)
driver_resume_mission_cmd	saf_vehicle_resume_mission (saf_vehicle.c)
	driver_execute_overlay (driver.c)
driver_set_direction	saf_vehicle_set_direction (saf_vehicle.c)
driver_set_leader_mis	saf_vehicle_set_leader (saf_vehicle.c)
driver_set_route	saf_vehicle_set_route (saf_vehicle.c)
driver_set_speed	saf_vehicle_set_speed (saf_vehicle.c)
driver_show	saf_vehicle_show (saf_vehicle.c)
driver_simulator_in_command	driver_simulator_in_command (driver.c)
	saf_vehicle_simulator_in_command (saf_vehicle.c)
driver_stop_mission	saf_vehicle_cancel_overlay (saf_vehicle.c)
driver_tick	saf_vehicle_tick (saf_vehicle.c)
dstsetkernel	tzsetkernel (ctime.c)
eliminate_duplicates	generate_route_from_route_msg (route.c)
eliminate_vehicles	saf_reset_for_workstation (saf.c)
	sbx_connection_reset_msg (sbx.c)
	saf_complete_reset (saf.c)
enable_predicate	enable_predicates (predicates.c)
enable_predicates	adjust_behaviors (cis.c)
endnetgrent	yp_getpwent (yp_getpwent.c)
enemy_disposition	action_drill_needed (predicates.c)
	hasty_attack_needed (predicates.c)
enqueue_on_rcvq	protocol_sim_process (simnet.c)
enqueue_terrain_patch	rotate_queue (cache_queue.c)
	terrain_cache_inquire (cache_query.c)
entity_fill_in_appearance_data	composite_fill_in_appearance_data (composite.c)
	saf_vehicle_fill_in_appearance_data (saf_vehicle.c)
	remote_fill_in_appearance_data (remote.c)
entity_fill_in_position_data	composite_fill_in_position_data (composite.c)
	saf_vehicle_fill_in_position_data (saf_vehicle.c)
	remote_fill_in_position_data (remote.c)
entity_show	composite_show (composite.c)
	saf_vehicle_show (saf_vehicle.c)
	remote_show (remote.c)
exagerate_bridges	generate_route_from_route_msg (route.c)
execute_overlay	cm_execute_overlay (cm.c)

Called Function	Calling Function (filename)
execve	start_simnet (simnet.c)
existnd	__start (../crt1text.s)
	abort_handler (main.c)
	cons (list.c)
	create_bridges (bridges.c)
	create_buildings (trees.c)
	create_contours (trees.c)
	create_lakes (bridges.c)
	create_river_ints (rivers.c)
	create_rivers (rivers.c)
	create_road_ints (roads.c)
	create_roads (roads.c)
	create_tree_canopies (trees.c)
	create_trees (trees.c)
	res_init (res_init.c)
exit	create_river_ints (rivers.c)
	exit_handler (main.c)
	init_terrain_stuff (main.c)
	read_feature (terrain.c)
	read_machine_file (main.c)
	read_quadtree (terrain.c)
	saf_print_help (main.c)
	scheduler_init (maint.c)
	start_simnet (simnet.c)
	yylex (lexer.c)
exit_all_sbx_conns	abort_handler (main.c)
	start_simnet (simnet.c)
	exit_handler (main.c)
exit_handler	init_stuff (main.c)
	main (main.c)
	saf_exit (main.c)
expand_road_route	generate_route_from_route_msg (route.c)
fake_resupply	vehicle_fake_resupply (parser.c)
fclose	AssocReadParams (params.c)
	existnd (res_init.c)
	freopen (freopen.c)
	hostalias (res_query.c)
	reader_read_file (parser.y)
	res_init (res_init.c)
	vis_nextserv (vis.c)
fcntl	load_dom_binding (yp_bind.c)
	openlog (syslog.c)
fflush	_filbuf (filbuf.c)
	buf_rudp_discard_all_buffers (buf_conn.c)
	fclose (flsbuf.c)
	fprintf (fprintf.c)
	fseek (fseek.c)
	init_stuff (main.c)
	init_terrain_stuff (main.c)
	parser_invoke_fcn_on_char (geyer.c)
	read_machine_file (main.c)
	rewind (rew.c)

Called Function	Calling Function (filename)
fflush (cont.)	rudp_discard_all_buffers (rel_conn.c)
	rudp_post_message (rel_conn.c)
	sbx_connection_reset_msg (sbx.c)
	write_string_to_tty (more.c)
fgetpwent	ht_getpwent (ht_getpwent.c)
fgets	AssocReadParams (params.c)
	hostalias (res_query.c)
	parse_servfile (vis.c)
	res_init (res_init.c)
fill	doit (yp_gnetgr.c)
fill_dc_pdu_tags	simnet_send_status_change (DataCollectionPDU.c)
fill_generic_cis_report	change_movement_and_report (cis.c)
	activate_new_cis (cis.c)
fill_in_appearance_data	broadcast_appearance_data (sbx.c)
	broadcast_pae_data (sbx.c)
	sbx_connection_stat_pulse (sbx.c)
fill_in_echelon_data	broadcast_pae_data (sbx.c)
	sbx_connection_stat_pulse (sbx.c)
fill_in_position_data	broadcast_pae_data (sbx.c)
	sbx_connection_wat_pulse (sbx.c)
	sbx_connection_stat_pulse (sbx.c)
fill_ivis_pdu_tags	ivis_send_contact_report (IvisPDU.c)
	ivis_send_shell_report (IvisPDU.c)
	ivis_send_spot_report (IvisPDU.c)
fill_ivis_report_tags	ivis_send_shell_report (IvisPDU.c)
	ivis_send_spot_report (IvisPDU.c)
	ivis_send_contact_report (IvisPDU.c)
fill_sbx_opfor_header	broadcast_appearance_data (sbx.c)
	broadcast_pae_data (sbx.c)
	broadcast_veh_is_gone (sbx.c)
	composite_generate_status_report (composite.c)
	fill_generic_cis_report (cis.c)
	gasp (main.c)
	remote_tick (remote.c)
	saf_vehicle_checkpoint_state (saf_vehicle.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
	sbx_connection_process_ground_impact (sbx.c)
	sbx_connection_process_indirect_fire (sbx.c)
	sbx_connection_process_vehicle_impact (sbx.c)
	sbx_connection_stat_pulse (sbx.c)
	sbx_connection_synch_received (sbx.c)
	sbx_connection_wat_pulse (sbx.c)
	sbx_printf (sbx.c)
	send_contact_report (reporter.c)
	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
	send_stealth_gone_msg (remote.c)
fill_sim_pdu_tags	simnet_send_appearance (SimulationPDU.c)
	simnet_send_collision (SimulationPDU.c)
	simnet_send_deactivate (SimulationPDU.c)
	simnet_send_fire (SimulationPDU.c)

Called Function	Calling Function (filename)
fill_sim_pdu_tags (cont.)	simnet_send_impact (SimulationPDU.c)
	simnet_send_indirect_fire (SimulationPDU.c)
	simnet_send_radiate (SimulationPDU.c)
	simnet_send_resupply_cancel (SimulationPDU.c)
	simnet_send_resupply_received (SimulationPDU.c)
	simnet_send_resupply_request (SimulationPDU.c)
find_a_quad	next_road_point (roads.c)
find_closest_building	buildings_thru (building_check.c)
find_extra_contour_info	create_contours (trees.c)
find_height_on_poly	tdb_get_z (elevation.c)
find_new_host_mask	generate_vehicle_id (id_hash.c)
find_support	tdb_get_z (elevation.c)
find_tag	create_composite (composite.c)
	create_damage (damage.c)
	create_saf_vehicle (saf_vehicle.c)
	create_unit (create.c)
	create_weapon_systems (weapons.c)
	database_if_damage_query (if_damage.c)
	ft_float (misc.c)
	ft_int (misc.c)
	ft_symbol (misc.c)
	ft_table (misc.c)
	ft_untagged_table (misc.c)
	init_mappings (map.c)
	make_create_list (create.c)
	map_echelon_symbol_to_echelon_number (map.c)
	map_echelon_type_symbol_to_icon_number (map.c)
	map_icon_symbol_to_icon_number (map.c)
	read_machine_file (main.c)
	weapon_configure (weapons.c)
find_tag_sorted	database_if_damage_query (if_damage.c)
	database_df_damage_query (df_damage.c)
findbucket	realloc (malloc.c)
finddatum	dbm_fetch (ndbm.c)
fire_missile_at_target	shoot_target (gunner.c)
fire_weapon_at_target	shoot_target (gunner.c)
firestatus_to_string	sbx_set_targeting_parameters (sbx.c)
	targeting_show (targeting.c)
fix_coords	test_maxima (pve_checkvis.c)
flock	_yp_dobind (yp_bind.c)
fly_missile	fly_round (weapons.c)
fly_round	gunner_tick (gunner.c)
fmat_to_rmat	remote_tick (remote.c)
followvehicle	driver_tick (driver.c)
open	AssocReadParams (params.c)
	existnd (res_init.c)
	hostalias (res_query.c)
	reader_read_file (parser.y)
	res_init (res_init.c)
	vis_nextserv (vis.c)

Called Function	Calling Function (filename)
forceID_to_string	sbx_connection_create_unit_msg (sbx.c)
	sbx_connection_show (sbx.c)
	entity_show (entity.c)
fork	start_simnet (simnet.c)
fp_class_d	cvt (doprnt.c)
fprintf	bse (breakset.c)
	buffer__deallocate (queue_fix.c)
	buffer_allocate (queue_fix.c)
	check_hits (pve_checkvis.c)
	clip_to_tdb (pve_checkvis.c)
	clnt_pcreateerror (clnt_perror.c)
	clnt_perror (clnt_perror.c)
	get_args (fncl.c)
	init_stuff (main.c)
	invoke (invoke.c)
	mat_rot_init (mat_r_init.c)
	parser_invoke_fcn_on_char (geyer.c)
	patch (pve_checkvis.c)
	proc_switches (args.c)
	pve_checkvis (pve_checkvis.c)
	rassoc_sym (map.c)
	reader_read_file (parser.y)
	rpc_perror (clnt_perror.c)
	rpc_perror1 (clnt_perror.c)
	rpc_perror2 (clnt_perror.c)
	scheduler_init (maint.c)
	simple_queue_allocate (bufpool.c)
	sort_trim_hits (pve_checkvis.c)
	store_hit (pve_checkvis.c)
	tag_error (tags.c)
	test_maxima (pve_checkvis.c)
	treelines (pve_checkvis.c)
	tty_invoke_fcn_on_char (breakset.c)
	tty_malloc_break_set (breakset.c)
	warn_about_point_space (terrain.c)
	which_event_ring (maint.c)
	yyerror (parser.y)
	yylex (lexer.c)
fragment_route	cm_delete_cm_from_overlay (cm.c)
free	AssocClose (close.c)
	cache_and_file_terminate (cache_alloc.c)
	cfree (calloc.c)
	clntudp_bufcreate (clnt_udp.c)
	clntudp_destroy (clnt_udp.c)
	copy_stack_to_array (parser.y)
	fclose (flsbuf.c)
	free_stack (parser.y)
	heap_deallocate (alloc.c)
	parse_servfile (vis.c)
	realloc (malloc.c)

Called Function	Calling Function (filename)
free (cont.)	setbuf (setbuf.c)
	xdr_bytes (xdr.c)
	xdr_string (xdr.c)
free_function	cancel_fncl (fncl.c)
	invoke (invoke.c)
free_list	buildings_thru (building_check.c)
	free_list (list.c)
	free_water_list (water_check.c)
	get_building_indecies (building_check.c)
	get_water_indecies (water_check.c)
	point_in_building (building_check.c)
free_stack	free_stack (parser.y)
	yyparse (parser.y)
free_water_list	water_check (water_check.c)
freeminuslist	yp_setgrent (yp_getgrent.c)
freeminuslist	yp_setpwent (yp_getpwent.c)
freenode	makenode (yp_match.c)
freopen	tty_setup_modes (setup.c)
fscanf	existnd (res_init.c)
fseek	ht_fgetgrent (ht_getgrent.c)
fstat	_filbuf (filbuf.c)
	_flsbuf (flsbuf.c)
	tdb_get_tdb_info (tdb_init.c)
ft_float	airveh_init (flyingveh.c)
	create_damage (damage.c)
	create_groundveh (groundveh.c)
	create_missile (missile.c)
	create_saf_vehicle (saf_vehicle.c)
	create_targeting (targeting.c)
	create_turret (turret.c)
	weapon_configure (weapons.c)
ft_int	create_saf_vehicle (saf_vehicle.c)
	create_unit (create.c)
	get_guises (saf_vehicle.c)
	lookup_cis_default_speed (cis.c)
	lookup_cis_resumable (cis.c)
	lookup_cis_situational (cis.c)
	weapon_configure (weapons.c)
ft_symbol	create_damage (damage.c)
	create_targeting (targeting.c)
	create_weapon_systems (weapons.c)
	lookup_cis_communication_string (cis.c)
	lookup_cis_formation (cis.c)
	lookup_cis_movement_method (cis.c)
	weapon_configure (weapons.c)
ft_table	create_weapon_systems (weapons.c)
	get_formation_layout (formation.c)
	lookup_cis_communication_string (cis.c)
	lookup_cis_default_speed (cis.c)
	lookup_cis_enabled_predicates (cis.c)

Called Function	Calling Function (filename)
ft_table (cont.)	lookup_cis_formation (cis.c)
	lookup_cis_movement_method (cis.c)
	lookup_cis_resumable (cis.c)
	lookup_cis_situational (cis.c)
ft_untagged_table	create_composite (composite.c)
ftell	ht_fgetgrent (ht_getgrent.c)
ftime	get_millisecond_time (time.c)
	init_clocks (time.c)
fvec_to_rvec	compute_damage_keys (df_damage.c)
generate_a_deactivate	saf_vehicle_go_away (saf_vehicle.c)
generate_in_place_route	cm_execute_overlay (cm.c)
generate_indirect_fire_pkt	sbx_connection_generate_arty_msg (sbx.c)
generate_route_from_route_msg	cm_add_route (cm.c)
generate_status_report	sbx_connection_query_sub_state (sbx.c)
generate_vehicle_id	create_composite (composite.c)
	create_missile (missile.c)
	create_saf_vehicle (saf_vehicle.c)
generate_weapon_hit	impact_weapon (weapons.c)
generate_weapon_miss	impact_weapon (weapons.c)
get_a_leader	commander_attack (commander.c)
	commander_change_speed (commander.c)
	commander_execute_overlay (commander.c)
	commander_face_direction (commander.c)
	commander_follow_vehicle (commander.c)
	commander_goto_point (commander.c)
	commander_halt (commander.c)
	commander_land (commander.c)
	commander_set_mission_direction (commander.c)
	commander_set_mission_speed (commander.c)
	commander_simulator_in_command (commander.c)
get_args	periodic_fncl (fncl.c)
	deferred_fncl (fncl.c)
get_binder_port	_yp_dobind (yp_bind.c)
get_building_indecies	point_in_building (building_check.c)
	buildings_thru (building_check.c)
get_control_measure	cm_add_zone (cm.c)
	cm_add_route (cm.c)
	cm_add_point (cm.c)
	cm_add_line (cm.c)
	cm_add_area (cm.c)
get_device_number	open_cmc (net_open.c)
get_formation_layout	get_station_info (formation.c)
get_grid_number	find_support (elevation.c)
get_guises	create_missile (missile.c)
	create_saf_vehicle (saf_vehicle.c)
get_i_and_t_from_normal	missile_maybe_hit_target (missile.c)
	get_impact_and_trajectory (impact.c)
get_impact_and_trajectory	generate_weapon_hit (weapons.c)
get_leader_state	followvehicle (driver.c)
get_local_address	open_io_connections (sbx.c)
get_locks	open_cmc (net_open.c)

Called Function	Calling Function (filename)
get_me_a_random_fraction	check_prob (random.c)
	create_saf_vehicle (saf_vehicle.c)
	damage_indirect_fire (damage.c)
	damage_vehicle_impact (damage.c)
	damage_vehicle_rammed (damage.c)
	detectable (detection.c)
	generate_weapon_miss (weapons.c)
	randomize_vector (impact.c)
	reporter_tick (reporter.c)
	targeting_tick (targeting.c)
	turret_scan (turret.c)
get_millisecond_time	buf_rudp_open (buf_conn.c)
	buf_rudp_tick (buf_conn.c)
	change_fncl_period (fncl.c)
	create_tickable (tickable.c)
	deferred_fncl (fncl.c)
	init_stuff (main.c)
	invoke_functions_until (sched.c)
	perf_monitor_on (perf.c)
	periodic_fncl (fncl.c)
	report_id_hashing_statistics (id_hash.c)
	rudp_init (rel_conn.c)
	rudp_open (rel_conn.c)
	rudp_show (rel_conn.c)
	sbx_connection_synch_received (sbx.c)
	set_critical_performance_level (perf.c)
	start_collecting_id_hashing_statistics (id_hash.c)
	wait_until (sched.c)
et_next_circle_point	pilot_orbithold (pilot.c)
get_overlay	cm_add_line (cm.c)
	cm_add_route (cm.c)
	cm_add_zone (cm.c)
	cm_add_point (cm.c)
	cm_add_area (cm.c)
get_quad_nodes	get_building_indecies (building_check.c)
	get_water_indecies (water_check.c)
get_quad_nodes_internal	get_quad_nodes_internal (search.c)
	get_quad_nodes (search.c)
get_relative_vehicle_agility	fire_weapon_at_target (weapons.c)
get_sbx_from_port	sbx_connection_send_to_port (sbx.c)
	parser_create_vehicle (parser.c)
get_sender_of_last_packet	protocol_sim_process (simnet.c)
	protocol_stealth_process (simnet.c)
	simnet_tick (simnet.c)
get_soil_type	groundveh_move (groundveh.c)
	remote_tick (remote.c)
get_station_info	composite_assume_formation_internal (composite.c)
get_string	yylex (lexer.c)
get_symbol	cm_add_area (cm.c)
	cm_add_line (cm.c)
	cm_add_point (cm.c)
	cm_add_zone (cm.c)

Called Function	Calling Function (filename)
get_symbol (cont.)	cm_execute_overlay (cm.c)
	create_unit (create.c)
	get_string (symbol.c)
	init_global_symbols (symbols.c)
	parser_create_vehicle (parser.c)
	read_machine_file (main.c)
	sbx_change_formation (sbx.c)
	sbx_connection_create_unit_msg (sbx.c)
	yylex (lexer.c)
get_symbol_value	yylex (lexer.c)
	yyparse (parser.y)
get_turret	get_turret (formation.c)
	get_formation_layout (formation.c)
get_type	net_open (net_open.c)
get_unit_direction	followvehicle (driver.c)
get_water_indecies	water_check (water_check.c)
getanswer	dns_gethostbyname (dns_getnamadr.c)
	dns_gethostbyaddr (dns_getnamadr.c)
getbit	dbm_forder (ndbm.c)
getdomainname	ht_innetgr (ht_innetgr.c)
getdtablesize	_f_morefiles (findiop.c)
	_rpc_dtablesize (rpcdtablesize.c)
getenv	vis_nextserv (vis.c)
	perror (perror.c)
	hostalias (res_query.c)
	timezone (timezone.c)
	res_init (res_init.c)
	tzset (ctime.c)
geteuid	check_binder_up (yp_bind.c)
getfirstfromyellow	yp_getgrent (yp_getgrent.c)
getfirstfromyellow	yp_getpwent (yp_getpwent.c)
getgidfromyellow	matchgid (yp_getgrent.c)
gethostbyname	get_local_address (udp_berkeley.c)
gethostent	interpret (yp_gethostent.c)
gethostname	parser_init (parser.c)
	udp_open (udp_berkeley.c)
	res_init (res_init.c)
	get_local_address (udp_berkeley.c)
getnamefromyellow	yp_getpwent (yp_getpwent.c)
getnamefromyellow	matchname (yp_getgrent.c)
getnetent	interpret (yp_gnetent.c)
getnetgrent	yp_getpwent (yp_getpwent.c)
getnextfromyellow	yp_getgrent (yp_getgrent.c)
getnextfromyellow	yp_getpwent (yp_getpwent.c)
getpagesize	malloc (malloc.c)
getpid	newborn (yp_bind.c)
	abort_handler (main.c)
	syslog (syslog.c)
	bindresvport (bindresvport.c)
	clntudp_bufcreate (clnt_udp.c)
getprotoent	interpret (yp_getpent.c)
getservent	interpret (yp_getservent.c)

Called Function	Calling Function (filename)
getsockname	check_binding (yp_bind.c)
gettimeofday	ftime (ftime.c)
	tzsetkernel (ctime.c)
	clntudp_bufcreate (clnt_udp.c)
	time (time.c)
getuidfromyellow	matchuid (yp_getpwent.c)
gidof	matchgid (yp_getgrent.c)
go_away	eliminate_vehicles (saf.c)
grid_locword	test_maxima (pve_checkvis.c)
ground_impact_to_sbx	protocol_sim_process (simnet.c)
ground_level	pilot_start_landed (pilot.c)
	pilot_start_takingoff (pilot.c)
	air_tick (flyingveh.c)
groundveh_mobility_kill	saf_vehicle_catastrophic_kill (saf_vehicle.c)
	saf_vehicle_mobility_kill (saf_vehicle.c)
groundveh_move	groundveh_tick (groundveh.c)
groundveh_tick	saf_vehicle_tick (saf_vehicle.c)
grskip	ht_fgetgrent (ht_getgrent.c)
grskip	interpret (yp_getgrent.c)
gtty	tty_setup_modes (setup.c)
gunner_round_flying	pilot_hoverattack (pilot.c)
	saf_vehicle_tick (saf_vehicle.c)
gunner_tick	targeting_tick (targeting.c)
hasty_attack_needed	check_predicates (predicates.c)
heading_to_compass	send_spot_report (reporter.c)
	send_contact_report (reporter.c)
	show_vehicle_cluster (reporter.c)
heap_allocate	_queue_hanger_hang_this (queue_fix.c)
	add_executing_unit (cm.c)
	add_removed_vehicle (reporter.c)
	add_vehicle_to_cluster (reporter.c)
	allocate_bitfield (bitfield.c)
	allocate_points (terrain.c)
	allocate_safobj (safobj.c)
	buf_rudp_init (buf_conn.c)
	buffer_allocate (queue_fix.c)
	buffer_statistics_init (queue_fix.c)
	cm_execute_overlay (cm.c)
	cm_get_cm_from_overlay (cm.c)
	cm_recompute_cm_from_overlay (cm.c)
	cons (list.c)
	create_airveh (flyingveh.c)
	create_bridges (bridges.c)
	create_buildings (trees.c)
	create_cis (cis.c)
	create_collision (collision.c)
	create_commander (commander.c)
	create_composite (composite.c)
	create_contours (trees.c)
	create_damage (damage.c)
	create_detection (detection.c)

Called Function	Calling Function (filename)
heap_allocate (cont.)	create_driver (driver.c)
	create_entity (entity.c)
	create_groundveh (groundveh.c)
	create_lakes (bridges.c)
	create_logistics (logistics.c)
	create_missile (missile.c)
	create_navigator (navigator.c)
	create_new_cluster (reporter.c)
	create_new_shelling (reporter.c)
	create_pilot (pilot.c)
	create_predicates (predicates.c)
	create_remote (remote.c)
	create_reporter (reporter.c)
	create_river_ints (rivers.c)
	create_rivers (rivers.c)
	create_road_ints (roads.c)
	create_roads (roads.c)
	create_routepoint (route.c)
	create_saf_vehicle (saf_vehicle.c)
	create_spotter (spotter.c)
	create_targeting (targeting.c)
	create_tickable (tickable.c)
	create_tree_canopies (trees.c)
	create_trees (trees.c)
	create_turret (turret.c)
	create_vehicle (vehicle.c)
	create_weapon_systems (weapons.c)
	deferred_fncl (fncl.c)
	generate_in_place_route (route.c)
	generate_route_from_route_msg (route.c)
	get_args (fncl.c)
	get_control_measure (cm.c)
	get_overlay (cm.c)
	init_grid_tables (iterator.c)
	logistics_supply_offer_received (logistics.c)
	open_io_connections (sbx.c)
	periodic_fncl (fncl.c)
	read_feature (terrain.c)
	read_quadtree (terrain.c)
	remote_next_road_point (remote.c)
	rudp_init (rel_conn.c)
	sbx_connection_init (sbx.c)
	scheduler_init (maint.c)
	simple_queue_allocate (bufpool.c)
heap_calloc	Alloc (par_unix.c)
	tty_malloc_break_set (breakset.c)
	udp_open (udp_berkeley.c)
heap_create	init_stuff (main.c)
heap_deallocate	_queue_hanger_unhang (queue_fix.c)
	buffer__deallocate (queue_fix.c)
	composite_go_away (composite.c)
	deallocate_bitfield (bitfield.c)

Called Function	Calling Function (filename)
heap_deallocate (cont.)	deallocate_safobj (safobj.c)
	destroy_airveh (flyingveh.c)
	destroy_area_cm (cm.c)
	destroy_cis (cis.c)
	destroy_cm (cm.c)
	destroy_cm_list_item (cm.c)
	destroy_collision (collision.c)
	destroy_damage (damage.c)
	destroy_detection (detection.c)
	destroy_driver (driver.c)
	destroy_entity (entity.c)
	destroy_groundveh (groundveh.c)
	destroy_line_cm (cm.c)
	destroy_logistics (logistics.c)
	destroy_missile (missile.c)
	destroy_overlay (cm.c)
	destroy_overlay_list_item (cm.c)
	destroy_pilot (pilot.c)
	destroy_point_cm (cm.c)
	destroy_predicates (predicates.c)
	destroy_rem_veh_id_item (reporter.c)
	destroy_reporter (reporter.c)
	destroy_route (route.c)
	destroy_routepoint (route.c)
	destroy_shell_summary (reporter.c)
	destroy_spotter (spotter.c)
	destroy_targeting (targeting.c)
	destroy_tickable (tickable.c)
	destroy_turret (turret.c)
	destroy_veh_id_item (reporter.c)
	destroy_vehicle (vehicle.c)
	destroy_vehicle_cluster (reporter.c)
	destroy_weapon_systems (weapons.c)
	destroy_zone_cm (cm.c)
	free_function (maint.c)
	free_list (list.c)
	logistics_tick (logistics.c)
	queue_flush_hangers (queue_fix.c)
	remote_go_away (remote.c)
	remove_dead_clustered_vehicles (reporter.c)
	remove_duplicates (list.c)
	remove_executing_unit (cm.c)
	remove_shelling_cluster (reporter.c)
	saf_vehicle_go_away (saf_vehicle.c)
	simple_queue_deallocate (bufpool.c)
	splice_out_vehicle_cluster (reporter.c)
	udp_close (udp_berkeley.c)
heap_destroy	abort_handler (main.c)
	exit_handler (main.c)
helo_tick	air_tick (flyingveh.c)
host_local_vehicle_predicate	simnet_send_collision (SimulationPDU.c)
	simnet_send_impact (SimulationPDU.c)

Called Function	Calling Function (filename)
host_loopback_function	ivis_send_contact_report (IvisPDU.c)
	ivis_send_shell_report (IvisPDU.c)
	ivis_send_spot_report (IvisPDU.c)
	simnet_send_collision (SimulationPDU.c)
	simnet_send_fire (SimulationPDU.c)
	simnet_send_impact (SimulationPDU.c)
	simnet_send_indirect_fire (SimulationPDU.c)
hostalias	res_search (res_query.c)
ht_endgrent	ht_getgrgid (ht_getgrgid.c)
ht_endhostent	_endhtent (ht_getnamadr.c)
ht_endnetent	ht_getnetbyaddr (ht_gnbyaddr.c)
ht_endnetgrent	ht_innetgr (ht_innetgr.c)
ht_endprotoent	ht_getprotobynumber (ht_getproto.c)
ht_endpwent	ht_getpwnam (ht_getpwent.c)
ht_endrpcent	ht_getrpcbynumber (ht_grpcent.c)
ht_endservent	ht_getservbyname (ht_gservent.c)
ht_fgetgrent	ht_getgrent (ht_getgrent.c)
ht_getgrent	ht_getgrgid (ht_getgrgid.c)
ht_gethostent	_host_interpret (ht_gethent.c)
ht_getnetent	ht_getnetbyaddr (ht_gnbyaddr.c)
ht_getnetgrent	ht_innetgr (ht_innetgr.c)
ht_getprotoent	ht_getprotobynumber (ht_getproto.c)
ht_getpwent	ht_getpwnam (ht_getpwent.c)
ht_getrpcent	ht_getrpcbynumber (ht_grpcent.c)
ht_getservent	ht_getservbyname (ht_gservent.c)
ht_setgrent	ht_getgrgid (ht_getgrgid.c)
ht_sethostent	_sethtent (ht_getnamadr.c)
ht_setnetent	ht_getnetbyaddr (ht_gnbyaddr.c)
ht_setnetgrent	ht_innetgr (ht_innetgr.c)
ht_setprotoent	ht_getprotobynumber (ht_getproto.c)
ht_setpwent	ht_getpwnam (ht_getpwent.c)
ht_setrpcent	ht_getrpcbynumber (ht_grpcent.c)
ht_setservent	ht_getservbyname (ht_gservent.c)
htons	ht_getservent (ht_gservent.c)
hull_to_world_from_direction	fly_missile (missile.c)
	fire_missile_at_target (missile.c)
hull_to_world_from_orientation	plane_tick (plane.c)
	helo_tick (helo.c)
icon_from_object_type	remote_tick (remote.c)
	create_saf_vehicle (saf_vehicle.c)
identify_version	main (main.c)
idle_tick	pilot_tick (pilot.c)
impact_weapon	fly_round (weapons.c)
index	parse_servfile (vis.c)
	res_init (res_init.c)
	res_mkquery (res_mkquery.c)
	strtok (strtok.c)
	syslog (syslog.c)
	timezone (timezone.c)
indirect_fire_to_sbx	protocol_sim_process (simnet.c)
induce_roll	plane_tick (plane.c)
induce_tail_spin	plane_tick (plane.c)

Called Function	Calling Function (filename)
inet_addr	interpret (yp_gethostent.c)
	sockaddr_init (udp_berkeley.c)
	res_init (res_init.c)
inet_makeaddr	nettoa (yp_gnetent.c)
inet_network	interpret (yp_gnetent.c)
inet_ntoa	get_local_address (udp_berkeley.c)
init_cache_map	cache_init (cache_alloc.c)
init_clocks	init_stuff (main.c)
init_global_symbols	init_stuff (main.c)
init_grid_entry_list	create_remote (remote.c)
	remote_init_vars (remote.c)
	create_saf_vehicle (saf_vehicle.c)
init_grid_tables	init_terrain_stuff (main.c)
init_mappings	database_init (main.c)
init_object_and_texture_names	tdb_init_memory (memory.c)
	tdb_init_cache (tdb_init.c)
init_patch_indices	memory_init (memory.c)
	cache_init (cache_alloc.c)
init_pilot_state_machine	create_pilot (pilot.c)
nit_saf_to_simnet_id_table	init_stuff (main.c)
init_safobj_table	init_stuff (main.c)
init_static_matricies	init_stuff (main.c)
init_stuff	main (main.c)
init_symbol_table	init_stuff (main.c)
init_target_items	update_target_list (targeting.c)
init_target_list	update_target_list (targeting.c)
	create_targeting (targeting.c)
init_terrain_cache	cache_init (cache_alloc.c)
init_terrain_stuff	init_stuff (main.c)
initialize_weapon_priority_list	create_weapon_systems (weapons.c)
nlist	lookup (yp_innetgr.c)
innetgr	yp_getpwent (yp_getpwent.c)
insert_function	periodic_fncl (fncl.c)
	change_fncl_period (fncl.c)
	deferred_fncl (fncl.c)
insert_grid_entry	init_grid_entry_list (iterator.c)
	update_grid_entry_list (iterator.c)
insert_periodic_function	invoke_functions_until (sched.c)
int_dist	closest_leader (formation.c)
	closest_to_a_leader (formation.c)
interesting_packet_test	host_loopback_function (simnet.c)
	simnet_tick (simnet.c)
interior_angle_between_vectors	angle_between_vectors (math.c)
	choose_skirt_point (collision.c)
	detectable (detection.c)
	next_road_point (roads.c)
	start_avoiding_collision (collision.c)
	start_disengaging (collision.c)
interpolate_curve	fire_weapon_at_target (weapons.c)
	database_detection_query (detection.c)

Called Function	Calling Function (filename)
interpret	ht_getrpcent (ht_grpcent.c)
	yp_getgrnam (yp_getgrent.c)
	yp_gethostbyname (yp_gethostent.c)
	yp_getnetbyaddr (yp_gnetent.c)
	yp_getprotobynumber (yp_getpent.c)
	yp_getpwuid (yp_getpwent.c)
	yp_getrpcbynumber (yp_getrpcent.c)
	yp_getservbyport (yp_getservent.c)
interpretwithsave	yp_getgrent (yp_getgrent.c)
	yp_getpwent (yp_getpwent.c)
intervis_can_see_pt_to_pt	choose_target_and_weapon (targeting.c)
	fly_missile (missile.c)
	shoot_target (gunner.c)
	spotter_tick (spotter.c)
	track_target (gunner.c)
intervis_get_high_ground	change_movement_and_report (cis.c)
intervis_get_view	fire_weapon_at_target (weapons.c)
intervis_possibly_visible	detection_tick (detection.c)
intervis_pt_to_pt	intervis_can_see_pt_to_pt (intervis.c)
	intervis_get_view (intervis.c)
invoke	invoke_functions_until (sched.c)
nvoke_functions_until	invoke_functions_until (sched.c)
	main (main.c)
ioctl	access_cmc (net_acce.c)
	clntudp_bufcreate (clnt_udp.c)
	get_type (net_open.c)
	gtty (gtty.c)
	isatty (isatty.c)
	make_me_non_blocking (udp_berkeley.c)
	stty (stty.c)
	tty_restore_modes (setup.c)
	tty_setup_modes (setup.c)
is_cm_executed_in_list	cm_recompute_cm_from_overlay (cm.c)
is_probably_a_string	tag_error (tags.c)
satty	_flsbuf (flsbuf.c)
issue_reports	reporter_tick (reporter.c)
ivis_send_contact_report	send_contact_report (reporter.c)
ivis_send_shell_report	send_shelling_report (reporter.c)
ivis_send_spot_report	send_spot_report (reporter.c)
kill	existnd (res_init.c)
left_bottom_region	top_left_corner (minimum_clip.c)
left_column	minimum_clip (minimum_clip.c)
eft_edge	center_column (minimum_clip.c)
	left_column (minimum_clip.c)
line_intersection	cm_count_intersections (navigator.c)
	cm_intersection (navigator.c)
	cm_poly_intersection (navigator.c)
	count_intersections (intersection.c)
	seg_intersection (intersection.c)
line_intersection_core	line_intersection (intersection.c)
	open_line_intersection (intersection.c)
load_dom_binding	_yp_dobind (yp_bind.c)

Called Function	Calling Function (filename)
loader_tick	targeting_tick (targeting.c)
localtime	ctime (ctime.c)
logistics_remove_vehicles	saf_vehicle_remove_vehicles (saf_vehicle.c)
logistics_supply_offer_canceled	saf_vehicle_pro_sim (saf_vehicle.c)
logistics_supply_offer_received	saf_vehicle_pro_sim (saf_vehicle.c)
logistics_tick	saf_vehicle_tick (saf_vehicle.c)
lookup	yp_innetgr (yp_innetgr.c)
lookup_cis_communication_string	fill_generic_cis_report (cis.c)
lookup_cis_default_speed	check_predicates (predicates.c)
ookup_cis_enabled_predicates	adjust_behaviors (cis.c)
lookup_cis_formation	adjust_behaviors (cis.c)
lookup_cis_movement_method	adjust_behaviors (cis.c)
lookup_cis_resumable	activate_new_cis (cis.c)
	pop_resumable_cis (cis.c)
lookup_cis_situational	change_movement_and_report (cis.c)
lookup_vehicle_with_range_check	parser_set_targeting_parameters (parser.c)
	show_vehicle (parser.c)
	stealth_attach_to (parser.c)
	vehicle_defuel (parser.c)
	vehicle_fake_resupply (parser.c)
	vehicle_resupply (parser.c)
lseek	fopen (fopen.c)
	freopen (freopen.c)
	fseek (fseek.c)
	ftell (ftell.c)
	init_patch_indices (cache_init.c)
	memory_init (memory.c)
	rewind (rew.c)
	tdb_init_patch_guards (cache_init.c)
	terrain_cache_inquire (cache_query.c)
main	__start (../crt1text.s)
major_detection_increase	targeting_tick (targeting.c)
major_spotter_increase	targeting_tick (targeting.c)
makdatum	dbm_fetch (ndbm.c)
make_array	yyparse (parser.y)
ake_create_list	make_create_list (create.c)
	create_unit (create.c)
make_leader	compute_who_follows_whom (formation.c)
make_me_non_blocking	udp_open (udp_berkeley.c)
make_new_hash_entry	saf_id_from_simnet_id (id_hash.c)
	generate_vehicle_id (id_hash.c)
makekey	lookup (yp_innetgr.c)
makenode	yp_match (yp_match.c)
malloc	_buf (clnt_perror.c)
	_filbuf (filbuf.c)
	_flsbuf (flsbuf.c)
	AssocCreateFreeList (free_list.c)
	AssocGrowFreeList (free_list.c)
	AssocOpen (open.c)
	cache_init (cache_alloc.c)
	calloc (calloc.c)
	clntudp_bufcreate (clnt_udp.c)

Called Function	Calling Function (filename)
malloc (cont.)	get_symbol (symbol.c)
	heap_allocate (alloc.c)
	make_array (parser.y)
	map_buffers (net_stuf.c)
	memory_init (memory.c)
	parse_servfile (vis.c)
	realloc (malloc.c)
	stack_push (parser.y)
	tdb_init_patch_guards (cache_init.c)
	xdr_bytes (xdr.c)
	xdr_string (xdr.c)
map_buffers	open_cmc (net_open.c)
map_echelon_number_to_echelon_symbol	sbx_connection_create_unit_msg (sbx.c)
map_echelon_symbol_to_echelon_number	init_mappings (map.c)
	create_composite (composite.c)
map_echelon_type_number_to_echelon_type_symbol	
	sbx_connection_create_unit_msg (sbx.c)
map_echelon_type_symbol_to_icon_number	create_composite (composite.c)
map_enp	open_cmc (net_open.c)
map_icon_symbol_to_icon_number	icon_from_object_type (map.c)
	map_echelon_type_symbol_to_icon_number (map.c)
map_key	tty_setup_parser_break_set (geyer.c)
ap_role_sym_to_role_number	create_targeting (targeting.c)
mat_copy	get_i_and_t_from_normal (impact.c)
	saf_vehicle_tick (saf_vehicle.c)
mat_ident_init	create_vehicle (vehicle.c)
mat_init	create_missile (missile.c)
mat_mat_mul	create_groundveh (groundveh.c)
	get_i_and_t_from_normal (impact.c)
	groundveh_tick (groundveh.c)
	induce_roll (flyingveh.c)
	induce_tail_spin (flyingveh.c)
	tdb_get_hull_to_world (h_to_w.c)
mat_rot_init	create_groundveh (groundveh.c)
	get_i_and_t_from_normal (impact.c)
	groundveh_move (groundveh.c)
	init_static_matricies (flyingveh.c)
	pilot_start_landed (pilot.c)
	saf_vehicle_teleport (saf_vehicle.c)
	tdb_get_hull_to_world (h_to_w.c)
	vec_z_rotate (math.c)
mat_transpose	tdb_get_hull_to_world (h_to_w.c)
at_vec_mul	get_i_and_t_from_normal (impact.c)
	get_i_and_t_from_normal (impact.c)
	turret_point_at_target (turret.c)
match	doit (yp_gnetgr.c)
matchgid	yp_getgrgid (yp_getgrent.c)
matchname	yp_getgrnam (yp_getgrent.c)
matchname	yp_getpwnam (yp_getpwent.c)
matchuid	yp_getpwuid (yp_getpwent.c)
memchr	fgetpwent (ht_getpwent.c)
memcpy	realloc (malloc.c)

Called Function	Calling Function (filename)
memory_init	tdb_init_memory (memory.c)
minimum_clip	point_in_building (building_check.c)
	buildings_thru (building_check.c)
	possible_intersection (intersection.c)
missile_deactivate	missile_ground_impact (missile.c)
	missile_maybe_hit_target (missile.c)
missile_ground_impact	fly_missile (missile.c)
missile_maybe_hit_target	fly_missile (missile.c)
missile_send_appearance	fly_missile (missile.c)
	fire_missile_at_target (missile.c)
missile_set_desired_direction	fly_missile (missile.c)
missile_show	weapon_show (weapons.c)
missile_state_to_string	missile_show (missile.c)
mission_stationpoint	saf_vehicle_teleport_to_station (saf_vehicle.c)
mmap	map_buffers (net_stuf.c)
morecore	malloc (malloc.c)
munmap	unmap_buffers (net_stuf.c)
muzzle_position_in_world_coordinates	fire_weapon_at_target (weapons.c)
	fire_missile_at_target (missile.c)
navigator_entered_area	navigator_tick (navigator.c)
navigator_entered_zone	navigator_tick (navigator.c)
navigator_left_area	navigator_tick (navigator.c)
navigator_left_zone	navigator_tick (navigator.c)
navigator_passed_line	navigator_tick (navigator.c)
navigator_passed_point	navigator_tick (navigator.c)
navigator_reset_overlay	composite_readjust_overlay (composite.c)
navigator_set_overlay	composite_execute_overlay (composite.c)
navigator_show	composite_show (composite.c)
navigator_tick	composite_tick (composite.c)
need_decluster	update_clusters (reporter.c)
net_access	do_mode_cmd_cmc (net_mode.c)
	net_add_mca (net_mca.c)
	net_add_type (net_type.c)
	net_alive (net_ctl.c)
	net_del_mca (net_mca.c)
	net_get_parameters (net_open.c)
	net_get_statistics (net_stat.c)
	net_init_mca (net_mca.c)
	net_init_type (net_type.c)
	net_stop (net_run.c)
	net_zero_statistics (net_stat.c)
	open_cmc (net_open.c)
	send_cmc (net_osend.c)
	send_cmc_8023 (net_send.c)
net_add_mca	AssocSubscribeWithMask (subscribe.c)
net_add_type	AssocOpen (open.c)
net_addr_compare	AssocSendAggregate (aggregate.c)
net_alive	open_cmc (net_open.c)
	AssocOpen (open.c)
net_close	AssocClose (close.c)

Called Function	Calling Function (filename)
net_current_time	AssocAddTransaction (origin.c)
	AssocCacheResponse (respondent.c)
	AssocSendAggregate (aggregate.c)
	AssocTickAssocLayer (tick.c)
	AssocTimeOutOldResponses (respondent.c)
	UpdateTransactions (tick.c)
net_del_mca	AssocUnsubscribeWithMask (subscribe.c)
net_get_parameters	open_cmc (net_open.c)
net_get_rcv	AssocReceivePDU (receive.c)
net_get_rcv_from_addr	AssocGetLastAddress (who.c)
net_get_rcv_to_addr	AssocProcessRequestPDU (proc_req.c)
net_get_statistics	simnet_getstats (simnet.c)
net_init_mca	AssocOpen (open.c)
net_init_type	AssocOpen (open.c)
net_norm	AssocOpen (open.c)
net_open	AssocOpen (open.c)
net_release_rcv	AssocReceivePDU (receive.c)
et_send	SAF_NET_SND_KLUDGE (simnet.c)
net_set_snd_from_addr	AssocOpen (open.c)
net_set_snd_type	AssocOpen (open.c)
net_snd	SAF_NET_SND_KLUDGE (simnet.c)
net_stat_string	simnet_getstats (simnet.c)
net_stop	AssocOpen (open.c)
net_zero_statistics	simnet_zerostats (simnet.c)
nettoa	yp_getnetbyaddr (yp_gnetent.c)
newborn	_yp_dobind (yp_bind.c)
next_road_point	remote_next_road_point (remote.c)
next_serv	parse_servfile (vis.c)
noa_damp	z_velocity (pilot.c)
object_include	tdb_get_z (elevation.c)
objects	patch (pve_checkvis.c)
obstacle	objects (pve_checkvis.c)
offtime	localtime (ctime.c)
	gmtime (ctime.c)
onminuslist	yp_getgrent (yp_getgrent.c)
onminuslist	yp_getpwent (yp_getpwent.c)
open	access_cmc (net_acce.c)
	create_bridges (bridges.c)
	create_buildings (trees.c)
	create_contours (trees.c)
	create_lakes (bridges.c)
	create_river_ints (rivers.c)
	create_rivers (rivers.c)
	create_road_ints (roads.c)
	create_roads (roads.c)
	create_tree_canopies (trees.c)
	create_trees (trees.c)
	fopen (fopen.c)
	freopen (freopen.c)
	get_type (net_open.c)

Called Function	Calling Function (filename)
open (cont.)	read_quadtree (terrain.c)
	syslog (syslog.c)
	tdb_get_db_format (version.c)
	tdb_get_tdb_info (tdb_init.c)
	tzload (ctime.c)
open_147	net_open (net_open.c)
open_cmc	net_open (net_open.c)
open_io_connections	read_machine_file (main.c)
open_line_intersection	cm_count_intersections (navigator.c)
	count_intersections (intersection.c)
openlog	syslog (syslog.c)
or_bitfield	add_into_spotter_table (spotter.c)
	sbx_set_specific_known_vehicles (sbx.c)
	sbx_set_top_level_known_vehicles (sbx.c)
p2_bottom	left_edge (minimum_clip.c)
p_follow_form_flip	p_follower_new_rel_rt_pnt (p_follower.c)
p_follower_am_i_flying_coord	p_follower_set_desired_vel (p_follower.c)
	p_follower_followroute_tick (p_follower.c)
p_follower_arrive_at_same_time	p_follower_set_desired_vel (p_follower.c)
p_follower_comp_reset_route	cm_add_route (cm.c)
p_follower_find_pnt_in_turn	p_follower_new_rel_rt_pnt (p_follower.c)
p_follower_flip_in_turn	pilot_followroute_tick (pilot.c)
p_follower_fly_in_coord_pos	p_follower_hoverhold (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
	p_follower_landhold_tick (p_follower.c)
	p_follower_set_desired_vel (p_follower.c)
p_follower_flying_independent	p_follower_followroute_tick (p_follower.c)
p_follower_followroute_tick	pilot_follow_leader_tick (p_follower.c)
p_follower_gen_coord_goal_pnt	p_follower_flying_independent (p_follower.c)
p_follower_get_wrld_offset	commander_attack (commander.c)
	p_follower_fly_in_coord_pos (p_follower.c)
	p_follower_gen_coord_goal_pnt (p_follower.c)
	p_follower_hoverhold (p_follower.c)
	p_follower_land (p_follower.c)
	p_follower_pnt_in_no_turn (p_follower.c)
p_follower_hoverattack_tick	pilot_follow_leader_tick (p_follower.c)
p_follower_hoverhold	p_follower_find_pnt_in_turn (p_follower.c)
	p_follower_followroute_tick (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
p_follower_hoverhold_tick	pilot_follow_leader_tick (p_follower.c)
p_follower_land	p_follower_landhold_tick (p_follower.c)
p_follower_landhold_tick	pilot_follow_leader_tick (p_follower.c)
p_follower_leader_passed_point	pilot_gotopoint (pilot.c)
p_follower_new_rel_rt_pnt	p_follower_flying_independent (p_follower.c)
	p_follower_followroute_tick (p_follower.c)
p_follower_passed_point	p_follower_flying_independent (p_follower.c)
	p_follower_followroute_tick (p_follower.c)
_follower_pnt_in_hard_turn	p_follower_find_pnt_in_turn (p_follower.c)
p_follower_pnt_in_no_turn	p_follower_find_pnt_in_turn (p_follower.c)
p_follower_pnt_in_shallow_turn	p_follower_find_pnt_in_turn (p_follower.c)
p_follower_set_desired_vel	p_follower_followroute_tick (p_follower.c)
p_follower_set_follow	saf_vehicle_reset_station_keeper (saf_vehicle.c)

Called Function	Calling Function (filename)
p_follower_stop_coord	p_follower_gen_coord_goal_pnt (p_follower.c)
	p_follower_followroute_tick (p_follower.c)
p_poly_provides_support	find_support (elevation.c)
parse_servfile	vis_nextserv (vis.c)
parser_create	main (main.c)
parser_init	parser_create (parser.c)
parser_restore_term	abort_handler (main.c)
	exit_handler (main.c)
patch	test_maxima (pve_checkvis.c)
perf_monitor_off	set_monitor_period (parser.c)
perf_monitor_on	set_monitor_period (parser.c)
periodic_fncl	open_io_connections (sbx.c)
	parser_create (parser.c)
	start_simnet (simnet.c)
	start_ticking (tickable.c)
perror	map_buffers (net_stuf.c)
	tty_restore_modes (setup.c)
	tty_setup_modes (setup.c)
	unmap_buffers (net_stuf.c)
pflush	perror (perror.c)
	print_chr (perror.c)
	print_str (perror.c)
phase_two_collision_check	detect_imminent_collision_tick (collision.c)
pilot_cancel_immediate	saf_vehicle_resume_mission (saf_vehicle.c)
	pilot_set_route_mis (pilot.c)
pilot_change_altitude_im	commander_change_altitude (commander.c)
	sbx_altitude (sbx.c)
pilot_change_speed_im	saf_vehicle_change_speed (saf_vehicle.c)
pilot_check_state	pilot_tick (pilot.c)
pilot_execute_overlay	saf_vehicle_execute_overlay (saf_vehicle.c)
pilot_executing_immediate_command	saf_vehicle_useless (saf_vehicle.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
pilot_face_direction	pilot_face_direction_im (pilot.c)
pilot_face_direction_im	saf_vehicle_face_direction (saf_vehicle.c)
pilot_follow_leader	saf_vehicle_set_leader (saf_vehicle.c)
pilot_follow_leader_tick	pilot_tick2 (pilot.c)
pilot_follow_vehicle	pilot_follow_vehicle_im (pilot.c)
pilot_follow_vehicle_im	saf_vehicle_follow_vehicle (saf_vehicle.c)
pilot_followroute_tick	pilot_tick2 (pilot.c)
pilot_get_asm	pilot_follow_leader_tick (p_follower.c)
	commander_attack (commander.c)
pilot_get_speed	attackatwill_tick (pilot.c)
	idle_tick (pilot.c)
	p_follower_arrive_at_same_time (p_follower.c)
	p_follower_flying_independent (p_follower.c)
	p_follower_gen_coord_goal_pnt (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
	p_follower_landhold_tick (p_follower.c)
	p_follower_leader_passed_point (p_follower.c)
	p_follower_passed_point (p_follower.c)

Called Function	Calling Function (filename)
pilot_get_speed (cont.)	p_follower_set_desired_vel (p_follower.c)
	pilot_followroute_tick (pilot.c)
	pilot_hoverattack_tick (pilot.c)
	pilot_hoverhold_tick (pilot.c)
	pilot_landhold_tick (pilot.c)
	pilot_orbit_tick (pilot.c)
pilot_goto_endpoint	pilot_hoverattack_tick (pilot.c)
	pilot_hoverhold_tick (pilot.c)
	pilot_landhold_tick (pilot.c)
pilot_goto_point_im	saf_vehicle_goto_point (saf_vehicle.c)
pilot_gotopoint	pilot_hoverattack_tick (pilot.c)
	pilot_orbit_tick (pilot.c)
	pilot_followroute_tick (pilot.c)
pilot_hold	pilot_hold_im (pilot.c)
pilot_hold_im	commander_halt (commander.c)
	sbx_hold (sbx.c)
pilot_hoverattack	pilot_hoverattack_tick (pilot.c)
pilot_hoverattack_complete	pilot_start_hoverattack_complete (pilot.c)
	pilot_hoverattack_tick (pilot.c)
pilot_hoverattack_im	commander_attack (commander.c)
	sbx_attack (sbx.c)
pilot_hoverattack_tick	pilot_tick2 (pilot.c)
pilot_hoverhold	pilot_start_hoverhold (pilot.c)
	pilot_followroute_tick (pilot.c)
	pilot_hoverhold_tick (pilot.c)
pilot_hoverhold_tick	pilot_tick2 (pilot.c)
pilot_init_followroute	pilot_set_route_mis (pilot.c)
pilot_init_hoverattack	pilot_hoverattack_im (pilot.c)
pilot_init_hoverhold	p_follower_hoverhold (p_follower.c)
	pilot_face_direction (pilot.c)
	pilot_followroute_tick (pilot.c)
	pilot_goto_point_im (pilot.c)
	pilot_hold (pilot.c)
pilot_init_land	are_we_there (pilot.c)
	p_follower_land (p_follower.c)
	pilot_land_im (pilot.c)
pilot_init_orbithold	pilot_hold (pilot.c)
pilot_is_facing_direction	z_velocity (pilot.c)
	pilot_hoverhold (pilot.c)
	vehicle_facing_point (pilot.c)
pilot_land_im	commander_land (commander.c)
	sbx_land (sbx.c)
pilot_landhold_tick	pilot_tick2 (pilot.c)
pilot_landing	idle_tick (pilot.c)
	pilot_landhold_tick (pilot.c)
pilot_on_same_route	p_follower_followroute_tick (p_follower.c)
pilot_orbit_tick	pilot_follow_leader_tick (p_follower.c)
	pilot_tick2 (pilot.c)
pilot_orbithold	pilot_orbit_tick (pilot.c)
pilot_point_at_target	point_weapon_at_target (gunner.c)
pilot_remove_vehicles	saf_vehicle_remove_vehicles (saf_vehicle.c)
pilot_set_leader_mis	saf_vehicle_set_leader (saf_vehicle.c)

Called Function	Calling Function (filename)
pilot_set_route_mis	saf_vehicle_set_route (saf_vehicle.c)
	pilot_execute_overlay (pilot.c)
pilot_set_speed_mis	saf_vehicle_set_speed (saf_vehicle.c)
pilot_show	saf_vehicle_show (saf_vehicle.c)
pilot_show_machine	pilot_show (pilot.c)
pilot_start_gotopoint	cm_add_route (cm.c)
	pilot_followroute_tick (pilot.c)
	pilot_hoverattack_tick (pilot.c)
	pilot_hoverhold_tick (pilot.c)
	pilot_init_followroute (pilot.c)
	pilot_init_hoverattack (pilot.c)
	pilot_init_hoverhold (pilot.c)
	pilot_init_land (pilot.c)
	pilot_init_orbithold (pilot.c)
	pilot_landhold_tick (pilot.c)
	pilot_orbit_tick (pilot.c)
	pilot_start_hoverattack_approach (pilot.c)
	pilot_start_hoverattack_egress (pilot.c)
	pilot_start_takingoff (pilot.c)
pilot_start_hoverattack	pilot_hoverattack_tick (pilot.c)
pilot_start_hoverattack_approach	pilot_hoverattack_tick (pilot.c)
pilot_start_hoverattack_complete	pilot_hoverattack_tick (pilot.c)
pilot_start_hoverattack_egress	pilot_hoverattack_tick (pilot.c)
pilot_start_hoverhold	pilot_init_hoverhold (pilot.c)
	pilot_hoverhold_tick (pilot.c)
pilot_start_landed	pilot_start_landing (pilot.c)
	idle_tick (pilot.c)
	pilot_landhold_tick (pilot.c)
pilot_start_landing	idle_tick (pilot.c)
	pilot_landhold_tick (pilot.c)
pilot_start_orbithold	pilot_orbit_tick (pilot.c)
pilot_start_takingoff	p_follower_followroute_tick (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
	p_follower_landhold_tick (p_follower.c)
	pilot_followroute_tick (pilot.c)
	pilot_hoverattack_tick (pilot.c)
	pilot_hoverhold_tick (pilot.c)
	pilot_landhold_tick (pilot.c)
	pilot_orbit_tick (pilot.c)
pilot_state_to_string	pilot_show_machine (pilot.c)
ilot_stationpoint	saf_vehicle_teleport_to_station (saf_vehicle.c)
pilot_stop_mission	saf_vehicle_cancel_overlay (saf_vehicle.c)
pilot_takingoff	p_follower_followroute_tick (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
	p_follower_landhold_tick (p_follower.c)
	pilot_followroute_tick (pilot.c)
	pilot_hoverattack_tick (pilot.c)
	pilot_hoverhold_tick (pilot.c)
	pilot_landhold_tick (pilot.c)
	pilot_orbit_tick (pilot.c)
pilot_tick	saf_vehicle_tick (saf_vehicle.c)
pilot_tick2	pilot_tick (pilot.c)

Called Function	Calling Function (filename)
plane_tick	air_tick (flyingveh.c)
pmap_getport	clntudp_bufcreate (clnt_udp.c)
point_in_building	detect_collision_tick (collision.c)
point_inside_polygon	segment_inside_polygon (intersection.c)
point_weapon_at_target	track_target (gunner.c)
	acquire_target (gunner.c)
poll_request_to_string	sbx_connection_service_poll_msg (sbx.c)
	sbx_connection_show (sbx.c)
polygon_include	p_poly_provides_support (elevation.c)
pop_cis	check_predicates (predicates.c)
pop_resumable_cis	commander_resume_mission (commander.c)
possible_intersection	segment_thru_lake (water_check.c)
	segment_thru_river (water_check.c)
predicate_impact_hook	damage_vehicle_impact (damage.c)
predicates_show	composite_show (composite.c)
print_break_set	next_page (more.c)
print_break_set_entry	print_break_set (breakset.c)
print_chr	perror (perror.c)
	print_num (perror.c)
print_lists	compute_who_follows_whom (formation.c)
print_message_position	sbx_connection_create_unit_msg (sbx.c)
print_num	perror (perror.c)
print_opfor_header	sbx_connection_process_message (sbx.c)
print_perf_stats	collect_perf_stat (perf.c)
print_reasons	print_reasons_and_clear (parser.c)
print_route	driver_show (driver.c)
print_routepoint	driver_show (driver.c)
	print_route (route.c)
	remote_show (remote.c)
	show_route_points (route.c)
print_sockaddr	udp_write (udp_berkeley.c)
print_str	perror (perror.c)
	print_num (perror.c)
print_vector	airveh_show (flyingveh.c)
	driver_show (driver.c)
	entity_show (entity.c)
	pilot_show (pilot.c)
	pilot_show_machine (pilot.c)
print_vehicle_ids	sbx_connection_reset_msg (sbx.c)
print_vehicle_marking	composite_show (composite.c)
	vehicle_show (vehicle.c)
printdirection	driver_show (driver.c)
printf	abort_handler (main.c)
	access_147 (net_acce.c)
	add_vehicle_to_cluster (reporter.c)
	air_tick (flyingveh.c)
	airveh_show (flyingveh.c)
	allocate_safobj (safobj.c)
	AssocOpen (open.c)
	broadcast_appearance_data (sbx.c)
	broadcast_pae_data (sbx.c)

Called Function	Calling Function (filename)
printf (cont.)	buf_rudp_discard_all_buffers (buf_conn.c)
	buf_rudp_flush (buf_conn.c)
	buf_rudp_show (buf_conn.c)
	buf_rudp_tick (buf_conn.c)
	buffer_statistics_print (queue_fix.c)
	choose_skirt_point (collision.c)
	choose_target_and_weapon (targeting.c)
	cis_show (cis.c)
	cluster_shell (reporter.c)
	cluster_vehicle (reporter.c)
	cm_add_area (cm.c)
	cm_add_line (cm.c)
	cm_add_point (cm.c)
	cm_add_route (cm.c)
	cm_add_zone (cm.c)
	cm_delete_cm (cm.c)
	cm_delete_cm_from_overlay (cm.c)
	cm_delete_overlay (cm.c)
	cm_execute_overlay (cm.c)
	collision_dead_tick (collision.c)
	collision_show (collision.c)
	collision_tick (collision.c)
	commander_execute_overlay (commander.c)
	commander_show (commander.c)
	composite_assume_formation_internal (composite.c)
	composite_pro_sim (composite.c)
	composite_remove_inferior_composite (composite.c)
	composite_remove_inferior_vehicle (composite.c)
	composite_remove_member_vehicle (composite.c)
	composite_set_superior (composite.c)
	composite_show (composite.c)
	composite_tick (composite.c)
	compute_who_follows_whom (formation.c)
	cons (list.c)
	count_forces (parser.c)
	count_hosts (parser.c)
	count_sites (parser.c)
	count_vehicles (parser.c)
	countries_from_battle_scheme_and_force (parser.c)
	create_bridges (bridges.c)
	create_buildings (trees.c)
	create_contours (trees.c)
	create_remote_vehicle (remote.c)
	create_river_ints (rivers.c)
	create_rivers (rivers.c)
	create_road_ints (roads.c)
	create_roads (roads.c)
	create_targeting (targeting.c)
	create_tree_canopies (trees.c)
	create_trees (trees.c)
	create_weapon_systems (weapons.c)

Called Function	Calling Function (filename)
printf (cont.)	damage_indirect_fire (damage.c)
	damage_vehicle_impact (damage.c)
	damage_vehicle_rammed (damage.c)
	database_df_damage_query (df_damage.c)
	database_if_damage_query (if_damage.c)
	database_read (database.c)
	debugging_on (main.c)
	destroy_cm_data (cm.c)
	detect_building_on_path_tick (collision.c)
	detect_collision_tick (collision.c)
	detect_imminent_collision_tick (collision.c)
	detection_show (detection.c)
	detection_show_type (detection.c)
	do_heap_collect (alloc.c)
	do_heap_statistics (alloc.c)
	do_heap_verify (alloc.c)
	do_mode_cmd (net_mode.c)
	do_mode_cmd_147 (net_mode.c)
	driver_execute_overlay (driver.c)
	driver_follow_leader (driver.c)
	driver_follow_vehicle_cmd (driver.c)
	driver_set_leader_mis (driver.c)
	driver_show (driver.c)
	driver_tick (driver.c)
	eliminate_vehicles (saf.c)
	enable_predicate (predicates.c)
	enqueue_on_rcvq (simnet.c)
	entity_show (entity.c)
	ethernet_2_packets (main.c)
	execute_overlay (dispatch.c)
	fake_resupply (dispatch.c)
	fill_in_appearance_data (dispatch.c)
	fill_in_echelon_data (dispatch.c)
	fill_in_position_data (dispatch.c)
	find_new_host_mask (id_hash.c)
	fire_missile_at_target (missile.c)
	fire_weapon_at_target (weapons.c)
	flush (more.c)
	fly_missile (missile.c)
	fly_round (weapons.c)
	followvehicle (driver.c)
	gasp (main.c)
	generate_a_deactivate (pro_sim.c)
	generate_indirect_fire_pkt (pro_sim.c)
	generate_status_report (dispatch.c)
	generate_weapon_hit (weapons.c)
	generate_weapon_miss (weapons.c)
	get_a_leader (commander.c)
	get_control_measure (cm.c)
	get_overlay (cm.c)
	go_away (dispatch.c)

Called Function	Calling Function (filename)
printf (cont.)	groundveh_tick (groundveh.c)
	gunner_tick (gunner.c)
	host_loopback_function (simnet.c)
	identify_exercise (parser.c)
	identify_version (version.c)
	init_mappings (map.c)
	init_stuff (main.c)
	init_terrain_stuff (main.c)
	initialize_weapon_priority_list (weapons.c)
	intervis_possibly_visible (intervis.c)
	isolate_connection (main.c)
	issue_reports (reporter.c)
	ivis_send_contact_report (IvisPDU.c)
	ivis_send_shell_report (IvisPDU.c)
	ivis_send_spot_report (IvisPDU.c)
	loader_tick (loader.c)
	logistics_supply_offer_received (logistics.c)
	logistics_tick (logistics.c)
	main (main.c)
	major_detection_increase (detection.c)
	major_spotter_increase (spotter.c)
	make_me_non_blocking (udp_berkeley.c)
	map_key (breakset.c)
	missile_ground_impact (missile.c)
	missile_maybe_hit_target (missile.c)
	missile_show (missile.c)
	navigator_entered_area (navigator.c)
	navigator_entered_zone (navigator.c)
	navigator_left_area (navigator.c)
	navigator_left_zone (navigator.c)
	navigator_passed_line (navigator.c)
	navigator_passed_point (navigator.c)
	navigator_show (navigator.c)
	net_access (net_acce.c)
	net_current_time (net_time.c)
	net_get_rcv (net_recv.c)
	net_open (net_open.c)
	net_release_rcv (net_recv.c)
	net_send (net_osend.c)
	net_snd (net_send.c)
	network_silent (main.c)
	open_147 (net_open.c)
	p_follower_comp_reset_route (p_follower.c)
	p_follower_find_pnt_in_turn (p_follower.c)
	p_follower_flying_independent (p_follower.c)
	p_follower_followroute_tick (p_follower.c)
	p_follower_gen_coord_goal_pnt (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
	p_follower_land (p_follower.c)
	p_follower_landhold_tick (p_follower.c)
	p_follower_pnt_in_hard_turn (p_follower.c)

Called Function	Calling Function (filename)
printf (cont.)	p_follower_pnt_in_shallow_turn (p_follower.c)
	p_follower_stop_coord (p_follower.c)
	parser_set_targeting_parameters (parser.c)
	pilot_check_state (pilot.c)
	pilot_execute_overlay (pilot.c)
	pilot_follow_leader (pilot.c)
	pilot_init_hoverattack (pilot.c)
	pilot_init_land (pilot.c)
	pilot_point_at_target (pilot.c)
	pilot_set_leader_mis (pilot.c)
	pilot_show (pilot.c)
	pilot_show_machine (pilot.c)
	pilot_start_gotopoint (pilot.c)
	pilot_start_hoverattack (pilot.c)
	pilot_start_hoverattack_approach (pilot.c)
	pilot_start_hoverattack_complete (pilot.c)
	pilot_start_hoverattack_egress (pilot.c)
	pilot_start_hoverhold (pilot.c)
	pilot_start_landing (pilot.c)
	pilot_start_orbithold (pilot.c)
	pilot_start_takingoff (pilot.c)
	plane_tick (plane.c)
	pop_cis (cis.c)
	predicates_show (predicates.c)
	print_break_set (breakset.c)
	print_commands_from_sbx (main.c)
	print_lists (formation.c)
	print_message_position (sbx.c)
	print_opfor_header (sbx.c)
	print_perf_stats (perf.c)
	print_reasons (thresh.c)
	print_route (route.c)
	print_routepoint (route.c)
	print_sockaddr (udp_berkeley.c)
	print_vector (math.c)
	print_vehicle_ids (sbx.c)
	print_vehicle_marking (vehicle.c)
	printdirection (driver.c)
	printimmediate (driver.c)
	printmission (driver.c)
	printroutedirection (driver.c)
	protocol_data_process (simnet.c)
	protocol_sim_process (simnet.c)
	protocol_stealth_process (simnet.c)
	read_feature (terrain.c)
	read_machine_file (main.c)
	read_quadtree (terrain.c)
	readjust_overlay (dispatch.c)
	remote_deactivate (remote.c)
	remote_show (remote.c)

Called Function	Calling Function (filename)
printf (cont.)	remote_tick (remote.c)
	remove_clustered_vehicle (reporter.c)
	remove_dead_clustered_vehicles (reporter.c)
	report_error_from_tdb_once (misc.c)
	report_id_hashing_statistics (id_hash.c)
	reporter_show (reporter.c)
	reset_counters_etc (more.c)
	rudp_ack_received (rel_conn.c)
	rudp_discard_all_buffers (rel_conn.c)
	rudp_disconnect (rel_conn.c)
	rudp_post_message (rel_conn.c)
	rudp_read_message (rel_conn.c)
	rudp_retransmit (rel_conn.c)
	rudp_retransmit_buf (rel_conn.c)
	rudp_send_bare_ack (rel_conn.c)
	rudp_show (rel_conn.c)
	rudp_synch (rel_conn.c)
	saf_print_help (main.c)
	saf_vehicle_fake_resupply (saf_vehicle.c)
	saf_vehicle_out_of_gas (saf_vehicle.c)
	saf_vehicle_pro_sim (saf_vehicle.c)
	saf_vehicle_set_leader (saf_vehicle.c)
	saf_vehicle_set_superior (saf_vehicle.c)
	saf_vehicle_show (saf_vehicle.c)
	sbx_add_area (sbx.c)
	sbx_add_line (sbx.c)
	sbx_add_point (sbx.c)
	sbx_add_route (sbx.c)
	sbx_add_zone (sbx.c)
	sbx_altitude (sbx.c)
	sbx_attach_stealth (sbx.c)
	sbx_attack (sbx.c)
	sbx_change_formation (sbx.c)
	sbx_change_speed (sbx.c)
	sbx_connection_create_unit_msg (sbx.c)
	sbx_connection_exit (sbx.c)
	sbx_connection_generate_arty_msg (sbx.c)
	sbx_connection_process_message (sbx.c)
	sbx_connection_query_sub_state (sbx.c)
	sbx_connection_reset_msg (sbx.c)
	sbx_connection_service_poll_msg (sbx.c)
	sbx_connection_set_ivis_parameters (sbx.c)
	sbx_connection_set_ivis_xmit_modes (sbx.c)
	sbx_connection_show (sbx.c)
	sbx_connection_show_top_level_units (sbx.c)
	sbx_connection_vehicle_reinit (sbx.c)
	sbx_connection_write_buffer (sbx.c)
	sbx_delete_cm (sbx.c)
	sbx_delete_overlay (sbx.c)
	sbx_execute_overlay (sbx.c)
	sbx_face_direction (sbx.c)

Called Function	Calling Function (filename)
printf (cont.)	sbx_follow_vehicle (sbx.c)
	sbx_goto_point (sbx.c)
	sbx_halt (sbx.c)
	sbx_hold (sbx.c)
	sbx_land (sbx.c)
	sbx_rejoin_unit (sbx.c)
	sbx_resume_mission (sbx.c)
	sbx_resupply (sbx.c)
	sbx_set_specific_known_vehicles (sbx.c)
	sbx_set_targeting_parameters (sbx.c)
	sbx_set_top_level_known_vehicles (sbx.c)
	sbx_simulator_in_command (sbx.c)
	send_147 (net_osend.c)
	send_147_8023 (net_send.c)
	send_contact_report (reporter.c)
	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
	set_exercise_id (main.c)
	set_ground_impact_mode (parser.c)
	set_indirect_fire_mode (parser.c)
	set_monitor_period (parser.c)
	set_targeting_parameters (dispatch.c)
	set_tdb_to_cache (main.c)
	set_terrain_dbase (main.c)
	show (dispatch.c)
	show_applies_to (cm.c)
	show_area_cm (cm.c)
	show_control_measure (cm.c)
	show_control_measures (cm.c)
	show_df_damage_entry (df_damage.c)
	show_if_damage_entry (if_damage.c)
	show_line_cm (cm.c)
	show_overlay (cm.c)
	show_overlays_all_sbx (sbx.c)
	show_point_cm (cm.c)
	show_point_list (cm.c)
	show_removed_vehicles (reporter.c)
	show_route (route.c)
	show_shelling_clusters (reporter.c)
	show_vehicle (parser.c)
	show_vehicle_cluster (reporter.c)
	show_vehicle_id_list (reporter.c)
	show_vehicle_ids (parser.c)
	show_zone_cm (cm.c)
	sigh (main.c)
	simnet_60_constants (main.c)
	simnet_getstats (simnet.c)
	simnet_id_from_saf_id (id_hash.c)
	simnet_send_appearance (SimulationPDU.c)
	simnet_send_collision (SimulationPDU.c)
	simnet_send_deactivate (SimulationPDU.c)

Called Function	Calling Function (filename)
printf (cont.)	simnet_send_fire (SimulationPDU.c)
	simnet_send_impact (SimulationPDU.c)
	simnet_send_indirect_fire (SimulationPDU.c)
	simnet_send_radiate (SimulationPDU.c)
	simnet_send_resupply_cancel (SimulationPDU.c)
	simnet_send_resupply_received (SimulationPDU.c)
	simnet_send_resupply_request (SimulationPDU.c)
	simnet_tick (simnet.c)
	spotter_show (spotter.c)
	spotter_show_table (spotter.c)
	start_avoiding_house (collision.c)
	start_resupply_of_to (logistics.c)
	start_simnet (simnet.c)
	stealth_attach_to (parser.c)
	stealth_teleport_to (parser.c)
	targeting_set_parameters (targeting.c)
	targeting_show (targeting.c)
	targeting_tick (targeting.c)
	tty_invoke_fcn_on_char (breakset.c)
	tty_restore_modes (setup.c)
	tty_setup_modes (setup.c)
	turret_scan (turret.c)
	turret_show (turret.c)
	turret_slew (turret.c)
	udp_open (udp_berkeley.c)
	udp_read (udp_berkeley.c)
	udp_write (udp_berkeley.c)
	update_clusters (reporter.c)
	update_target_list (targeting.c)
	vehicle_defuel (parser.c)
	vehicle_fake_resupply (parser.c)
	vehicle_kill (pro_sim.c)
	vehicle_manager_print (iterator.c)
	vehicle_reappeared (reporter.c)
	vehicle_resupply (parser.c)
	vehicle_show (vehicle.c)
	weapon_deselect (weapons.c)
	weapon_load (weapons.c)
	weapon_priority_list_show (weapons.c)
	weapon_select (weapons.c)
	weapon_show (weapons.c)
	weapon_systems_show (weapons.c)
	weapon_unload (weapons.c)
	write_string_to_tty (more.c)
	yylex (lexer.c)
printimmediate	driver_show (driver.c)
printmission	driver_show (driver.c)
printroutedirection	driver_show (driver.c)
proc_switches	main (main.c)
project_point	pilot_init_hoverattack (pilot.c)

Called Function	Calling Function (filename)
protocol_data_process	host_loopback_function (simnet.c)
	simnet_tick (simnet.c)
protocol_sim_process	host_loopback_function (simnet.c)
	simnet_tick (simnet.c)
protocol_stealth_process	simnet_tick (simnet.c)
push	buildings_thru (building_check.c)
	get_building_indecies (building_check.c)
	get_quad_nodes_internal (search.c)
	get_water_indecies (water_check.c)
push_cis	activate_new_cis (cis.c)
	check_predicates (predicates.c)
utlong	res_mkquery (res_mkquery.c)
putshort	res_mkquery (res_mkquery.c)
pve_checkvis	intervis_get_high_ground (intervis.c)
	intervis_pt_to_pt (intervis.c)
	intervis_possibly_visible (intervis.c)
pwskip	_shadow (ht_getpwent.c)
pwskip	interpret (yp_getpwent.c)
qsort	sort_form_db (formation.c)
	sort_tag_table (tags.c)
	update_target_list (targeting.c)
qst	qsort (qsort.c)
	qst (qsort.c)
queue_allocate	rudp_init (rel_conn.c)
	buf_rudp_init (buf_conn.c)
queue_dequeue	buffer_dequeue (queue_fix.c)
queue_enqueue	buffer_enqueue (queue_fix.c)
queue_flush	buffer_flush (queue_fix.c)
queue_flush_hangers	queue_flush_hangers (queue_fix.c)
	queue_flush (queue_fix.c)
queue_length	buf_rudp_show (buf_conn.c)
	buf_rudp_tick (buf_conn.c)
	rudp_ack_received (rel_conn.c)
	rudp_post_message (rel_conn.c)
	rudp_show (rel_conn.c)
	sbx_connection_overloaded (sbx.c)
adiate_target	track_target (gunner.c)
rand	get_me_a_random_fraction (random.c)
randomize_vector	get_i_and_t_from_normal (impact.c)
range_squared	choose_target_and_weapon (targeting.c)
	damage_indirect_fire (damage.c)
	fly_missile (missile.c)
	generate_weapon_hit (weapons.c)
	generate_weapon_miss (weapons.c)
	logistics_tick (logistics.c)
	missile_maybe_hit_target (missile.c)
	shoot_target (gunner.c)
	target_in_position (targeting.c)
	update_target_list (targeting.c)
rassoc_sym	map_echelon_type_number_to_echelon_type_symbol (map.c)
	map_echelon_number_to_echelon_symbol (map.c)

Called Function	Calling Function (filename)
re_assign_routepoints	cm_add_route (cm.c)
read	_filbuf (filbuf.c)
	create_bridges (bridges.c)
	create_buildings (trees.c)
	create_contours (trees.c)
	create_lakes (bridges.c)
	create_river_ints (rivers.c)
	create_rivers (rivers.c)
	create_road_ints (roads.c)
	create_roads (roads.c)
	create_tree_canopies (trees.c)
	create_trees (trees.c)
	init_patch_indices (cache_init.c)
	memory_init (memory.c)
	read_feature (terrain.c)
	read_quadtree (terrain.c)
	res_send (res_send.c)
	tdb_get_db_format (version.c)
	tdb_init_patch_guards (cache_init.c)
	tdb_read_header (header.c)
	terrain_cache_inquire (cache_query.c)
	tzload (ctime.c)
read_feature	read_quadtree (terrain.c)
read_machine_file	init_stuff (main.c)
read_quadtree	read_quadtree_database (terrain.c)
read_quadtree_database	init_terrain_stuff (main.c)
read_terrain_files	read_quadtree_database (terrain.c)
reader_read_file	database_init (main.c)
	database_read (database.c)
	init_mappings (map.c)
	read_machine_file (main.c)
readjust_overlay	cm_force_overlay_recalculation (cm.c)
realloc	ht_fgetgrent (ht_getgrent.c)
recv	res_send (res_send.c)
recvfrom	udp_read (udp_berkeley.c)
	clntudp_call (clnt_udp.c)
rejected	_seterr_reply (rpc_prot.c)
remote_change_stealth_controlling_port	sbx_attach_stealth (sbx.c)
remote_deactivate	remote_tick (remote.c)
remote_fill_in_appearance_data	fill_in_appearance_data (dispatch.c)
remote_fill_in_echelon_data	fill_in_echelon_data (dispatch.c)
remote_fill_in_position_data	fill_in_position_data (dispatch.c)
remote_go_away	go_away (dispatch.c)
remote_init_vars	create_remote_vehicle (remote.c)
remote_new_appearance	protocol_sim_process (simnet.c)
	protocol_stealth_process (simnet.c)
remote_next_road_point	get_leader_state (driver.c)
remote_show	show (dispatch.c)
remote_start_being_watched	remote_next_road_point (remote.c)
remote_start_ticking	create_remote_vehicle (remote.c)
remote_stop_being_watched	remote_tick (remote.c)

Called Function	Calling Function (filename)
remove_clustered_vehicle	update_reporter_vehicles (reporter.c)
	reporter_remove_vehicles (reporter.c)
remove_dead_clustered_vehicles	update_clusters (reporter.c)
remove_duplicates	get_building_indecies (building_check.c)
	get_water_indecies (water_check.c)
remove_executing_unit	cm_execute_overlay (cm.c)
remove_function	change_fncl_period (fncl.c)
	cancel_fncl (fncl.c)
remove_grid_entry	update_grid_entry_list (iterator.c)
	destroy_grid_entry_list (iterator.c)
remove_removed_vehicle	vehicle_reappeared (reporter.c)
remove_shelling_cluster	issue_reports (reporter.c)
remove_vehicles	remote_deactivate (remote.c)
	eliminate_vehicles (saf.c)
report_cluster_vehicle_type	send_spot_report (reporter.c)
	send_contact_report (reporter.c)
report_error_from_tdb_once	get_soil_type (misc.c)
	groundveh_move (groundveh.c)
	pilot_start_landed (pilot.c)
	saf_vehicle_teleport (saf_vehicle.c)
	tdb_get_gl (misc.c)
	tdb_get_zl (misc.c)
report_hit	check_hits (pve_checkvis.c)
report_ivis_network	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
	send_contact_report (reporter.c)
report_last_hit	pve_checkvis (pve_checkvis.c)
report_tree_block	check_tree_hit (pve_checkvis.c)
reporter_remove_vehicles	composite_remove_vehicles (composite.c)
reporter_show	composite_show (composite.c)
reporter_tick	composite_tick (composite.c)
res_init	res_mkquery (res_mkquery.c)
	res_query (res_query.c)
	res_search (res_query.c)
	res_send (res_send.c)
res_mkquery	res_query (res_query.c)
res_query	dns_gethostbyaddr (dns_getnamadr.c)
	res_search (res_query.c)
	res_querydomain (res_query.c)
res_querydomain	res_search (res_query.c)
res_search	dns_gethostbyname (dns_getnamadr.c)
res_send	res_query (res_query.c)
reset_counters_etc	flush (more.c)
	print_break_set (breakset.c)
reset_predicates	adjust_behaviors (cis.c)
	create_predicates (predicates.c)
resuply_check_ok	logistics_tick (logistics.c)
resupply_bld_ammo_needs	choose_resupply_item (logistics.c)
	logistics_tick (logistics.c)
rewind	yp_setnetent (yp_gnetent.c)
rindex	nettoa (yp_gnetent.c)
rotate_queue	terrain_cache_inquire (cache_query.c)

Called Function	Calling Function (filename)
routepoint_distance	append_routepoint (route.c)
rpc_perror	xdr_bytes (xdr.c)
	xdr_string (xdr.c)
rudp_ack_received	rudp_read_message (rel_conn.c)
rudp_close	buf_rudp_close (buf_conn.c)
rudp_discard_all_buffers	rudp_disconnect (rel_conn.c)
	rudp_synch (rel_conn.c)
rudp_disconnect	buf_rudp_disconnect (buf_conn.c)
rudp_init	buf_rudp_init (buf_conn.c)
rudp_open	buf_rudp_open (buf_conn.c)
rudp_post_message	buf_rudp_flush (buf_conn.c)
rudp_read_message	buf_rudp_read_message (buf_conn.c)
rudp_retransmit	rudp_tick (rel_conn.c)
rudp_send	rudp_retransmit_buf (rel_conn.c)
	rudp_post_message (rel_conn.c)
	rudp_send_bare_ack (rel_conn.c)
rudp_send_bare_ack	rudp_retransmit (rel_conn.c)
rudp_show	buf_rudp_show (buf_conn.c)
rudp_synch	rudp_read_message (rel_conn.c)
rudp_tick	buf_rudp_tick (buf_conn.c)
saf_complete_reset	saf_exit (main.c)
	parser_global_reset (parser.c)
saf_id_from_simnet_id	collision_vehicle_rammed (collision.c)
	damage_indirect_fire (damage.c)
	damage_vehicle_impact (damage.c)
	damage_vehicle_rammed (damage.c)
	interesting_packet_test (simnet.c)
	logistics_supply_offer_received (logistics.c)
	report_id_hashing_statistics (id_hash.c)
	sbx_connection_process_ground_impact (sbx.c)
	sbx_connection_process_vehicle_impact (sbx.c)
saf_reset_for_workstation	sbx_connection_reset_msg (sbx.c)
saf_vehicle_cancel_overlay	commander_cancel_overlay (commander.c)
	saf_vehicle_rejoin_unit (saf_vehicle.c)
saf_vehicle_catastrophic_kill	air_tick (flyingveh.c)
	damage_indirect_fire (damage.c)
	damage_vehicle_impact (damage.c)
	saf_vehicle_mobility_kill (saf_vehicle.c)
	saf_vehicle_reinit (saf_vehicle.c)
	vehicle_kill (pro_sim.c)
saf_vehicle_change_speed	commander_change_speed (commander.c)
	sbx_change_speed (sbx.c)
saf_vehicle_checkpoint_state	sbx_connection_wat_pulse (sbx.c)
saf_vehicle_collision_over	collision_tick (collision.c)
saf_vehicle_doing_collision_stuff	followvehicle (driver.c)
saf_vehicle_est_position	fly_missile (missile.c)
	p_follower_followroute_tick (p_follower.c)
	p_follower_hoverhold (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
	p_follower_landhold_tick (p_follower.c)
	shoot_target (gunner.c)
saf_vehicle_execute_overlay	execute_overlay (dispatch.c)

Called Function	Calling Function (filename)
saf_vehicle_face_direction	commander_face_direction (commander.c)
	sbx_face_direction (sbx.c)
saf_vehicle_fake_resupply	pilot_start_landed (pilot.c)
	composite_fake_resupply (composite.c)
	fake_resupply (dispatch.c)
saf_vehicle_fill_in_appearance_data	fill_in_appearance_data (dispatch.c)
saf_vehicle_fill_in_echelon_data	fill_in_echelon_data (dispatch.c)
saf_vehicle_fill_in_position_data	fill_in_position_data (dispatch.c)
saf_vehicle_firepower_kill	damage_indirect_fire (damage.c)
	damage_vehicle_impact (damage.c)
	saf_vehicle_reinit (saf_vehicle.c)
saf_vehicle_follow_vehicle	commander_follow_vehicle (commander.c)
	sbx_follow_vehicle (sbx.c)
saf_vehicle_generate_status_report	generate_status_report (dispatch.c)
saf_vehicle_go_away	go_away (dispatch.c)
saf_vehicle_goto_point	commander_goto_point (commander.c)
	sbx_goto_point (sbx.c)
saf_vehicle_halt	commander_halt (commander.c)
	collision_tick (collision.c)
	sbx_halt (sbx.c)
saf_vehicle_indirect_fire	saf_vehicle_pro_sim (saf_vehicle.c)
saf_vehicle_mimic_vehicle	commander_restore_leader_state (commander.c)
	commander_simulator_in_command (commander.c)
saf_vehicle_mobility_kill	damage_indirect_fire (damage.c)
	damage_vehicle_impact (damage.c)
	damage_vehicle_rammed (damage.c)
	saf_vehicle_reinit (saf_vehicle.c)
saf_vehicle_next_event_id	detect_collision_tick (collision.c)
	fire_weapon_at_target (weapons.c)
	fire_missile_at_target (missile.c)
saf_vehicle_out_of_gas	saf_vehicle_tick (saf_vehicle.c)
saf_vehicle_pro_sim	saf_vehicle_tick (saf_vehicle.c)
saf_vehicle_reinit	sbx_connection_vehicle_reinit (sbx.c)
saf_vehicle_rejoin_unit	sbx_rejoin_unit (sbx.c)
saf_vehicle_remove_vehicles	remove_vehicles (dispatch.c)
saf_vehicle_reset_station_keeper	commander_execute_overlay (commander.c)
	commander_face_direction (commander.c)
	commander_follow_vehicle (commander.c)
	commander_goto_point (commander.c)
	commander_halt (commander.c)
	commander_land (commander.c)
	commander_simulator_in_command (commander.c)
saf_vehicle_resume_mission	commander_resume_mission (commander.c)
	sbx_resume_mission (sbx.c)
saf_vehicle_send_appearance	saf_vehicle_tick (saf_vehicle.c)
saf_vehicle_set_direction	commander_set_mission_direction (commander.c)
saf_vehicle_set_leader	composite_assume_formation_internal (composite.c)
saf_vehicle_set_marking	create_saf_vehicle (saf_vehicle.c)
saf_vehicle_set_route	commander_execute_overlay (commander.c)
saf_vehicle_set_speed	commander_set_mission_speed (commander.c)
saf_vehicle_set_superior	create_unit (create.c)

Called Function	Calling Function (filename)
saf_vehicle_set_targeting_parameters	composite_set_targeting_parameters (composite.c)
	set_targeting_parameters (dispatch.c)
saf_vehicle_show	show (dispatch.c)
saf_vehicle_simulator_in_command	collision_tick (collision.c)
saf_vehicle_start_ticking	create_unit (create.c)
saf_vehicle_stop_mimicing	commander_face_direction (commander.c)
	commander_follow_vehicle (commander.c)
	commander_goto_point (commander.c)
	commander_halt (commander.c)
	commander_restore_leader_state (commander.c)
	commander_resume_mission (commander.c)
saf_vehicle_sudden_stop	start_disengaging (collision.c)
	collision_tick (collision.c)
saf_vehicle_teleport	saf_vehicle_teleport_to_station (saf_vehicle.c)
	create_saf_vehicle (saf_vehicle.c)
saf_vehicle_teleport_to_station	composite_teleport_to_station (composite.c)
saf_vehicle_useless	commander_attack (commander.c)
	commander_change_altitude (commander.c)
	commander_change_speed (commander.c)
	commander_execute_overlay (commander.c)
	commander_face_direction (commander.c)
	commander_follow_vehicle (commander.c)
	commander_get_mission_status (commander.c)
	commander_goto_point (commander.c)
	commander_halt (commander.c)
	commander_land (commander.c)
	commander_resume_mission (commander.c)
	commander_simulator_in_command (commander.c)
	get_station_info (formation.c)
saf_vehicle_vehicle_impact	saf_vehicle_pro_sim (saf_vehicle.c)
saf_vehicle_vehicle_rammed	saf_vehicle_pro_sim (saf_vehicle.c)
save	yp_getgrent (yp_getgrent.c)
save	yp_getpwent (yp_getpwent.c)
sbrk	morecore (malloc.c)
	malloc (malloc.c)
bx_add_area	sbx_connection_process_message (sbx.c)
sbx_add_line	sbx_connection_process_message (sbx.c)
sbx_add_point	sbx_connection_process_message (sbx.c)
sbx_add_route	sbx_connection_process_message (sbx.c)
sbx_add_zone	sbx_connection_process_message (sbx.c)
sbx_altitude	sbx_connection_process_message (sbx.c)
sbx_attach_stealth	sbx_connection_process_message (sbx.c)
sbx_attack	sbx_connection_process_message (sbx.c)
sbx_change_formation	sbx_connection_process_message (sbx.c)
sbx_change_speed	sbx_connection_process_message (sbx.c)
sbx_connection_create_unit_msg	sbx_connection_process_message (sbx.c)
sbx_connection_disconnect	sbx_connection_process_message (sbx.c)
sbx_connection_exit	exit_all_sbx_conns (sbx.c)
sbx_connection_generate_arty_msg	sbx_connection_process_message (sbx.c)
sbx_connection_init	open_io_connections (sbx.c)
sbx_connection_open	open_io_connections (sbx.c)

Called Function	Calling Function (filename)
sbx_connection_overloaded	sbx_connection_wat_pulse (sbx.c)
	sbx_connection_stat_pulse (sbx.c)
sbx_connection_process_ground_impact	ground_impact_to_sbx (sbx.c)
sbx_connection_process_indirect_fire	indirect_fire_to_sbx (sbx.c)
sbx_connection_process_message	sbx_connection_process_messages (sbx.c)
sbx_connection_process_messages	sbx_connection_tick (sbx.c)
sbx_connection_process_vehicle_impact	vehicle_impact_to_sbx (sbx.c)
sbx_connection_query_sub_state	sbx_connection_process_message (sbx.c)
sbx_connection_reset_msg	sbx_connection_process_message (sbx.c)
sbx_connection_send_to_port	activate_new_cis (cis.c)
	change_movement_and_report (cis.c)
	composite_generate_status_report (composite.c)
	remote_tick (remote.c)
	saf_vehicle_checkpoint_state (saf_vehicle.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
	sbx_printf (sbx.c)
	send_contact_report (reporter.c)
	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
	send_stealth_gone_msg (remote.c)
sbx_connection_service_poll_msg	sbx_connection_process_message (sbx.c)
sbx_connection_set_ivis_parameters	sbx_connection_process_message (sbx.c)
sbx_connection_set_ivis_xmit_modes	sbx_connection_process_message (sbx.c)
sbx_connection_show	show_connection_all_sbx (sbx.c)
sbx_connection_show_top_level_units	sbx_connection_show (sbx.c)
sbx_connection_synch_received	sbx_connection_tick (sbx.c)
sbx_connection_vehicle_reinit	sbx_connection_process_message (sbx.c)
sbx_connection_write_buffer	sbx_connection_process_ground_impact (sbx.c)
	sbx_connection_process_indirect_fire (sbx.c)
	sbx_connection_process_vehicle_impact (sbx.c)
	sbx_connection_reset_msg (sbx.c)
	sbx_connection_send_to_port (sbx.c)
	sbx_connection_stat_pulse (sbx.c)
	sbx_connection_synch_received (sbx.c)
	sbx_connection_wat_pulse (sbx.c)
	write_buffer_all_sbx (sbx.c)
sbx_delete_cm	sbx_connection_process_message (sbx.c)
sbx_delete_overlay	sbx_connection_process_message (sbx.c)
sbx_execute_overlay	sbx_connection_process_message (sbx.c)
sbx_face_direction	sbx_connection_process_message (sbx.c)
sbx_follow_vehicle	sbx_connection_process_message (sbx.c)
sbx_goto_point	sbx_connection_process_message (sbx.c)
sbx_halt	sbx_connection_process_message (sbx.c)
sbx_hold	sbx_connection_process_message (sbx.c)
sbx_land	sbx_connection_process_message (sbx.c)
sbx_printf	collision_tick (collision.c)
	composite_send_unit_strength_message (composite.c)
	logistics_tick (logistics.c)
	parser_send_string (parser.c)
	remote_change_stealth_controlling_port (remote.c)
	remote_deactivate (remote.c)

Called Function	Calling Function (filename)
sbx_printf (cont.)	resuply_check_ok (logistics.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
	sbx_simulator_in_command (sbx.c)
sbx_rejoin_unit	sbx_connection_process_message (sbx.c)
sbx_resume_mission	sbx_connection_process_message (sbx.c)
sbx_resupply	sbx_connection_process_message (sbx.c)
sbx_set_all_known_vehicles	sbx_connection_service_poll_msg (sbx.c)
sbx_set_specific_known_vehicles	sbx_connection_service_poll_msg (sbx.c)
sbx_set_targeting_parameters	sbx_connection_process_message (sbx.c)
sbx_set_top_level_known_vehicles	sbx_connection_service_poll_msg (sbx.c)
sbx_simulator_in_command	sbx_connection_process_message (sbx.c)
sbx_swap_known_vehicles	sbx_set_specific_known_vehicles (sbx.c)
	sbx_set_top_level_known_vehicles (sbx.c)
scan_weapon	gunner_tick (gunner.c)
scheduler_init	init_stuff (main.c)
seg_intersection	segment_thru_lake (water_check.c)
	segment_thru_river (water_check.c)
segment_inside_polygon	segment_thru_lake (water_check.c)
segment_thru_lake	water_check (water_check.c)
segment_thru_river	water_check (water_check.c)
select	res_send (res_send.c)
	clntudp_call (clnt_udp.c)
select_weapon_priority_list	choose_target_and_weapon (targeting.c)
send	syslog (syslog.c)
	res_send (res_send.c)
send_147	net_send (net_osend.c)
send_147_8023	net_snd (net_send.c)
send_cmc	net_send (net_osend.c)
send_cmc_8023	net_snd (net_send.c)
send_contact_report	issue_reports (reporter.c)
send_shelling_report	issue_reports (reporter.c)
send_spot_report	issue_reports (reporter.c)
send_stealth_gone_msg	remote_deactivate (remote.c)
	remote_change_stealth_controlling_port (remote.c)
sendto	clntudp_call (clnt_udp.c)
	res_send (res_send.c)
	udp_write (udp_berkeley.c)
set_bit	sbx_set_specific_known_vehicles (sbx.c)
	sbx_set_top_level_known_vehicles (sbx.c)
	spotter_tick (spotter.c)
set_bitfield	sbx_set_all_known_vehicles (sbx.c)
	sbx_set_top_level_known_vehicles (sbx.c)
set_critical_performance_level	init_stuff (main.c)
set_pv_params	intervis_get_high_ground (intervis.c)
	intervis_pt_to_pt (intervis.c)
	intervis_possibly_visible (intervis.c)
set_speed_dir	followvehicle (driver.c)
set_symbol_value	yyparse (parser.y)
set_targeting_parameters	sbx_set_targeting_parameters (sbx.c)
	parser_set_targeting_parameters (parser.c)
setbit	dbm_store (ndbm.c)
setbuf	tty_setup_modes (setup.c)

Called Function	Calling Function (filename)
setgrjunk	yp_getgrnam (yp_getgrent.c)
setitimer	alarm (alarm.c)
	sleep (sleep.c)
setnetgrent	yp_getpwent (yp_getpwent.c)
shell_report_needed	issue_reports (reporter.c)
shell_type_to_string	show_shelling_clusters (reporter.c)
shoot_target	gunner_tick (gunner.c)
show	show_vehicle (parser.c)
show_applies_to	show_control_measure (cm.c)
show_area_cm	show_control_measure (cm.c)
show_connection_all_sbx	show_connection (parser.c)
show_control_measure	show_control_measures (cm.c)
show_control_measures	show_overlay (cm.c)
	navigator_show (navigator.c)
show_df_damage_entry	database_df_damage_query (df_damage.c)
show_if_damage_entry	database_if_damage_query (if_damage.c)
show_line_cm	show_control_measure (cm.c)
show_overlay	show_overlays (cm.c)
show_overlays	show_overlays_all_sbx (sbx.c)
show_overlays_all_sbx	show_sbx_overlays (parser.c)
show_point_cm	show_control_measure (cm.c)
show_point_list	show_line_cm (cm.c)
	show_zone_cm (cm.c)
	show_area_cm (cm.c)
show_removed_vehicles	reporter_show (reporter.c)
show_route	show_control_measure (cm.c)
show_route_points	show_route (route.c)
show_shelling_clusters	reporter_show (reporter.c)
show_vehicle_cluster	show_vehicle_clusters (reporter.c)
show_vehicle_clusters	reporter_show (reporter.c)
show_vehicle_id_list	show_overlay (cm.c)
	show_vehicle_cluster (reporter.c)
show_zone_cm	show_control_measure (cm.c)
sigblock	sleep (sleep.c)
	syslog (syslog.c)
signal	main (main.c)
	syslog (syslog.c)
sigpause	sleep (sleep.c)
sigsetmask	sleep (sleep.c)
	syslog (syslog.c)
sigvec	signal (signal.c)
	sleep (sleep.c)
simnet_exit	abort_handler (main.c)
	start_simnet (simnet.c)
	exit_handler (main.c)
simnet_id_from_saf_id	sbx_attach_stealth (sbx.c)
	simnet_id_string_from_saf_id (id_hash.c)
	simnet_send_appearance (SimulationPDU.c)
	simnet_send_collision (SimulationPDU.c)
	simnet_send_deactivate (SimulationPDU.c)

Called Function	Calling Function (filename)
simnet_id_from_saf_id (cont.)	simnet_send_fire (SimulationPDU.c)
	simnet_send_impact (SimulationPDU.c)
	simnet_send_indirect_fire (SimulationPDU.c)
	simnet_send_radiate (SimulationPDU.c)
	simnet_send_resupply_cancel (SimulationPDU.c)
	simnet_send_resupply_received (SimulationPDU.c)
	simnet_send_resupply_request (SimulationPDU.c)
	simnet_send_status_change (DataCollectionPDU.c)
	stealth_send_attach_to_vehicle (StealthPDU.c)
	vehicle_manager_count_hosts (iterator.c)
	vehicle_manager_count_sites (iterator.c)
simnet_id_string_from_saf_id	missile_show (missile.c)
	protocol_sim_process (simnet.c)
	protocol_stealth_process (simnet.c)
	vehicle_show (vehicle.c)
simnet_send_appearance	missile_send_appearance (missile.c)
	saf_vehicle_send_appearance (saf_vehicle.c)
simnet_send_collision	detect_collision_tick (collision.c)
simnet_send_deactivate	saf_vehicle_mimic_vehicle (saf_vehicle.c)
	missile_deactivate (missile.c)
	generate_a_deactivate (pro_sim.c)
simnet_send_fire	fire_weapon_at_target (weapons.c)
	fire_missile_at_target (missile.c)
simnet_send_impact	generate_weapon_hit (weapons.c)
	generate_weapon_miss (weapons.c)
	missile_ground_impact (missile.c)
	missile_maybe_hit_target (missile.c)
	vehicle_kill_remote (pro_sim.c)
	vehicle_ping (pro_sim.c)
simnet_send_indirect_fire	generate_indirect_fire_pkt (pro_sim.c)
simnet_send_radiate	radiate_target (weapons.c)
simnet_send_resupply_cancel	logistics_tick (logistics.c)
simnet_send_resupply_received	logistics_tick (logistics.c)
simnet_send_resupply_request	logistics_tick (logistics.c)
simnet_send_status_change	damage_indirect_fire (damage.c)
	damage_vehicle_impact (damage.c)
	damage_vehicle_rammed (damage.c)
imnet_zerostats	start_simnet (simnet.c)
simple_queue_allocate	create_entity (entity.c)
	buffer_pool_allocate (bufpool.c)
simple_queue_deallocate	destroy_entity (entity.c)
simple_queue_dequeue	buffer_simple_dequeue (bufpool.c)
	buffer_allocate_from_pool (bufpool.c)
simple_queue_enqueue	buffer_pool_return_buffer (bufpool.c)
	buffer_simple_enqueue (bufpool.c)
sin	create_entity (entity.c)
	driver_face_direction_cmd (driver.c)
	helo_tick (helo.c)
	hull_to_world_from_orientation (flyingveh.c)
	mat_rot_init (mat_r_init.c)
	pilot_face_direction (pilot.c)

Called Function	Calling Function (filename)
sin (cont.)	plane_tick (plane.c)
	turret_muzzle_position_in_world_coordinates (turret.c)
	turret_set_scan_parms (turret.c)
sleep	yp_first (yp_enum.c)
sockaddr_init	udp_open (udp_berkeley.c)
socket	check_pmap_up (yp_bind.c)
	clntudp_bufcreate (clnt_udp.c)
	openlog (syslog.c)
	res_send (res_send.c)
	udp_open (udp_berkeley.c)
sort_damage_table	database_init (main.c)
sort_form_db	database_init (main.c)
sort_tag_table	sort_damage_table (df_damage.c)
	database_init (main.c)
sort_trim_hits	patch (pve_checkvis.c)
splice_out_vehicle_cluster	remove_clustered_vehicle (reporter.c)
	update_clusters (reporter.c)
potter_init_table	spotter_tick (spotter.c)
	allocate_spotter_tables (spotter.c)
spotter_remove_vehicles	composite_remove_vehicles (composite.c)
spotter_save_weight	targeting_tick (targeting.c)
spotter_show	composite_show (composite.c)
spotter_show_table	spotter_show (spotter.c)
potter_sum_inferior_composite_tables	spotter_tick (spotter.c)
spotter_tick	composite_tick (composite.c)
spread_increment	spread_increment (formation.c)
	get_formation_layout (formation.c)
sprintf	_tztab (timezone.c)
	asctime (ctime.c)
	change_movement_and_report (cis.c)
	clnt_spcreateerror (clnt_perror.c)
	clnt_sperror (clnt_perror.c)
	commander_get_mission_status (commander.c)
	composite_generate_status_report (composite.c)
	dns_gethostbyaddr (dns_getnamadr.c)
	fill_generic_cis_report (cis.c)
	gasp (main.c)
	get_sender_of_last_packet (simnet.c)
	inet_ntoa (inet_ntoa.c)
	parser_init (parser.c)
	print_break_set_entry (breakset.c)
	res_mkquery (res_mkquery.c)
	res_querydomain (res_query.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
	saf_vehicle_set_marking (saf_vehicle.c)
	send_contact_report (reporter.c)
	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
	simnet_id_string_from_saf_id (id_hash.c)
	syslog (syslog.c)
	tdb_error (error.c)
	vehicle_status_string (saf_vehicle.c)

Called Function	Calling Function (filename)
sqrt	asin (../asincos.s)
	combined_velocity (pilot.c)
	compute_damage_keys (df_damage.c)
	database_df_damage_query (df_damage.c)
	database_if_damage_query (if_damage.c)
	denorm_sqrt (../sqrt.s)
	detectable (detection.c)
	diffraction_diff (pve_checkvis.c)
	distance (water_utilities.c)
	exagerate_bridges (route.c)
	fire_missile_at_target (missile.c)
	fly_missile (missile.c)
	followvehicle (driver.c)
	generate_weapon_hit (weapons.c)
	generate_weapon_miss (weapons.c)
	groundveh_tick (groundveh.c)
	helo_tick (helo.c)
	hull_to_world_from_direction (missile.c)
	interior_angle_between_vectors (math.c)
	missile_ground_impact (missile.c)
	missile_maybe_hit_target (missile.c)
	missile_set_desired_direction (missile.c)
	missile_show (missile.c)
	noa_damp (misc.c)
	p_follower_gen_coord_goal_pnt (p_follower.c)
	p_follower_leader_passed_point (p_follower.c)
	phase_two_collision_check (collision.c)
	pilot_goto_endpoint (pilot.c)
	pilot_point_at_target (pilot.c)
	plane_tick (plane.c)
	remote_next_road_point (remote.c)
	reporter_show (reporter.c)
	routepoint_distance (route.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
	shoot_target (gunner.c)
	startup (pve_checkvis.c)
	targeting_show (targeting.c)
	tdb_get_hull_to_world (h_to_w.c)
	turret_point_at_target (turret.c)
	vec2_mag (math.c)
	vec2_norm (math.c)
	vec_normalize (vec_norm.c)
	vel2point (driver.c)
	weapon_show (weapons.c)
	xy_dir_and_range (pilot.c)
	z_velocity (pilot.c)
srand	init_stuff (main.c)
sscanf	isolate_connection (main.c)
	ProcessAddlDescriptors (params.c)
	ProcessHost (params.c)

Called Function	Calling Function (filename)
sscanf (cont.)	ProcessInitDescriptors (params.c)
	ProcessMaxSubscriptions (params.c)
	ProcessSite (params.c)
	send_contact_report (reporter.c)
	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
	set_exercise_id (main.c)
	yylex (lexer.c)
stack_push	yyparse (parser.y)
	stack_push_array (parser.y)
stack_push_array	yyparse (parser.y)
start_avoiding_collision	collision_tick (collision.c)
start_avoiding_house	collision_tick (collision.c)
start_disengaging	collision_tick (collision.c)
start_resupply_of_to	sbx_resupply (sbx.c)
	vehicle_resupply (parser.c)
start_simnet	main (main.c)
start_ticking	remote_start_ticking (remote.c)
	composite_start_ticking (composite.c)
	saf_vehicle_start_ticking (saf_vehicle.c)
startup	pve_checkvis (pve_checkvis.c)
stat	parse_servfile (vis.c)
stationpoint	followvehicle (driver.c)
	pilot_stationpoint (pilot.c)
	mission_stationpoint (driver.c)
stealth_send_attach_to_vehicle	sbx_attach_stealth (sbx.c)
	stealth_attach_to (parser.c)
stealth_send_teleport_to	sbx_attach_stealth (sbx.c)
	stealth_attach_to (parser.c)
	stealth_teleport_to (parser.c)
stop	driver_tick (driver.c)
	followvehicle (driver.c)
stop_ticking	remote_deactivate (remote.c)
store_hit	check_edge_hit (pve_checkvis.c)
	check_tree_hit (pve_checkvis.c)
	check_object (pve_checkvis.c)
strcasecmp	ht_gethostbyname (ht_getnamadr.c)
	hostalias (res_query.c)
strcat	clnt_spcreateerror (clnt_perror.c)
	create_bridges (bridges.c)
	create_buildings (trees.c)
	create_contours (trees.c)
	create_lakes (bridges.c)
	create_river_ints (rivers.c)
	create_rivers (rivers.c)
	create_road_ints (roads.c)
	create_roads (roads.c)
	create_tree_canopies (trees.c)
	create_trees (trees.c)
	init_terrain_stuff (main.c)
	read_quadtree (terrain.c)

Called Function	Calling Function (filename)
strcat (cont.)	syslog (syslog.c)
	tzload (ctime.c)
strcmp	activate_new_cis (cis.c)
	AssocReadParams (params.c)
	get_symbol (symbol.c)
	next_serv (vis.c)
	parse_servfile (vis.c)
	ProcessProtocolFamily (params.c)
	tdb_right_format (version.c)
strcpy	clnt_sperror (clnt_perror.c)
	cm_execute_overlay (cm.c)
	create_bridges (bridges.c)
	create_buildings (trees.c)
	create_contours (trees.c)
	create_lakes (bridges.c)
	create_river_ints (rivers.c)
	create_rivers (rivers.c)
	create_road_ints (roads.c)
	create_roads (roads.c)
	create_tree_canopies (trees.c)
	create_trees (trees.c)
	cvt (doprnt.c)
	dstsetkernel (ctime.c)
	get_string (symbol.c)
	get_symbol (symbol.c)
	init_terrain_stuff (main.c)
	net_stat_string (net_stat.c)
	open_cmc (net_open.c)
	print_str (perror.c)
	read_quadtree (terrain.c)
	res_init (res_init.c)
	syslog (syslog.c)
	tdb_error (error.c)
	tdb_get_db_name (dump.c)
	tzload (ctime.c)
	tzsetgmt (ctime.c)
	tzsetkernel (ctime.c)
string_for_altitude_type	sbx_altitude (sbx.c)
string_for_attack_type	sbx_attack (sbx.c)
string_for_hold_type	sbx_hold (sbx.c)
	sbx_goto_point (sbx.c)
stringcopy	CommandLog (par_hist.c)
strlen	_doprnt (doprnt.c)
	change_movement_and_report (cis.c)
	clnt_spcreateerror (clnt_perror.c)
	clnt_sperror (clnt_perror.c)
	commander_get_mission_status (commander.c)
	composite_generate_status_report (composite.c)
	cvt (doprnt.c)
	fill_generic_cis_report (cis.c)
	get_string (symbol.c)

Called Function	Calling Function (filename)
strlen (cont.)	get_symbol (symbol.c)
	getanswer (dns_getnamadr.c)
	identify_version (version.c)
	net_open (net_open.c)
	print_break_set_entry (breakset.c)
	print_str (perror.c)
	res_querydomain (res_query.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
	simnet_getstats (simnet.c)
	syslog (syslog.c)
	tzload (ctime.c)
	vehicle_status_string (saf_vehicle.c)
	xdr_string (xdr.c)
strncmp	doit (yp_innetgr.c)
	_findenv (getenv.c)
	res_init (res_init.c)
strncpy	hostalias (res_query.c)
	openlog (syslog.c)
	print_str (perror.c)
	res_init (res_init.c)
	tdb_get_tdb_info (tdb_init.c)
	timezone (timezone.c)
strpbrk	doit (yp_gnetgr.c)
strsave	ht_getnetgrent (ht_gngrent.c)
strtok	parser_init (parser.c)
	AssocReadParams (params.c)
trtol	interpret (yp_getpwent.c)
stty	tty_restore_modes (setup.c)
	tty_setup_modes (setup.c)
syslog	clnt_pcreateerror (clnt_perror.c)
	clnt_perrno (clnt_perror.c)
	clntudp_bufcreate (clnt_udp.c)
	doit (yp_gnetgr.c)
	rpc_perror (clnt_perror.c)
	rpc_perror1 (clnt_perror.c)
	rpc_perror2 (clnt_perror.c)
tag_error	find_tag (tags.c)
	find_tag_sorted (tags.c)
tan	create_missile (missile.c)
target_in_position	update_target_list (targeting.c)
target_priority	target_type_ok (targeting.c)
target_type_ok	update_target_list (targeting.c)
targeting_set_fire_at_pointair	pilot_start_hoverattack (pilot.c)
targeting_set_hold_fire	pilot_start_hoverattack_egress (pilot.c)
targeting_set_parameters	saf_vehicle_set_targeting_parameters (saf_vehicle.c)
targeting_show	saf_vehicle_show (saf_vehicle.c)
targeting_tick	saf_vehicle_tick (saf_vehicle.c)
tdb_cache_disable	tdb_init_memory (memory.c)
tdb_error	report_error_from_tdb_once (misc.c)
	test_maxima (pve_checkvis.c)
	init_terrain_stuff (main.c)

Called Function	Calling Function (filename)
tdb_get_db_format	tdb_error (error.c)
	tdb_right_format (version.c)
tdb_get_db_name	init_terrain_stuff (main.c)
tdb_get_gl	create_entity (entity.c)
	ground_level (pilot.c)
	parser_create_vehicle (parser.c)
	generate_indirect_fire_pkt (pro_sim.c)
tdb_get_hull_to_world	tdb_place_vehicle (h_to_w.c)
tdb_get_tdb_info	tdb_init_memory (memory.c)
	tdb_init_cache (tdb_init.c)
tdb_get_terrain	tdb_get_z (elevation.c)
	test_maxima (pve_checkvis.c)
tdb_get_z	clip_to_tdb (pve_checkvis.c)
	fly_missile (missile.c)
	get_soil_type (misc.c)
	sort_trim_hits (pve_checkvis.c)
	stealth_teleport_to (parser.c)
	tdb_get_gl (misc.c)
	tdb_get_zl (misc.c)
	tracks_set_support_plane (h_to_w.c)
tdb_get_zl	z_velocity (pilot.c)
tdb_giv_xy_get_utm	change_movement_and_report (cis.c)
	composite_generate_status_report (composite.c)
	fill_generic_cis_report (cis.c)
	saf_vehicle_generate_status_report (saf_vehicle.c)
	send_contact_report (reporter.c)
	send_shelling_report (reporter.c)
	send_spot_report (reporter.c)
tdb_init_cache	init_terrain_stuff (main.c)
tdb_init_memory	init_terrain_stuff (main.c)
tdb_init_patch_guards	init_terrain_stuff (main.c)
tdb_map_xy_to_utm	tdb_giv_xy_get_utm (map.c)
tdb_place_vehicle	pilot_start_landed (pilot.c)
	groundveh_move (groundveh.c)
	saf_vehicle_teleport (saf_vehicle.c)
tdb_read_header	tdb_get_tdb_info (tdb_init.c)
tdb_right_format	tdb_init_memory (memory.c)
	tdb_init_cache (tdb_init.c)
tdb_set_dumpfile	tdb_init_memory (memory.c)
	tdb_init_cache (tdb_init.c)
terrain	patch (pve_checkvis.c)
terrain_cache_inquire	tdb_get_terrain (get_patch.c)
terrain_memory_inquire	tdb_get_terrain (get_patch.c)
test_clutter	pve_checkvis (pve_checkvis.c)
test_maxima	pve_checkvis (pve_checkvis.c)
test_mins	pve_checkvis (pve_checkvis.c)
tickable_note_start_tick	saf_vehicle_tick (saf_vehicle.c)
	remote_tick (remote.c)
time	syslog (syslog.c)
top_left_corner	left_column (minimum_clip.c)
track_target	fly_round (weapons.c)
	gunner_tick (gunner.c)

Called Function	Calling Function (filename)
tracks_calc_unit_normal	tracks_set_support_plane (h_to_w.c)
tracks_set_support_plane	tdb_get_hull_to_world (h_to_w.c)
treelines	canopies (pve_checkvis.c)
	patch (pve_checkvis.c)
trees	patch (pve_checkvis.c)
tty_exit	parser_restore_term (parser.c)
tty_getchar	tty_tick (interface.c)
tty_malloc_break_set	tty_setup_parser_break_set (geyer.c)
tty_parser_init	parser_init (parser.c)
tty_restore_modes	tty_exit (interface.c)
tty_setup_modes	tty_parser_init (geyer.c)
tty_setup_parser_break_set	tty_parser_init (geyer.c)
turret_firepower_kill	saf_vehicle_firepower_kill (saf_vehicle.c)
turret_interest_dir	compute_interest_direction (detection.c)
turret_muzzle_position_in_world_coordinates	muzzle_position_in_world_coordinates (weapons.c)
turret_point_at_target	point_weapon_at_target (gunner.c)
turret_scan	scan_weapon (gunner.c)
turret_set_scan_parms	composite_assume_formation_internal (composite.c)
turret_show	saf_vehicle_show (saf_vehicle.c)
turret_slew	turret_point_at_target (turret.c)
	turret_scan (turret.c)
tzload	tzset (ctime.c)
tzset	localtime (ctime.c)
tzsetgmt	tzset (ctime.c)
tzsetkernel	tzset (ctime.c)
uc	ParseMatch (par_util.c)
udp_close	rudp_close (rel_conn.c)
udp_open	rudp_open (rel_conn.c)
udp_read	rudp_read_message (rel_conn.c)
udp_write	rudp_send (rel_conn.c)
	udp_read (udp_berkeley.c)
uidof	matchuid (yp_getpwent.c)
unget_locks	open_cmc (net_open.c)
ungetc	_doscan (doscan.c)
	_instr (doscan.c)
	_innum (doscan.c)
unhash_saf_id	composite_go_away (composite.c)
	remote_go_away (remote.c)
	saf_vehicle_go_away (saf_vehicle.c)
unit_send_pae_data	unit_send_pae_data (composite.c)
	create_unit (create.c)
unmap_buffers	open_cmc (net_open.c)
upcase	parser_init (parser.c)
update_cluster_com	add_vehicle_to_cluster (reporter.c)
	update_clusters (reporter.c)
update_cluster_heading	update_clusters (reporter.c)
update_clusters	reporter_tick (reporter.c)
update_grid_entry_list	saf_vehicle_tick (saf_vehicle.c)
	remote_tick (remote.c)
	saf_vehicle_teleport (saf_vehicle.c)
update_reporter_vehicles	reporter_tick (reporter.c)
update_target_list	targeting_tick (targeting.c)

Called Function	Calling Function (filename)
upshift	AssocReadParams (params.c)
	ProcessProtocolFamily (params.c)
usingypmap	match (yp_gnetgr.c)
v1filter	v1dofirst (yp_enum.c)
v1prot_dofirst	v1dofirst (yp_enum.c)
v1prot_donext	v1donext (yp_enum.c)
valid_quad	get_quad_nodes_internal (search.c)
vec2_add	followvehicle (driver.c)
	get_next_circle_point (pilot.c)
	p_follower_find_pnt_in_turn (p_follower.c)
	p_follower_flip_in_turn (p_follower.c)
	p_follower_fly_in_coord_pos (p_follower.c)
	p_follower_get_wrld_offset (p_follower.c)
	p_follower_pnt_in_hard_turn (p_follower.c)
	p_follower_pnt_in_shallow_turn (p_follower.c)
	project_point (pilot.c)
	stationpoint (driver.c)
vec2_copy	cm_add_route (cm.c)
	commander_attack (commander.c)
	compute_situation (pilot.c)
	get_leader_state (driver.c)
	get_next_circle_point (pilot.c)
	ground_level (pilot.c)
	p_follower_find_pnt_in_turn (p_follower.c)
	p_follower_flip_in_turn (p_follower.c)
	p_follower_get_wrld_offset (p_follower.c)
	p_follower_new_rel_rt_pnt (p_follower.c)
	p_follower_pnt_in_shallow_turn (p_follower.c)
	p_follower_set_desired_vel (p_follower.c)
	p_follower_stop_coord (p_follower.c)
	pilot_init_followroute (pilot.c)
	pilot_init_hoverattack (pilot.c)
	pilot_is_facing_direction (pilot.c)
	pilot_orbithold (pilot.c)
	pilot_start_gotopoint (pilot.c)
	plane_tick (plane.c)
	project_point (pilot.c)
	stationpoint (driver.c)
	vec2_rotate (pilot.c)
vec2_dot	create_new_cluster (reporter.c)
	enemy_disposition (predicates.c)
	exagerate_bridges (route.c)
	followvehicle (driver.c)
	get_leader_state (driver.c)
	get_unit_direction (driver.c)
	p_follower_leader_passed_point (p_follower.c)
	p_follower_passed_point (p_follower.c)
	pilot_goto_endpoint (pilot.c)
	pilot_gotopoint (pilot.c)
	pilot_point_at_target (pilot.c)
	re_assign_routepoints (route.c)

Called Function	Calling Function (filename)
vec2_dot (cont.)	remote_next_road_point (remote.c)
	update_cluster_heading (reporter.c)
vec2_init	cm_get_center_of_mass (navigator.c)
	p_follower_new_rel_rt_pnt (p_follower.c)
	p_follower_set_desired_vel (p_follower.c)
	p_follower_stop_coord (p_follower.c)
	re_assign_routepoints (route.c)
vec2_mag	get_leader_state (driver.c)
	get_unit_direction (driver.c)
	p_follower_arrive_at_same_time (p_follower.c)
	p_follower_fly_in_coord_pos (p_follower.c)
	p_follower_gen_coord_goal_pnt (p_follower.c)
	vel2point (driver.c)
vec2_mag2	are_we_there (pilot.c)
	commander_attack (commander.c)
	p_follower_find_pnt_in_turn (p_follower.c)
	p_follower_flip_in_turn (p_follower.c)
	p_follower_flying_independent (p_follower.c)
	p_follower_gen_coord_goal_pnt (p_follower.c)
	pilot_point_at_target (pilot.c)
vec2_norm	commander_attack (commander.c)
	enemy_disposition (predicates.c)
	followvehicle (driver.c)
	p_follower_find_pnt_in_turn (p_follower.c)
	p_follower_flip_in_turn (p_follower.c)
	p_follower_fly_in_coord_pos (p_follower.c)
	p_follower_get_wrld_offset (p_follower.c)
	p_follower_set_desired_vel (p_follower.c)
	plane_tick (plane.c)
	set_speed_dir (driver.c)
	vehicle_facing_point (pilot.c)
vec2_range_squared	action_drill_needed (predicates.c)
	cluster_shell (reporter.c)
	cluster_vehicle (reporter.c)
	hasty_attack_needed (predicates.c)
	issue_reports (reporter.c)
	need_decluster (reporter.c)
	update_clusters (reporter.c)
vec2_rotate	get_next_circle_point (pilot.c)
vec2_scale	cm_get_center_of_mass (navigator.c)
	followvehicle (driver.c)
	get_next_circle_point (pilot.c)
	p_follower_arrive_at_same_time (p_follower.c)
	p_follower_fly_in_coord_pos (p_follower.c)
	p_follower_gen_coord_goal_pnt (p_follower.c)
	p_follower_leader_passed_point (p_follower.c)
	p_follower_passed_point (p_follower.c)
	p_follower_pnt_in_hard_turn (p_follower.c)
	p_follower_set_desired_vel (p_follower.c)
	pilot_goto_endpoint (pilot.c)
	pilot_gotopoint (pilot.c)

Called Function	Calling Function (filename)
vec2_scale (cont.)	plane_tick (plane.c)
	project_point (pilot.c)
	xy_dir_and_range (pilot.c)
	z_velocity (pilot.c)
vec2_set	pilot_init_hoverattack (pilot.c)
	commander_attack (commander.c)
vec2_sub	are_we_there (pilot.c)
	commander_attack (commander.c)
	followvehicle (driver.c)
	p_follower_find_pnt_in_turn (p_follower.c)
	p_follower_flip_in_turn (p_follower.c)
	p_follower_fly_in_coord_pos (p_follower.c)
	p_follower_flying_independent (p_follower.c)
	p_follower_gen_coord_goal_pnt (p_follower.c)
	p_follower_get_wrld_offset (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
	p_follower_landhold_tick (p_follower.c)
	p_follower_leader_passed_point (p_follower.c)
	p_follower_passed_point (p_follower.c)
	p_follower_set_desired_vel (p_follower.c)
	pilot_init_land (pilot.c)
	pilot_is_facing_direction (pilot.c)
	re_assign_routepoints (route.c)
	vehicle_facing_point (pilot.c)
	vel2point (driver.c)
	xy_dir_and_range (pilot.c)
vec2_veh2world	p_follower_get_wrld_offset (p_follower.c)
	p_follower_pnt_in_shallow_turn (p_follower.c)
	stationpoint (driver.c)
vec2_world2veh	enemy_disposition (predicates.c)
	followvehicle (driver.c)
vec_add	add_shelling_to_cluster (reporter.c)
	compute_explosion_point (missile.c)
	fire_missile_at_target (missile.c)
	fly_missile (missile.c)
	missile_set_desired_direction (missile.c)
	muzzle_position_in_world_coordinates (weapons.c)
	saf_vehicle_est_position (saf_vehicle.c)
	start_disengaging (collision.c)
	turret_muzzle_position_in_world_coordinates (turret.c)
	update_cluster_com (reporter.c)
	z_velocity (pilot.c)
vec_copy	collision_vehicle_rammed (collision.c)
	composite_tick (composite.c)
	compute_interest_direction (detection.c)
	create_entity (entity.c)
	create_new_cluster (reporter.c)
	create_new_shelling (reporter.c)
	create_pilot (pilot.c)
	create_saf_vehicle (saf_vehicle.c)
	detect_collision_tick (collision.c)

Called Function	Calling Function (filename)
vec_copy (cont.)	detect_imminent_collision_tick (collision.c)
	enemy_disposition (predicates.c)
	fire_missile_at_target (missile.c)
	fire_weapon_at_target (weapons.c)
	fly_missile (missile.c)
	followvehicle (driver.c)
	generate_indirect_fire_pkt (pro_sim.c)
	generate_weapon_miss (weapons.c)
	get_soil_type (misc.c)
	get_unit_direction (driver.c)
	p_follower_hoverhold (p_follower.c)
	pilot_init_hoverattack (pilot.c)
	pilot_orbithold (pilot.c)
	pilot_point_at_target (pilot.c)
	pilot_tick (pilot.c)
	remote_tick (remote.c)
	saf_vehicle_stop_mimicing (saf_vehicle.c)
	saf_vehicle_tick (saf_vehicle.c)
	sbx_attach_stealth (sbx.c)
	send_contact_report (reporter.c)
	send_spot_report (reporter.c)
	set_speed_dir (driver.c)
	shoot_target (gunner.c)
	start_disengaging (collision.c)
	stealth_attach_to (parser.c)
	stop (driver.c)
	update_clusters (reporter.c)
	vec2_veh2world (driver.c)
	vec2_world2veh (driver.c)
	vel2point (driver.c)
vec_cross_prod	tracks_calc_unit_normal (h_to_w.c)
	tdb_get_hull_to_world (h_to_w.c)
vec_dot_prod	compute_explosion_point (missile.c)
	detectable (detection.c)
	fly_missile (missile.c)
	intervis_get_view (intervis.c)
	missile_ground_impact (missile.c)
	missile_maybe_hit_target (missile.c)
	missile_set_desired_direction (missile.c)
	p_follower_pnt_in_hard_turn (p_follower.c)
	target_type_ok (targeting.c)
vec_init	air_tick (flyingveh.c)
	airveh_init (flyingveh.c)
	are_we_there (pilot.c)
	create_driver (driver.c)
	create_entity (entity.c)
	create_missile (missile.c)
	create_pilot (pilot.c)
	create_targeting (targeting.c)
	groundveh_tick (groundveh.c)
	helo_tick (helo.c)

Called Function	Calling Function (filename)
vec_init (cont.)	init_pilot_state_machine (pilot.c)
	missile_deactivate (missile.c)
	pilot_hoverattack_complete (pilot.c)
	pilot_hoverhold (pilot.c)
	pilot_start_hoverattack (pilot.c)
	pilot_start_landed (pilot.c)
	start_disengaging (collision.c)
	update_cluster_com (reporter.c)
vec_mat_mul	muzzle_position_in_world_coordinates (weapons.c)
	tdb_get_hull_to_world (h_to_w.c)
	tracks_set_support_plane (h_to_w.c)
	turret_interest_dir (turret.c)
	turret_muzzle_position_in_world_coordinates (turret.c)
	vec_z_rotate (math.c)
vec_normalize	fire_missile_at_target (missile.c)
	fire_weapon_at_target (weapons.c)
	fly_missile (missile.c)
	get_impact_and_trajectory (impact.c)
	get_leader_state (driver.c)
	missile_maybe_hit_target (missile.c)
	missile_set_desired_direction (missile.c)
	start_avoiding_house (collision.c)
	start_disengaging (collision.c)
	stationpoint (driver.c)
	tracks_calc_unit_normal (h_to_w.c)
vec_scale	add_shelling_to_cluster (reporter.c)
	attackatwill_tick (pilot.c)
	compute_explosion_point (missile.c)
	fire_missile_at_target (missile.c)
	fire_weapon_at_target (weapons.c)
	fly_missile (missile.c)
	followvehicle (driver.c)
	get_i_and_t_from_normal (impact.c)
	groundveh_move (groundveh.c)
	missile_set_desired_direction (missile.c)
	pilot_hoverattack (pilot.c)
	pilot_orbithold (pilot.c)
	pilot_point_at_target (pilot.c)
	saf_vehicle_est_position (saf_vehicle.c)
	tdb_get_hull_to_world (h_to_w.c)
	update_cluster_com (reporter.c)
vec_set	create_driver (driver.c)
	driver_face_direction_cmd (driver.c)
	driver_follow_vehicle_cmd (driver.c)
	driver_goto_point_cmd (driver.c)
	init_pilot_state_machine (pilot.c)
	p_follower_hoverhold (p_follower.c)
	pilot_face_direction (pilot.c)
	pilot_landing (pilot.c)
	pilot_start_takingoff (pilot.c)
	pilot_takingoff (pilot.c)
	vel2point (driver.c)

Called Function	Calling Function (filename)
vec_sub	compute_explosion_point (missile.c)
	enemy_disposition (predicates.c)
	exagerate_bridges (route.c)
	fire_missile_at_target (missile.c)
	fire_weapon_at_target (weapons.c)
	fly_missile (missile.c)
	get_impact_and_trajectory (impact.c)
	missile_ground_impact (missile.c)
	missile_maybe_hit_target (missile.c)
	missile_set_desired_direction (missile.c)
	pilot_point_at_target (pilot.c)
	remote_next_road_point (remote.c)
	routepoint_distance (route.c)
	start_avoiding_house (collision.c)
	start_disengaging (collision.c)
	tracks_calc_unit_normal (h_to_w.c)
	tracks_set_support_plane (h_to_w.c)
	turret_point_at_target (turret.c)
	update_cluster_heading (reporter.c)
vec_z_rotate	start_avoiding_collision (collision.c)
	groundveh_move (groundveh.c)
vehicle_bong	vehicle_bong_do (parser.c)
vehicle_component_name	database_df_damage_query (df_damage.c)
vehicle_facing_point	p_follower_flying_independent (p_follower.c)
vehicle_fill_in_appearance_data	saf_vehicle_fill_in_appearance_data (saf_vehicle.c)
	remote_fill_in_appearance_data (remote.c)
vehicle_fill_in_position_data	saf_vehicle_fill_in_position_data (saf_vehicle.c)
	remote_fill_in_position_data (remote.c)
vehicle_impact_to_sbx	protocol_sim_process (simnet.c)
vehicle_iterator_once_next	sbx_connection_stat_pulse (sbx.c)
ehicle_iterator_reset	sbx_connection_reset_msg (sbx.c)
	sbx_connection_synch_received (sbx.c)
vehicle_kill	sbx_connection_generate_arty_msg (sbx.c)
	vehicle_catastrophe (parser.c)
vehicle_kill_remote	vehicle_kill (pro_sim.c)
	vehicle_bong (pro_sim.c)
vehicle_list_append	update_clusters (reporter.c)
vehicle_manager_count	count_vehicles (parser.c)
vehicle_manager_count_force	count_forces (parser.c)
vehicle_manager_count_hosts	count_hosts (parser.c)
vehicle_manager_count_sites	count_sites (parser.c)
vehicle_manager_print	show_vehicle_ids (parser.c)
vehicle_ping	vehicle_ping_do (parser.c)
vehicle_reappeared	update_reporter_vehicles (reporter.c)
vehicle_show	saf_vehicle_show (saf_vehicle.c)
	remote_show (remote.c)
vehicle_status_string	saf_vehicle_generate_status_report (saf_vehicle.c)
	saf_vehicle_show (saf_vehicle.c)
vehicle_wall_name	database_df_damage_query (df_damage.c)
vel2point	followvehicle (driver.c)
	driver_tick (driver.c)
vfork	syslog (syslog.c)

Called Function	Calling Function (filename)
vis_nextserv	endnetent (getnetent.c)
	endnetgrent (getnetgrent.c)
	endprotoent (getprotoent.c)
	endservent (getservent.c)
	gethostbyaddr (gethostnamadr.c)
	gethostbyname (gethostnamadr.c)
	gethostent (gethostnamadr.c)
	getnetent (getnetent.c)
	getnetgrent (getnetgrent.c)
	getprotoent (getprotoent.c)
	getservbyname (getservent.c)
	getservbyport (getservent.c)
	getservent (getservent.c)
	innetgr (innetgr.c)
	setnetent (getnetent.c)
	setnetgrent (getnetgrent.c)
	setprotoent (getprotoent.c)
	setservent (getservent.c)
wait	start_simnet (simnet.c)
	syslog (syslog.c)
wait_for_empty_ring_element	send_cmc (net_osend.c)
	send_cmc_8023 (net_send.c)
wait_until	invoke_functions_until (sched.c)
warn_about_point_space	read_terrain_files (terrain.c)
water_check	groundveh_tick (groundveh.c)
weapon_configure	create_weapon_systems (weapons.c)
weapon_deselect	loader_tick (loader.c)
weapon_load	loader_tick (loader.c)
weapon_priority_list_show	weapon_systems_show (weapons.c)
weapon_select	loader_tick (loader.c)
weapon_show	weapon_systems_show (weapons.c)
weapon_state_to_string	weapon_show (weapons.c)
weapon_systems_checkpoint	saf_vehicle_checkpoint_state (saf_vehicle.c)
weapon_systems_rearm	saf_vehicle_fake_resupply (saf_vehicle.c)
weapon_systems_reinit	saf_vehicle_reinit (saf_vehicle.c)
weapon_systems_show	saf_vehicle_show (saf_vehicle.c)
weapon_unload	loader_tick (loader.c)
which_event_ring	set_critical_performance_level (perf.c)
	insert_periodic_function (maint.c)
	remove_function (maint.c)
which_side	angle_between_vectors (math.c)
	choose_skirt_point (collision.c)
	compute_damage_keys (df_damage.c)
	start_avoiding_collision (collision.c)
within_delta	driver_tick (driver.c)
	followvehicle (driver.c)
write	_flsbuf (flsbuf.c)
	fflush (flsbuf.c)
	pflush (perror.c)
	res_send (res_send.c)
	syslog (syslog.c)

Called Function	Calling Function (filename)
write_buffer_all_sbx	broadcast_appearance_data (sbx.c)
	broadcast_pae_data (sbx.c)
	broadcast_veh_is_gone (sbx.c)
	gasp (main.c)
write_string_to_tty	print_break_set_entry (breakset.c)
xdr_bytes	xdr_datum (ypxdr.c)
	xdr_netobj (xdr.c)
	xdr_opaque_auth (rpc_prot.c)
xdr_callhdr	clntudp_bufcreate (clnt_udp.c)
xdr_datum	xdr_ypreq_key (ypxdr.c)
	xdr_ypresp_key_val (ypxdr.c)
	xdr_ypresp_val (ypxdr.c)
xdr_enum	xdr_accepted_reply (rpc_prot.c)
	xdr_callhdr (rpc_prot.c)
	xdr_opaque_auth (rpc_prot.c)
	xdr_rejected_reply (rpc_prot.c)
	xdr_replymsg (rpc_prot.c)
	xdr_union (xdr.c)
xdr_int	xdr_char (xdr.c)
xdr_long	xdr_int (xdr.c)
	xdr_enum (xdr.c)
xdr_opaque	xdr_bytes (xdr.c)
	xdr_des_block (rpc_prot.c)
	xdr_string (xdr.c)
	xdr_yp_binding (ypxdr.c)
	xdr_yp_inaddr (ypxdr.c)
xdr_opaque_auth	authnone_create (auth_none.c)
	clntudp_call (clnt_udp.c)
	xdr_accepted_reply (rpc_prot.c)
xdr_replymsg	clntudp_call (clnt_udp.c)
xdr_string	xdr_wrapstring (xdr.c)
	xdr_ypdomain_wrap_string (ypxdr.c)
	xdr_ypmap_wrap_string (ypxdr.c)
	xdr_ypowner_wrap_string (ypxdr.c)
xdr_u_int	xdr_u_char (xdr.c)
	xdr_bytes (xdr.c)
	xdr_string (xdr.c)
xdr_u_long	xdr_accepted_reply (rpc_prot.c)
	xdr_callhdr (rpc_prot.c)
	xdr_deskey (rpc_prot.c)
	xdr_pmap (pmap_prot.c)
	xdr_rejected_reply (rpc_prot.c)
	xdr_replymsg (rpc_prot.c)
	xdr_u_int (xdr.c)
	xdr_ypmap_parms (ypxdr.c)
	xdr_ypresp_key_val (ypxdr.c)
	xdr_ypresp_val (ypxdr.c)
xdr_union	_xdr_yprequest (ypv1_xdr.c)
	_xdr_ypresponse (ypv1_xdr.c)
	xdr_replymsg (rpc_prot.c)
	xdr_ypbind_resp (ypxdr.c)
xdr_yp_binding	_xdr_ypbind_oldsetdom (ypv1_xdr.c)

Called Function	Calling Function (filename)
xdr_yp_inaddr	xdr_yp_binding (ypxdr.c)
xdr_ypdomain_wrap_string	_xdr_ypbind_oldsetdom (ypv1_xdr.c)
	xdr_ypmap_parms (ypxdr.c)
	xdr_ypreq_key (ypxdr.c)
	xdr_ypreq_nokey (ypxdr.c)
xdr_ypmap_wrap_string	xdr_ypreq_key (ypxdr.c)
	xdr_ypreq_nokey (ypxdr.c)
	xdr_ypmap_parms (ypxdr.c)
xdr_ypowner_wrap_string	xdr_ypmap_parms (ypxdr.c)
xdrmem_create	clntudp_bufcreate (clnt_udp.c)
	clntudp_call (clnt_udp.c)
	authnone_create (auth_none.c)
xy_dir_and_range	compute_situation (pilot.c)
	get_next_circle_point (pilot.c)
	pilot_orbithold (pilot.c)
	project_point (pilot.c)
yellowup	yp_getservbyport (yp_getservent.c)
	yp_sethostent (yp_gethostent.c)
	yp_setnetent (yp_gnetent.c)
	yp_setprotoent (yp_getpent.c)
	yp_setrpcent (yp_getrpcent.c)
yp_bind	yp_setgrent (yp_getgrent.c)
yp_endgrent	yp_getgrgid (yp_getgrent.c)
yp_endhostent	yp_gethostbyname (yp_gethostent.c)
yp_endnetent	yp_getnetbyaddr (yp_gnetent.c)
yp_endnetgrent	yp_setnetgrent (yp_gnetgr.c)
yp_endprotoent	yp_getprotobynumber (yp_getpent.c)
yp_endpwent	yp_getpwnam (yp_getpwent.c)
yp_endrpcent	yp_getrpcbynumber (yp_getrpcent.c)
yp_endservent	yp_getservbyport (yp_getservent.c)
yp_first	yp_getnetent (yp_gnetent.c)
yp_getgrent	yp_getgrgid (yp_getgrent.c)
yp_getpwent	yp_getpwnam (yp_getpwent.c)
yp_getrpcent	yp_getrpcbyname (yp_getrpcent.c)
yp_getservent	yp_getservbyport (yp_getservent.c)
yp_match	yp_gethostbyaddr (yp_gethostent.c)
yp_next	yp_getnetent (yp_gnetent.c)
yp_setgrent	yp_getgrgid (yp_getgrent.c)
yp_sethostent	yp_gethostbyname (yp_gethostent.c)
yp_setnetent	yp_getnetbyaddr (yp_gnetent.c)
yp_setprotoent	yp_getprotobynumber (yp_getpent.c)
yp_setpwent	yp_getpwnam (yp_getpwent.c)
yp_setrpcent	yp_getrpcbynumber (yp_getrpcent.c)
yp_setservent	yp_getservbyport (yp_getservent.c)
yp_unbind	yp_first (yp_enum.c)
ypprot_err	v1domatch (yp_match.c)
yyback	yylook (lexer.c)
yyerror	yylex (lexer.c)
	yyparse (parser.y)
yylex	yyparse (parser.y)
yparse	reader_read_file (parser.y)
yywrap	yylex (lexer.c)

Called Function	Calling Function (filename)
z_velocity	attackatwill_tick (pilot.c)
	p_follower_hoverhold (p_follower.c)
	p_follower_hoverhold_tick (p_follower.c)
	p_follower_landhold_tick (p_follower.c)
	p_follower_set_desired_vel (p_follower.c)
	pilot_goto_endpoint (pilot.c)
	pilot_gotopoint (pilot.c)
	pilot_hoverattack (pilot.c)
	pilot_hoverattack_complete (pilot.c)
	pilot_hoverhold (pilot.c)
	pilot_orbithold (pilot.c)
zero_perf_stats	collect_perf_stat (perf.c)
	perf_monitor_on (perf.c)

APPENDIX D

The following is a list of globally accessable user defined types and the files in which they are defined.

User Defined Type	Where Typedef Defined
*FUNC_PTR	sim_style.h
*PFV	sim_types.h
*POINTER	sim_style.h
*STRING	mass_stdc.h
ActivateReason	p_sim.h
ActivateRequestVariant	p_sim.h
ActivateResponseVariant	p_sim.h
ActivateResult	p_sim.h
address_entry	pnl_tst_dfn.h
AirframeSubsystems	status.h
AirVehicleSubsystems	status.h
Angle	basic.h
AssociationDataUnit	p_assoc.h
AssociationPDU	p_assoc.h
AssociationPDUKind	p_assoc.h
AssociationProtocolVersion	p_assoc.h
AssociationUserProtocol	p_assoc.h
AttachedVariant	p_stlth.h
AttachVariant	p_stlth.h
B1BBOX2D	dgi_stdg.h
B1BBOX3D	dgi_stdg.h
B1HSL	dgi_stdg.h
B1HSLO	dgi_stdg.h
B1MTX4X3	dgi_stdg.h
B1MTX4X4	dgi_stdg.h
B1P2D	dgi_stdg.h
B1P3D	dgi_stdg.h
B1P4D	dgi_stdg.h
B1RGB	dgi_stdg.h
B1RGBO	dgi_stdg.h
B_LEVEL_TYPE	libpfile_dfn.h
BattleScheme	basic.h
bbd_device	bbddev.h
BOOLEAN	sim_types.h
Boolean	basic.h
BOUND_BOX	libfilter.h
BOUND_INFO	libfilter.h
BreachedLaneStatusVariant	p_data.h
BreachedLaneVariant	p_sim.h
BridgePacket	bridge_com.h
BridgePacketHeader	bridge_com.h
BurstDescriptor	basic.h
BVOL_SUPT	if_targets.h
BYTE	sim_types.h
Byte	types.h
CharacterSetType	basic.h
ChassisSubsystems	status.h

User_Defined_Type	Where_Typedef_Defined
cif_cb	libcif.h
cif_connection_table	libcif.h
cif_vec	libcif.h
CockpitSubsystems	status.h
CollisionVariant	p_sim.h
ComponentStatus	p_mgmt.h
CONTROL_IDENTIFIER	libpfile_dfn.h
DamageCause	p_data.h
DamageType	basic.h
data_arg	enparg.h
DataCollectionPDU	p_data.h
DataCollectionPDUKind	p_data.h
DataCollectionProtocolVersion	p_data.h
DeactivateReason	p_sim.h
DeactivateRequestVariant	p_sim.h
DeactivateResponseVariant	p_sim.h
DeactivateResult	p_sim.h
DED_MAP_ENTRY	libmap_dfn.h
device	drdev.h
DiscrepancyThresholds	libapp.h
E_PARAM	sim_types.h
EDGE_TYPE	libpfile_dfn.h
EffectType	p_data.h
ElectronicSubsystems	status.h
elevationRequestPDUKind	p_terra.h
ElevationRequestVariant	p_terra.h
elevationResponsePDUKind	p_terra.h
ElevationResponseVariant	p_terra.h
enp_registers	enparg.h
EquipStatusVariant	p_mgmt.h
ErrorReportSeverity	p_mgmt.h
ErrorReportVariant	p_mgmt.h
EventFlagVariant	p_data.h
EventID	basic.h
exec_req	sersvr.h
ExerciseID	basic.h
ExerciseStatusVariant	p_data.h
ExtNetworkBuffer	network.h
FIFO	fifo_dfn.h
FireResult	p_sim.h
FireType	p_sim.h
FireVariant	p_sim.h
ForceID	basic.h
FOV	if_header_struct.h
FOV	sim_cig_if.h
G_EDGE_TYPE	libpfile_dfn.h
GenericVehicleStatus	status.h
getstat_arg	enparg.h
gettime_arg	enparg.h
GridZone	map.h
GroundVehicleSubsystems	status.h
HASH_ENTRY	rva_lcc.h Vehicles CSCI SDD

User Defined Type	Where Typedef Defined
HASH_TABLE	rva_lcc.h Vehicles CSCI SDD
HOST_INFO	libfilter.h
HWORD	mass_stdc.h
I2BBOX2D	dgi_stdg.h
I2BBOX3D	dgi_stdg.h
I2HSL	dgi_stdg.h
I2HSLO	dgi_stdg.h
I2MTX4X3	dgi_stdg.h
I2MTX4X4	dgi_stdg.h
I2P2D	dgi_stdg.h
I2P3D	dgi_stdg.h
I2P4D	dgi_stdg.h
I2RGB	dgi_stdg.h
I2RGBO	dgi_stdg.h
I4BBOX2D	dgi_stdg.h
I4BBOX3D	dgi_stdg.h
I4HSL	dgi_stdg.h
I4HSLO	dgi_stdg.h
I4MTX4X3	dgi_stdg.h
I4MTX4X4	dgi_stdg.h
I4P2D	dgi_stdg.h
I4P3D	dgi_stdg.h
I4P4D	dgi_stdg.h
I4RGB	dgi_stdg.h
I4RGBO	dgi_stdg.h
IDC_ENTRY	*idc*_dfn.h
ImpactVariant	p_sim.h
IndirectFireDetonation	p_sim.h
IndirectFireVariant	p_sim.h
INT_2	mass_stdc.h
INT_4	mass_stdc.h
key_t	sys/types.h
LaserRangeFinder	p_data.h
LaserRangeResult	p_data.h
LaserRangeVariant	p_data.h
LIGHT_TYPE	libpfile_dfn.h
liopb	serioctl.h
LoggerActivity	p_logger.h
LoggerAvailReplyVariant	p_logger.h
LoggerClockTickVariant	p_logger.h
LoggerCommandVariant	p_logger.h
LoggerInformationVariant	p_logger.h
LoggerMedium	p_logger.h
LoggerOperation	p_logger.h
LoggerPDU	p_logger.h
LoggerPDUKind	p_logger.h
LoggerProtocolVersion	p_logger.h
LoggerRTCType	p_logger.h
LoggerSeekRTC	p_logger.h
LoggerSpeedRTC	p_logger.h
LoggerStartRTC	p_logger.h
LoggerState	p_logger.h

User_Defined_Type	Where_Typedef_Defined
LoggerStatusReplyVariant	p_logger.h
LoggerTime	p_logger.h
LongPt	longpt.h
low_stuff	enparg.h
LT_BVOC_ENTRY	sim_cig_if.h
LT_POLY_ENTRY	sim_cig_if.h
M_LEVEL_TYPE	libpfile_dfn.h
ManagementPDU	p_data.h
ManagementPDUKind	p_mgmt.h
ManagementProtocolVersion	p_mgmt.h
MapCoordinates	map.h
MarkerDescriptor	p_sim.h
MarkerVariant	p_sim.h
MarkerVariantNumber	p_sim.h
matrix_device	matrixdev.h
maxElevations	p_terra.h
maxGenericVehicleStores	status.h
mca_arg	enparg.h
MetamorphoseVariant	p_stlth.h
METER_TYPE	libpfile_dfn.h
MimicVariant	p_stlth.h
MineDescriptor	p_data.h
MinefieldVariant	p_data.h
Monitor_PDU	p_mon.h
Monitor_PDUKind	p_mon.h
MonitorKind	p_mon.h
MonitorResponseVariant	p_mon.h
MotiveSubsystems	status.h
MSG-BLK	if_cig_ctl.h
MSG_1ROTATION	sim_cig_if.h
MSG_2D_SETUP	if_init.h
MSG_2D_SETUP	sim_cig_if.h
MSG_3ROTATIONS	sim_cig_if.h
MSG_ADD_TRAJ_TABLE	sim_cig_if.h
MSG_AGL	if_cig2sim.h
MSG_AGL	sim_cig_if.h
MSG_AGL_SETUP	sim_cig_if.h
MSG_AIRVEH_STATE	sim_cig_if.h
MSG_AMMO_DEFINE	sim_cig_if.h
MSG_ASID_OTHERVEH_STATE	sim_cig_if.h
MSG_ASID_SHOW_EFFECT	sim_cig_if.h
MSG_ASID_STATICVEH_STATE	sim_cig_if.h
MSG_CANCEL_ROUND	sim_cig_if.h
MSG_CGN_CONFIGTREE	sim_cig_if.h
MSG_CIG_CTL	if_cig_ctl.h
MSG_CIG_CTL	sim_cig_if.h
MSG_COLLISION	if_cig2sim.h
MSG_CREATE_CONFIG_NODE	sim_cig_if.h
MSG_CREATE_CONFIGNODE	if_init.h
MSG_DELETE_TRAJ_TABLE	sim_cig_if.h
MSG_DR11_PKT_SIZE	if_cig_ctl.h
MSG_DR11_PRT_SIZE	sim_cig_if.h

User Defined Type	Where Typedef Defined
MSG_END	if_cig_ctl.h
MSG_EO	if_phase_out.h
MSG_FILE_DESCR	if_tst_ctl.h
MSG_FILE_DESCR	sim_cig_if.h
MSG_FILE_STATUS	if_tst_ctl.h
MSG_FILE_STATUS	sim_cig_if.h
MSG_FILE_XFER	if_tst_ctl.h
MSG_FILE_XFER	sim_cig_if.h
MSG_GEN_CONFIGTREE	if_init.h
MSG_GENVEH_STATE	if_phase_out.h
MSG_GENVEH_STATE	sim_cig_if.h
MSG_GO	sim_cig_if.h
MSG_GUN_OVERLAY	sim_cig_if.h
MSG_HDR	if_cig_ctl.h
MSG_HDR	sim_cig_if.h
MSG_HIT	if_cig2sim.h
MSG_HIT	sim_cig_if.h
MSG_HIT_RETURN	sim_cig_if.h
MSG_HPRXYZS_MATRIX	sim_cig_if.h
MSG_LASER_RETURN	if_cig2sim.h
MSG_LASER_RETURN	sim_cig_if.h
MSG_LOCAL_TERRAIN	if_cig2sim.h
MSG_LOCAL_TERRAIN	sim_cig_if.h
MSG_LT_PIECE	sim_cig_if.h
MSG_M1VEH-STATE	if_phase_out.h
MSG_M1VEH_STATE	sim_cig_if.h
MSG_M2_VEH_STATE	if_phase_out.h
MSG_M2VEH_STATE	sim_cig_if.h
MSG_MISS	sim_cig_if.h
MSG_OBSCURE	sim_cig_if.h
MSG_OTHERVEH_STATE	sim_cig_if.h
MSG_OVERLAY_SETUP	if_init.h
MSG_OVERLAY_SETUP	sim_cig_if.h
MSG_PASS_BACK	if_cig2sim.h
MSG_PASS_BACK	sim_cig_if.h
MSG_PASS_ON	sim_cig_if.h
MSG_PROCESS_ROUND	sim_cig_if.h
MSG_REQUEST_LASER_RANGE	sim_cig_if.h
MSG_RETURN_POINT_INFO	if_cig2sim.h
MSG_ROT_2x1_MATRIX	sim_cig_if.h
MSG_ROUND_FIRED	sim_cig_if.h
MSG_RTN_LT	if_tst_ctl.h
MSG_RTN_LT	sim_cig_if.h
MSG_RTS4x3_MATRIX	sim_cig_if.h
MSG_SCALE	sim_cig_if.h
MSG_SHOW_EFFECT	sim_cig_if.h
MSG_STATICVEH_REM	sim_cig_if.h
MSG_STATICVEH_STATE	sim_cig_if.h
MSG_SYS_ERROR	if_cig_ctl.h
MSG_SYS_ERROR	sim_cig_if.h
MSG_TARGET_ENVIROMENT	if_cig2sim.h
MSG_TERRAIN_FEFDBACK	if_cig2sim.h

User Defined Type	Where Typedef Defined
MSG_TERRAIN_FEEDBACK_POINT_INIT	if_init.h
MSG_TERRAIN_FEEDBACK_SETUP	if_init.h
MSG_TEST_NAME	if_tst_ctl.h
MSG_TEST_NAME	sim_cig_if.h
MSG_TF_HDR	if_cig2sim.h
MSG_TF_PT	if_cig2sim.h
MSG_TRAJ_CHORD	sim_cig_if.h
MSG_TRAJ_ENTRY	sim_cig_if.h
MSG_TRAJ_ENTRY_XFER	if_init.h
MSG_TRAJ_ENTRY_XFER	sim_cig_if.h
MSG_TRAJ_TABLE_XFER	if_init.h
MSG_TRAJ_TABLE_XFER	sim_cig_if.h
MSG_TRANSLATION	sim_cig_if.h
MSG_VIEW_FLAGS	sim_cig_if.h
MSG_VIEW_MAGNIFICATION	sim_cig_if.h
MSG_VIEW_MODE	sim_cig_if.h
MSG_VIEWPORT_STATE	if_init.h
MSG_VIEWPORT_STATE	sim_cig_if.h
MSGS_BLK	sim_cig_if.h
MTXUNION	if_header_struct.h
MTXUNION	sim_cig_if.h
MulticastGroupID	p_assoc.h
MunitionQuantity	basic.h
NetworkAddress	network.h
NetworkBuffer	network.h
NetworkHeader	network.h
NetworkHeader8023	network.h
NetworkPacket	network.h
ObjectType	basic.h
ObjectID	basic.h
ObjectType	p_sim.h
OrganizationalUnit	basic.h
OrganizationType	basic.h
OWN_COLLISIONS	if_targets.h
PARSE_DATA	parser.h (for libparser)
PARSE_TABLE	parser.h (for libparser)
PFI	sim_types.h
PointToPointPDU	p_p2p.h
PointToPointProtocolVersion	p_p2p.h
POT_TYPE	libpfile_dfn.h
PowerSubsystems	status.h
psndr	sersvr.h
qcontrol	ser.h
QueueElement	network.h
R4BBOX2D	dgi_stdg.h
R4BBOX3D	dgi_stdg.h
R4HSL	dgi_stdg.h
R4HSLO	dgi_stdg.h
R4MTX4X4	dgi_stdg.h
R4P2D	dgi_stdg.h
R4P3D	dgi_stdg.h
R4P4D	dgi_stdg.h

User_Defined_Type	Where_Typedef_Defined
R4RGB	dgi_stdg.h
R4RGBO	dgi_stdg.h
R8BBOX2D	dgi_stdg.h
R8BBOX3D	dgi_stdg.h
R8HSL	dgi_stdg.h
R8HSLO	dgi_stdg.h
R8MTX4X3	dgi_stdg.h
R8MTX4X4	dgi_stdg.h
R8P2D	dgi_stdg.h
R8P3D	dgi_stdg.h
R8P4D	dgi_stdg.h
R8RGB	dgi_stdg.h
R8RGBO	dgi_stdg.h
RadarDutyCycle	p_sim.h
RadarMode	p_sim.h
RadiateVariant	p_sim.h
REAL	sim_types.h
REAL_4	mass_stdc.h
REAL_8	mass_stdc.h
REMOTE_OBJECT_CLASS	libfilter.h
RemoteCommandVariant	p_mon.h
RepairCause	p_data.h
RepairRequestVariant	p_sim.h
RepairResponseVariant	p_sim.h
RepairResult	p_sim.h
RepairType	basic.h
reset_req	sersvr.h
RESOLUTION	if_header_struct.h
RESOLUTION	sim_cig_if.h
ResupplyCancelVariant	p_sim.h
ResupplyVariant	p_sim.h
ReturnSwitch	p_data.h
RingElement	network.h
ROT2x1_MTX	if_header_struct.h
ROT2x1_MTX	sim_cig_if.h
ROT_MATRIX	sim_types.h
RTS3x3_MTX	if_header_struct.h
RTS3x3_MTX	sim_cig_if.h
RTS4x3_MTX	if_header_struct.h
RTS4x3_MTX	sim_cig_if.h
RVA_ENTRY	librva.h
S_BUFFER	Section 2.1.2.2.2.117 Vehicles CSCI SDD
serdev	sersvr.h
sersvr_req	sersvr.h
settime_arg	enparg.h
ShellAckVariant	p_mon.h
ShellOpenVariant	p_mon.h
ShellStatus	p_mon.h
ShellStatusVariant	p_mon.h
ShellTextVariant	p_mon.h
ShowEffectVariant	p_data.h
SignalFrequency	p_sim.h

User Defined Type	Where Typedef Defined
SignalPower	p_sim.h
SIMNET_M1_Status	stat_m1.h
SIMNET_M2_Status	stat_m2.h
SIMNET_MCC_Status	p_data.h
SIMNET_Sim_Status	p_mgmt.h
simnetFAADStatus	status.h
simnetM1Status	status.h
simnetM2Status	status.h
SimulationAddress	address.h
SimulationPDU	p_sim.h
SimulationPDUKind	p_sim.h
SimulationProtocolVersion	p_sim.h
SimulationStatusVariant	p_data.h
SimulatorType	basic.h
SiteID	address.h
SkyColor	basic.h
SOUND_ELEMENT	sound_dfn.h
SP_ActivateReason	p_sim.h
SP_ActivateRequestVariant	p_sim.h
SP_ActivateResponseVariant	p_sim.h
SP_ActivateResult	p_sim.h
SP_AirframeSubsystems	status.h
SP_AirVehicleSubsystems	status.h
SP_AssociationDataUnit	p_assoc.h
SP_AssociationPDU	p_assoc.h
SP_AssociationPDUKind	p_assoc.h
SP_AssociationProtocolVersion	p_assoc.h
SP_AssociationUserProtocol	p_assoc.h
SP_AttachedVariant	p_stlth.h
SP_AttachVariant	p_stlth.h
SP_BreachedLaneStatusVariant	p_data.h
SP_BreachedLaneVariant	p_sim.h
SP_ChassisSubsystems	status.h
SP_CockpitSubsystems	status.h
SP_CollisionVariant	p_sim.h
SP_ComponentStatus	p_mgmt.h
SP_DamageCause	p_data.h
SP_DataCollectionPDU	p_data.h
SP_DataCollectionPDUKind	p_data.h
SP_DataCollectionProtocolVersion	p_data.h
SP_DeactivateReason	p_sim.h
SP_DeactivateRequestVariant	p_sim.h
SP_DeactivateResponseVariant	p_sim.h
SP_DeactivateResult	p_sim.h
SP_EffectType	p_data.h
SP_ElectronicSubsystems	status.h
SP_elevationRequestPDUKind	p_terra.h
SP_ElevationRequestVariant	p_terra.h
SP_elevationResponsePDUKind	p_terra.h
SP_ElevationResponseVariant	p_terra.h
SP_EquipStatusVariant	p_mgmt.h
SP_ErrorReportSeverity	p_mgmt.h

User Defined Type	Where Typedef Defined
SP_ErrorReportVariant	p_mgmt.h
SP_EventFlagVariant	p_data.h
SP_ExerciseStatusVariant	p_data.h
SP_FireResult	p_sim.h
SP_FireType	p_sim.h
SP_FireVariant	p_sim.h
SP_GenericVehicleStatus	status.h
SP_GroundVehicleSubsystems	status.h
SP_ImpactVariant	p_sim.h
SP_IndirectFireDetonation	p_sim.h
SP_IndirectFireVariant	p_sim.h
SP_LaserRangeFinder	p_data.h
SP_LaserRangeResult	p_data.h
SP_LaserRangeVariant	p_data.h
SP_LoggerActivity	p_logger.h
SP_LoggerAvailReplyVariant	p_logger.h
SP_LoggerClockTickVariant	p_logger.h
SP_LoggerCommandVariant	p_logger.h
SP_LoggerInformationVariant	p_logger.h
SP_LoggerMedium	p_logger.h
SP_LoggerOperation	p_logger.h
SP_LoggerPDU	p_logger.h
SP_LoggerPDUKind	p_logger.h
SP_LoggerProtocolVersion	p_logger.h
SP_LoggerRTCType	p_logger.h
SP_LoggerSeekRTC	p_logger.h
SP_LoggerSpeedRTC	p_logger.h
SP_LoggerStartRTC	p_logger.h
SP_LoggerState	p_logger.h
SP_LoggerStatusReplyVariant	p_logger.h
SP_LoggerTime	p_logger.h
SP_ManagementPDU	p_data.h
SP_ManagementPDUKind	p_mgmt.h
SP_ManagementProtocolVersion	p_mgmt.h
SP_MarkerDescriptor	p_sim.h
SP_MarkerVariant	p_sim.h
SP_MarkerVariantNumber	p_sim.h
SP_maxElevations	p_terra.h
SP_maxGenericVehicleStores	status.h
SP_MetamorphoseVariant	p_stlth.h
SP_MimicVariant	p_stlth.h
SP_MineDescriptor	p_data.h
SP_MinefieldVariant	p_data.h
SP_Monitor_PDU	p_mon.h
SP_Monitor_PDUKind	p_mon.h
SP_MonitorKind	p_mon.h
SP_MonitorResponseVariant	p_mon.h
SP_MotiveSubsystems	status.h
SP_MulticastGroupID	p_assoc.h
SP_PointToPointPDU	p_p2p.h
SP_PointToPointProtocolVersion	p_p2p.h
SP_PowerSubsystems	status.h

User Defined Type	Where Typedef Defined
SP_RadarDutyCycle	p_sim.h
SP_RadarMode	p_sim.h
SP_RadiateVariant	p_sim.h
SP_RemoteCommandVariant	p_mon.h
SP_RepairCause	p_data.h
SP_RepairRequestVariant	p_sim.h
SP_RepairResponseVariant	p_sim.h
SP_RepairResult	p_sim.h
SP_ResupplyCancelVariant	p_sim.h
SP_ResupplyVariant	p_sim.h
SP_ReturnSwitch	p_data.h
SP_ShellAckVariant	p_mon.h
SP_ShellOpenVariant	p_mon.h
SP_ShellStatus	p_mon.h
SP_ShellStatusVariant	p_mon.h
SP_ShellTextVariant	p_mon.h
SP_ShowEffectVariant	p_data.h
SP_SignalFrequency	p_sim.h
SP_SignalPower	p_sim.h
SP_SIMNET_M1_Status	stat_m1.h
SP_SIMNET_M2_Status	stat_m2.h
SP_SIMNET_MCC_Status	p_data.h
SP_SIMNET_Sim_Status	p_mgmt.h
SP_simnetFAADStatus	status.h
SP_simnetM1Status	status.h
SP_simnetM2Status	status.h
SP_SimulationAddress	address.h
SP_SimulationPDU	p_sim.h
SP_SimulationPDUKind	p_sim.h
SP_SimulationProtocolVersion	p_sim.h
SP_SimulationStatusVariant	p_data.h
SP_SiteID	address.h
SP_SpecificStatusCategory	status.h
SP_StatusChangeEffect	p-data.h
SP_StatusChangeVariant	p_data.h
SP_StatusQueryVariant	p_data.h
SP_StatusResponseVariant	p_data.h
SP_StatusResult	p_data.h
SP_StealthAppearanceVariant	p_stlth.h
SP_StealthErrorVariant	p-stlth.h
SP_StealthPDU	p_stlth.h
SP_StealthVisibilityVariant	p_stlth.h
SP_subsystemExists	status.h
SP_SubsystemsCategory	status.h
SP_subsystemStatus	status.h
SP_TeleportVariant	p_stlth.h
SP_TerrainPDU	p_terra.h
SP_TerrainPDUKind	p_terra.h
SP_TerrainProtocolVersion	p_terra.h
SP_terrainProtocolVersionCurrent	p_terra.h
SP_terrainProtocolVersionJan90	p_terra.h
SP_TransactionIdentifier	p_assoc.h

User_Defined_Type	Where_Typedef_Defined
SP_TurretSubsystems	status.h
SP_UnitRelation	p_data.h
SP_VehicleAppearanceVariant	p_sim.h
SP_vehiclesDynamics	p_stlth.h
SP_VehicleSpecificStatus	status.h
SP_VehicleStatus	status.h
SP_VehicleStatusVariant	p_data.h
SP_VehicleSubsystems	status.h
SP_WeaponSubsystems	status.h
SpecificStatusCategory	status.h
startenp_arg	enparg.h
STATUS_TYPE	libpfile_dfn.h
StatusChangeEffect	p_data.h
StatusChangeVariant	p_data.h
StatusQueryVariant	p_data.h
StatusResponseVariant	p_data.h
StatusResult	p_data.h
StealthAppearanceVariant	p_stlth.h
StealthErrorVariant	p_stlth.h
StealthPDU	p_stlth.h
StealthVisibilityVariant	p_stlth.h
subsystemExists	status.h
SubsystemsCategory	status.h
subsystemStatus	status.h
T_MAT_PTR	sim_types.h
T_MATRIX	sim_types.h
TableControl	table.h
TableEnt	table.h
TARGET_SUPPORTS	if_targets.h
TargetDescriptor	basic.h
TargetType	basic.h
TDB_INFO	tdb.h
TDB_POINT	tdb.h
TDB_STATUS	tdb.h
TDB_THING	tdb.h
TeleportVariant	p_stlth.h
templet_arg	enparg.h
termio	termio.h
TerrainDatabaseID	basic.h
TerrainMap	map.h
TerrainPDU	p_terra.h
TerrainPDUKind	p_terra.h
TerrainProtocolVersion	p_terra.h
terrainProtocolVersionCurrent	p_terra.h
terrainProtocolVersionJan90	p_terra.h
TF1	if_header_struct.h
TF1	sim_cig_if.h
TF2	if_header_struct.h
TF2	sim_cig_if.h
TF_TYPE	if_header_struct.h
Time	basic.h
TIMER	timers_dfn.h

User_Defined_Type	Where_Typedef_Defined
TransactionIdentifier	p_assoc.h
TurretSubsystems	status.h
type_arg	enparg.h
UnitIdentifier	basic.h
UnitRelation	p_data.h
UnitType	basic.h
VECTOR	sim_types.h
VEH_LIMITS	libfilter.h
VehicleAppearance	libapp.h
VehicleAppearanceVariant	p_sim.h
Vehicleaspsbilities	basic.h
VehicleCapabilities	basic.h
VehicleClass	basic.h
VehicleComponent	basic.h
VehicleCoordinates	basic.h
VehicleGuises	basic.h
VehicleID	basic.h
VehicleIDStatus	veh_table.h
VehicleMarking	basic.h
vehiclesDynamics	p_stlth.h
VehicleSpecificStatus	status.h
VehicleStatus	status.h
VehicleStatusVariant	p_data.h
VehicleSubsystems	status.h
VelocityVector	basic.h
version_arg	enparg.h
VNS_1	mass_stdc.h
VNS_2	mass_stdc.h
VNS_4	mass_stdc.h
WeaponSubsystems	status.h
WORD	mass_stdc.h
WorldCoordinates	basic.h
write_req	sersvr.h
XYCoordinates	basic.h
XYZHPR	if_header_struct.h

APPENDIX E

The following is a list of globally accessable macros and the files in which they are defined.

Macro	Filename
abs	abs.h
abs	sim_macros.h
ASSOC_ADDRESS_EQUAL	assoc.h
between	sim_macros.h
bound	sim_macros.h
BSWAP	libcif.h
BYTE_COUNT	gbuffer.h
CALL	parser.h (for libparser)
CALL1	parser.h (for libparser)
circle_to_mil	trig.h
cntrl	sim_macros.h
CONSTANT	parser.h (for libparser)
CONSTANT_TABLE	parser.h (for libparser)
cube	powers.h
çube	sim_macros.h
DEFINE_TABLE	parser.h (for libparser)
deg_to_mil	sim_macros.h
deg_to_mil	sim_macros.h
deg_to_rad	sim_macros.h
deg_to_rad	sim_macros.h
deg_to_rad	trig.h
DO_KEYWORD_TABLE	parser.h (for libparser)
DOT	gbuffer.h
dtad_device	dtad.h
Dtad_error	dtad.h
Dtad_pull	dtad.h
Dtad_read	dtad.h
Dtad_start	dtad.h
END_FRAGMENT	bbd.h
END_FRAGMENT_MAX	bbd.h
END_OF_DMA_BUFFER	gbuffer.h
eq	sim_macros.h
eq(x,y,z)	tolerance.h
F_REFLECT	compat.h
FIELD	parser.h (for libparser)
FIELD_TABLE	parser.h (for libparser)
FIFO_EMPTY	fifo_dfn.h
FIFO_FULL	fifo_dfn.h
FOREVER	sim_macros.h
frame_delay_of_sound	sim_macros.h
GET_CONTROL(p)	network.h
GET_DATA_PTR(p)	network.h
GET_DSAP(p)	network.h
GET_ETHER_TYPE(p)	network.h
GET_PROTOID(p)	network.h
GET_SSAP(p)	network.h
GETCONSTANT	parser.h (for libparser)

Macro	Filename
GETDECIMAL	parser.h (for libparser)
GETFIELDS	parser.h (for libparser)
GETOCTAL	parser.h (for libparser)
GETSTRING	parser.h (for libparser)
GETWORD	ser.h
HELP_PRINT1	sim_macros.h
HELP_PRINT2	sim_macros.h
HEX2	sim_macros.h
HEX3	sim_macros.h
HIWORD	libcif.h
ischar	sim_style.h
iseven	sim_style.h
ishex	sim_style.h
isnum	sim_style.h
isodd	sim_style.h
isprntbl	sim_style.h
iswhite	sim_style.h
KEYWORD	parser.h (for libparser)
KEYWORD_SELECT	parser.h (for libparser)
L_REFLECT	compat.h
L_SWAP	compat.h
LGETHEX	parser.h (for libparser)
LOWORD	libcif.h
mag2	sim_macros.h
mag3	sim_macros.h
max	sim_macros.h
MGMT_HEADER_SIZE	p_size.h
mil_to_circle	trig.h
mil_to_deg	sim_macros.h
mil_to_deg	sim_macros.h
mil_to_rad	sim_macros.h
mil_to_rad	sim_macros.h
mil_to_rad	trig.h
min	sim_macros.h
N-GETPTR	itab.h
N_BUFPTR	itab.h
N_BUFSIZ	itab.h
N_ISLOCKED	itab.h
N_ISNEW	itab.h
N_LOCK	itab.h
N_NUMENTRIES	itab.h
N_SETNEW	itab.h
N_SETOLD	itab.h
N_SETPTR	itab.h
N_UNLOCK	itab.h
NEXT_IN	fifo_dfn.h
NEXT_OUT	fifo_dfn.h
p_arg0	parser.h (for libparser)
PACKET_LENGTH	gbuffer.h
PAGE_ALLIGN	enparg.h
PARSE_FUNCTION	parser.h (for libparser)
PARSE_KEYWORD	parser.h (for libparser)

Macro	Filename
parser_init	sec 2.5.2.1.1 in Sim Host CSCI SDD
parser_init	sec 2.5.2.1.1 in Sim Host CSCI SDD
parser_restore_term	sec 2.5.2.1.2 in Sim Host CSCI SDD
parser_restore_term	sec 2.5.2.1.2 in Sim Host CSCI SDD
PATCH_INDEX	tdb.h
PORE	parser.h (for libparser)
PORE_B	parser.h (for libparser)
PORE_L	parser.h (for libparser)
PORE_W	parser.h (for libparser)
potval	sim_macros.h
pr	serioctl.h
PRINT	parser.h (for libparser)
PRINT1	parser.h (for libparser)
PRINT_VID_ERROR	rva_lcc.h Vehicles CSCI SDD
PRINTFIELDS	parser.h (for libparser)
PRO_ASSOC_DATAGRAM_SIZE	p_size.h
PRO_ASSOC_PADDING_HDR_SIZE	p_size.h
PRO_ASSOC_PADDING_SIZE	p_size.h
PRO_ASSOC_REQUEST_HDR_SIZE	p_size.h
PRO_ASSOC_REQUEST_SIZE	p_size.h
PRO_ASSOC_RESPONSE_SIZE	p_size.h
PRO_DATA_EXERCISE_STATUS_SIZE	p_size.h
PRO_DATA_SIMULATION_STATUS_SIZE	p_size.h
PRO_DATA_STATUS_CHANGE_SIZE	p_size.h
PRO_DATA_VEHICLE_STATUS_SIZE	p_size.h
PRO_SIM_ACTIVATE_REQUEST	p_size.h
PRO_SIM_APPEARANCE_SIZE	p_size.h
PRO_SIM_DEACTIVATE_REQUEST	p_size.h
PRO_SIM_FIRE_SIZE	p_size.h
PRO_SIM_IND_FIRE_SIZE	p_size.h
PRO_SIM_MARKER_SIZE	p_size.h
PRO_SIM_REPAIR_REQUEST_SIZE	p_size.h
PRO_SIM_RESUPPLY_CANCEL_SIZE	p_size.h
PRO_SIM_RESUPPLY_OFFER_SIZE	p_size.h
PUTARG	parser.h (for libparser)
qnum_to_chnum	ser.h
rad_to_deg	sim_macros.h
rad_to_deg	sim_macros.h
rad_to_deg	trig.h
rad_to_mil	sim_macros.h
rad_to_mil	sim_macros.h
rad_to_mil	trig.h
radians_to_simnet_angle	libapp.h
readq	ser.h
REPORT_ERROR	gbuffer.h
Ringsize_in_chars(size)	network.h
roll_dice	sim_macros.h
SET_CONTROL(p,x)	network.h
SET_DSAP(p,x)	network.h
SET_ETHER_TYPE(p,x)	network.h
SET_PROTOID(p,x)	network.h
SET_SSAP(p,x)	network.h

Macro	Filename
shift_args	cigutil.h
sign	sim_macros.h
SIM_ADDRESS_EQUAL	sim_macros.h
SIM_ADDRESSES_EQUAL	sim_macros.h
simnet_angle_to_radians	libapp.h
SINES_GET_COS	sines.h
SINES_GET_SIN	sines.h
SINES_SHIFT_INDEX	sines.h
square	powers.h
square	sim_macros.h
START_FRAGMENT	bbd.h
step	sim_macros.h
step(x,y,z)	tolerance.h
vec_mag2	sim_macros.h
vec_mag3	sim_macros.h
VEHICLE_IDS_EQUAL	sim_macros.h
VEHICLE_IDS_EQUAL	sim_macros.h
W_SWAP	compat.h
writeq	ser.h
xor	sim_macros.h

INDEX BY SECTION NUMBER

abort_handler CSU	2.1.3.1.20
abs.h CSU	2.13.3.2
acquire_target CSU	2.6.9.5.1
action_drill_needed CSU	2.10.1.3.6
activate_new_cis CSU	2.10.1.1.16
add_executing_unit CSU	2.10.2.1.21
add_into_spotter_table CSU	2.8.2.3.9
add_last_hit CSU	2.6.7.1.24
add_removed_vehicle CSU	2.8.2.1.29
add_shelling_to_cluster CSU	2.8.2.1.37
add_to_free_list CSU	2.14.2.1.9
add_vehicle_to_cluster CSU	2.8.2.1.20
add_weapons_status CSU	2.6.9.8.12
adjust_behaviors CSU	2.10.1.1.15
agl_to_abs_altitude CSU	2.6.4.2.9
Air Maneuver CSC	2.6.4
airveh_catastrophic_kill CSU	2.6.4.6.7
airveh_init CSU	2.6.4.6.1
airveh_mobility_kill CSU	2.6.4.6.6
airveh_show CSU	2.6.4.6.4
air_raid_happening CSU	2.10.1.3.4
air_raid_over CSU	2.10.1.3.5
air_tick CSU	2.6.4.6.5
align_points CSU	2.12.1.12.3
Alloc CSU	2.5.1.9.1
alloc.c CSC	2.14.2.1
allocate_bitfield CSU	2.14.3.4.1
allocate_points CSU	2.12.1.1.5
allocate_safobj CSU	2.2.2.2
allocate_spotter_tables CSU	2.8.2.3.3
all_wide_segments_thru_water CSU	2.12.1.4.8
ammo_to_shell_type CSU	2.8.2.1.34
and_bitfield CSU	2.14.3.4.9
angle_between_vectors CSU	2.14.3.5.7
angle_clip CSU	2.14.3.5.4
any_wide_segment_thru_water CSU	2.12.1.4.3
append_routepoint CSU	2.10.2.4.2
append_vector_2d_on_route CSU	2.12.1.11.19
ApplicationGetSiteHost CSU	2.3.2.4
apply_to_blocks_of_size CSU	2.14.2.1.19
arc_of_attention CSU	2.6.5.1.9

are_we_there CSU	2.6.4.2.18
args.c CSC	2.14.3.1
arty_type_string CSU	2.4.3.2.36
AssocGetSimAddress CSU	2.3.2.3
attackatwill_tick CSU	2.6.4.2.61
attackatwill_tick_new CSU	2.6.4.2.62
attenuation_factor CSU	2.6.7.1.48
average CSU	2.6.7.1.20
avoid_water_to_next_point CSU	2.10.2.4.11
Back1Char CSU	2.5.1.4.12
BackChar CSU	2.5.1.4.25
BeginningOfLine CSU	2.5.1.4.11
BEGIN_ARG_TABLE(SWITCH) CSU	2.1.3.1.18
binarysearch_tag CSU	2.1.1.4.6
bitfield.c CSC	2.14.3.4
blaster.c CSC	2.15.1
break_block CSU	2.14.2.1.7
bridges.c CSC	2.12.1.3
broadcast_appearance_data CSU	2.4.3.2.73
broadcast_echelon_data CSU	2.4.3.2.72
broadcast_pae_data CSU	2.4.3.2.74
broadcast_veh_is_gone CSU	2.4.3.2.75
buffer_allocate CSU	2.14.4.2.12
buffer_allocate_from_pool CSU	2.14.4.1.10
buffer_allocate_into_pool CSU	2.14.4.1.12
buffer_deallocate CSU	2.14.4.2.15
buffer_dequeue CSU	2.14.4.2.20
buffer_describe CSU	2.14.4.2.24
buffer_distribute CSU	2.14.4.2.26
buffer_enqueue CSU	2.14.4.2.18
buffer_excise CSU	2.14.4.2.19
buffer_flush CSU	2.14.4.2.23
buffer_length CSU	2.14.4.2.16
buffer_one_less CSU	2.14.4.2.29
buffer_one_more CSU	2.14.4.2.28
buffer_pool_allocate CSU	2.14.4.1.13
buffer_pool_return_buffer CSU	2.14.4.1.11
buffer_set_size CSU	2.14.4.2.13
buffer_simple_dequeue CSU	2.14.4.1.15
buffer_simple_enqueue CSU	2.14.4.1.14
buffer_simple_flush CSU	2.14.4.1.16
buffer_statistics_init CSU	2.14.4.2.27
buffer_statistics_print CSU	2.14.4.2.30

buffer_traverse_and_apply CSU 2.14.4.2.21
buffer_traverse_and_apply_n_times CSU 2.14.4.2.22
buffer_up_refcnt CSU 2.14.4.2.17
buffer__deallocate CSU 2.14.4.2.14
buff_rudp_read_message CSU 2.4.2.5.11
bufpool.c CSC 2.14.4.1
buf_conn.c CSC 2.4.2.5
buf_conn.h CSU 2.4.2.6
buf_master_rudp_close CSU 2.4.2.5.7
buf_master_rudp_open CSU 2.4.2.5.4
buf_master_rudp_synch CSU 2.4.2.5.5
buf_rudp_close CSU 2.4.2.5.6
buf_rudp_discard_all_buffers CSU 2.4.2.5.8
buf_rudp_disconnect CSU 2.4.2.5.9
buf_rudp_flush CSU 2.4.2.5.10
buf_rudp_init CSU 2.4.2.5.1
buf_rudp_open CSU 2.4.2.5.3
buf_rudp_show CSU 2.4.2.5.2
buf_rudp_tick CSU 2.4.2.5.13
buf_rudp_write_message CSU 2.4.2.5.12
buildings_thru CSU 2.12.1.17.1
building_check.c CSC 2.12.1.17
build_packets CSU 2.15.1.2
cancel_fncl CSU 2.2.1.1.4
cancel_fncl_group CSU 2.2.1.1.5
cancel_overlay CSU 2.14.1.1.10
canopies CSU 2.6.7.1.18
center_column CSU 2.12.1.13.7
ChangePrompt CSU 2.5.1.4.1
change_fncl_period CSU 2.2.1.1.6
change_movement_and_report CSU 2.10.1.1.19
change_tick_rate CSU 2.2.3.5
checksum CSU 2.14.3.2.1
checksum.c CSC 2.14.3.2
check_box CSU 2.6.7.1.31
check_edges CSU 2.6.7.1.8
check_edge_hit CSU 2.6.7.1.11
check_hits CSU 2.6.7.1.21
check_object CSU 2.6.7.1.14
check_predicates CSU 2.10.1.3.13
check_prob CSU 2.14.3.7.2
check_tree_hit CSU 2.6.7.1.29
choose_resupply_item CSU 2.6.6.1.7

choose_skirt_point CSU 2.6.3.1.16
choose_target_and_weapon CSU 2.6.9.3.20
cis.c CSC 2.10.1.1
cis.h CSU 2.10.1.2
cistrcmp CSU 2.14.3.8.2
cistrncmp CSU 2.14.3.8.3
cis_show CSU 2.10.1.1.10
ClearLine CSU 2.5.1.4.19
clear_bit CSU 2.14.3.4.4
clear_bitfield CSU 2.14.3.4.6
clear_cis_stack CSU 2.10.1.1.14
clear_detection CSU 2.6.5.1.2
clear_weapons_status CSU 2.6.9.8.11
clip_angle_negative CSU 2.14.3.5.6
clip_angle_positive CSU 2.14.3.5.5
clip_to_tdb CSU 2.6.7.1.3
closest_leader CSU 2.8.1.4.7
closest_to_a_leader CSU 2.8.1.4.6
cluster_from_vehicle CSU 2.8.2.1.25
cluster_shell CSU 2.8.2.1.39
cluster_vehicle CSU 2.8.2.1.24
cm.c CSC 2.10.2.1
cm.h CSU 2.10.2.2
cmp_form_entries CSU 2.8.1.4.11
cmp_tags CSU 2.1.1.4.4
cm_add_area CSU 2.10.2.1.9
cm_add_line CSU 2.10.2.1.11
cm_add_point CSU 2.10.2.1.8
cm_add_route CSU 2.10.2.1.12
cm_add_zone CSU 2.10.2.1.10
cm_applies_to_units CSU 2.10.2.1.16
cm_count_intersections CSU 2.10.2.3.17
cm_delete_cm CSU 2.10.2.1.14
cm_delete_cm_from_overlay CSU 2.10.2.1.15
cm_delete_overlay CSU 2.10.2.1.13
cm_execute_overlay CSU 2.10.2.1.20
cm_find_cm CSU 2.10.2.1.7
cm_find_cm_list_item CSU 2.10.2.1.6
cm_find_overlay CSU 2.10.2.1.5
cm_find_overlay_list_item CSU 2.10.2.1.4
cm_force_overlay_recalculation CSU 2.10.2.1.23
cm_get_center_of_mass CSU 2.10.2.3.15
cm_get_cm_from_overlay CSU 2.10.2.1.17

cm_intersection CSU 2.10.2.3.13
cm_point_inside_polygon CSU 2.10.2.3.16
cm_point_line_intersection CSU 2.10.2.3.18
cm_poly_intersection CSU 2.10.2.3.14
cm_recompute_cm_from_overlay CSU 2.10.2.1.19
cm_to_passage_string CSU 2.10.1.1.17
collect CSU 2.14.2.1.6
collect_perf_stat CSU 2.2.1.4.5
collision.c CSC 2.6.3.1
collision_dead_tick CSU 2.6.3.1.7
collision_disengaging CSU 2.6.3.1.5
collision_remove_vehicles CSU 2.6.3.1.3
collision_show CSU 2.6.3.1.4
collision_tick CSU 2.6.3.1.6
collision_vehicle_rammed CSU 2.6.3.1.12
Combat Instruction Sets CSC 2.10.1
combined_velocity CSU 2.6.4.2.14
Command and Control CSC 2.8.1
commander.c CSC 2.8.1.1
commander.h CSU 2.8.1.2
commander_attack CSU 2.8.1.1.27
commander_cancel_overlay CSU 2.8.1.1.8
commander_change_altitude CSU 2.8.1.1.26
commander_change_formation CSU 2.8.1.1.16
commander_change_speed CSU 2.8.1.1.14
commander_execute_overlay CSU 2.8.1.1.7
commander_executing_order CSU 2.8.1.1.6
commander_face_direction CSU 2.8.1.1.21
commander_follow_vehicle CSU 2.8.1.1.17
commander_get_mission_status CSU 2.8.1.1.5
commander_goto_point CSU 2.8.1.1.19
commander_halt CSU 2.8.1.1.11
commander_inferior_changed_status CSU 2.8.1.1.22
commander_land CSU 2.8.1.1.25
commander_note_leader_state CSU 2.8.1.1.23
commander_restore_leader_state CSU 2.8.1.1.24
commander_resume_mission CSU 2.8.1.1.20
commander_set_mission_direction CSU 2.8.1.1.13
commander_set_mission_formation CSU 2.8.1.1.15
commander_set_mission_speed CSU 2.8.1.1.12
commander_show CSU 2.8.1.1.4
commander_simulator_in_command CSU 2.8.1.1.18
commander_state_string CSU 2.8.1.1.3

commander_tick CSU 2.8.1.1.9
CommandLog CSU 2.5.1.8.1
compare_targets CSU 2.6.9.3.14
composite.c CSC 2.8.1.3
composite_add_inferior_composite CSU 2.8.1.3.5
composite_add_inferior_vehicle CSU 2.8.1.3.7
composite_add_member_vehicle CSU 2.8.1.3.9
composite_assume_formation CSU 2.8.1.3.26
composite_assume_formation_internal CSU 2.8.1.3.25
composite_attack CSU 2.8.1.3.51
composite_cancel_overlay CSU 2.8.1.3.42
composite_change_altitude CSU 2.8.1.3.53
composite_change_formation CSU 2.8.1.3.47
composite_change_speed CSU 2.8.1.3.44
composite_check_unit_strength CSU 2.8.1.3.17
composite_distinguished_member CSU 2.8.1.3.16
composite_execute_overlay CSU 2.8.1.3.40
composite_executing_order CSU 2.8.1.3.35
composite_face_direction CSU 2.8.1.3.50
composite_fake_resupply CSU 2.8.1.3.32
composite_fill_in_appearance_data CSU 2.8.1.3.14
composite_fill_in_echelon_data CSU 2.8.1.3.12
composite_fill_in_position_data CSU 2.8.1.3.13
composite_find_living_vehicle CSU 2.8.1.3.15
composite_find_vehicle_for_stealth CSU 2.8.1.3.29
composite_follow_vehicle CSU 2.8.1.3.45
composite_generate_status_report CSU 2.8.1.3.33
composite_get_composite_bumpers CSU 2.8.1.3.22
composite_goto_point CSU 2.8.1.3.48
composite_go_away CSU 2.8.1.3.2
composite_halt CSU 2.8.1.3.43
composite_indirect_fire CSU 2.8.1.3.21
composite_inferior_changed_status CSU 2.8.1.3.37
composite_land CSU 2.8.1.3.52
composite_note_leader_state CSU 2.8.1.3.36
composite_note_member_vehicle_has_died CSU 2.8.1.3.38
composite_pro_sim CSU 2.8.1.3.20
composite_readjust_overlay CSU 2.8.1.3.41
composite_reassign_current_formation CSU 2.8.1.3.27
composite_rejoin_unit CSU 2.8.1.3.39
composite_remove_inferior_composite CSU 2.8.1.3.6
composite_remove_inferior_vehicle CSU 2.8.1.3.8
composite_remove_member_vehicle CSU 2.8.1.3.10

composite_remove_vehicles CSU 2.8.1.3.34
composite_resume_mission CSU 2.8.1.3.49
composite_send_unit_strength_message CSU 2.8.1.3.18
composite_set_recursive_ids CSU 2.8.1.3.23
composite_set_sbx_ids CSU 2.8.1.3.24
composite_set_superior CSU 2.8.1.3.4
composite_set_targeting_parameters CSU 2.8.1.3.31
composite_show CSU 2.8.1.3.3
composite_simulator_in_command CSU 2.8.1.3.46
composite_start_ticking CSU 2.8.1.3.11
composite_teleport_to_station CSU 2.8.1.3.28
composite_tick CSU 2.8.1.3.19
compute_damage_keys CSU 2.6.8.6.11
compute_enemy_weight CSU 2.6.5.1.7
compute_explosion_point CSU 2.6.9.1.9
compute_interest_direction CSU 2.6.5.1.8
compute_mid CSU 2.6.7.1.10
compute_situation CSU 2.6.4.2.20
compute_who_follows_whom CSU 2.8.1.4.4
CONFIGURATION AND CONFIGURATION MANAGEMENT 1.4
cons CSU 2.12.1.7.1
Control Measure CSC 2.10.2
CopyEarlierCommand CSU 2.5.1.8.2
copy_bitfield CSU 2.14.3.4.5
copy_float_point_list_to_real_point_list CSU 2.10.2.1.1
copy_list CSU 2.12.1.7.14
copy_matrix_row_to_vector CSU 2.14.3.5.18
copy_stack_to_array CSU 2.1.1.1.5
copy_xy_on_tdb CSU 2.14.1.2.5
countries_from_battle_scheme_and_force CSU 2.5.2.1.34
country_to_string CSU 2.4.3.2.28
count_bits CSU 2.14.3.4.10
count_forces CSU 2.5.2.1.13
count_hosts CSU 2.5.2.1.15
count_intersections CSU 2.12.1.14.4
count_mid_hit CSU 2.6.7.1.34
count_sites CSU 2.5.2.1.14
count_vehicles CSU 2.5.2.1.12
count_vtx1_hit CSU 2.6.7.1.32
count_vtx2_hit CSU 2.6.7.1.33
CREATE CSC 2.11
create.c CSC 2.11.1
create.c CSC 2.14.2.2

create_airveh CSU 2.6.4.6.2
create_bridges CSU 2.12.1.3.1
create_buildings CSU 2.12.1.16.5
create_cis CSU 2.10.1.1.8
create_collision CSU 2.6.3.1.1
create_commander CSU 2.8.1.1.1
create_composite CSU 2.8.1.3.1
create_contours CSU 2.12.1.16.7
create_damage CSU 2.6.2.1.1
create_detection CSU 2.6.5.1.1
create_driver CSU 2.6.3.2.1
create_entity CSU 2.9.2.1.1
create_groundveh CSU 2.6.3.4.1
create_lakes CSU 2.12.1.3.3
create_logistics CSU 2.6.6.1.1
create_missile CSU 2.6.9.1.3
create_navigator CSU 2.10.2.3.1
create_new_cluster CSU 2.8.2.1.19
create_new_shelling CSU 2.8.2.1.36
create_pilot CSU 2.6.4.2.2
create_predicates CSU 2.10.1.3.1
create_remote CSU 2.7.1.1
create_remote_vehicle CSU 2.7.1.17
create_reporter CSU 2.8.2.1.1
create_rivers CSU 2.12.1.2.1
create_river_ints CSU 2.12.1.2.3
create_roads CSU 2.12.1.15.1
create_road_ints CSU 2.12.1.15.3
create_routepoint CSU 2.10.2.4.3
create_saf_vehicle CSU 2.6.1.1.1
create_spotter CSU 2.8.2.3.1
create_targeting CSU 2.6.9.3.2
create_tickable CSU 2.2.3.1
create_trees CSU 2.12.1.16.1
create_tree_canopies CSU 2.12.1.16.3
create_turret CSU 2.6.9.6.1
create_unit CSU 2.11.1.2
create_vehicle CSU 2.9.2.2.1
create_weapon_systems CSU 2.6.9.8.2
crossing_location CSU 2.12.1.11.9
CSC Descriptions 2
cstring.c CSC 2.14.3.8
Damage CSC 2.6.2

damage.c CSC	2.6.2.1
damage_indirect_fire CSU	2.6.2.1.6
damage_string CSU	2.6.2.1.3
damage_vehicle_impact CSU	2.6.2.1.4
damage_vehicle_rammed CSU	2.6.2.1.5
database.c CSC	2.6.8.8
database_detection_query CSU	2.6.8.2.2
database_df_damage_query CSU	2.6.8.6.10
database_echelon_lobj_query CSU	2.6.8.5.3
database_echelon_response_formation_query CSU	2.6.8.5.4
database_echelon_template_query CSU	2.6.8.5.2
database_formation_form_query CSU	2.6.8.4.2
database_formation_sub_form_query CSU	2.6.8.4.3
database_if_damage_query CSU	2.6.8.7.6
database_init CSU	2.1.3.1.6
database_read CSU	2.6.8.8.1
database_read_echelon CSU	2.6.8.5.1
database_read_formation CSU	2.6.8.4.1
deallocate_bitfield CSU	2.14.3.4.2
deallocate_safobj CSU	2.2.2.3
deallocate_spotter_tables CSU	2.8.2.3.4
debug.h CSU	2.5.2.2
debugging_on CSU	2.1.3.1.7
debug_report CSU	2.6.7.1.41
deferred_fncl CSU	2.2.1.1.1
delete CSU	2.12.1.7.7
Delete1Backward CSU	2.5.1.4.17
Delete1Forward CSU	2.5.1.4.13
DeleteChar CSU	2.5.1.4.26
DeleteWord CSU	2.5.1.4.22
delete_last_item CSU	2.12.1.7.12
describe_symbol_table CSU	2.1.1.3.4
destroy_airveh CSU	2.6.4.6.3
destroy_area_cm CSU	2.10.2.1.26
destroy_cis CSU	2.10.1.1.9
destroy_cm CSU	2.10.2.1.29
destroy_cm_data CSU	2.10.2.1.24
destroy_cm_list CSU	2.10.2.1.31
destroy_cm_list_item CSU	2.10.2.1.30
destroy_collision CSU	2.6.3.1.2
destroy_commander CSU	2.8.1.1.2
destroy_damage CSU	2.6.2.1.2
destroy_detection CSU	2.6.5.1.3

destroy_driver CSU 2.6.3.2.2
destroy_entity CSU 2.9.2.1.2
destroy_grid_entry_list CSU 2.9.3.1.13
destroy_groundveh CSU 2.6.3.4.2
destroy_line_cm CSU 2.10.2.1.28
destroy_logistics CSU 2.6.6.1.2
destroy_missile CSU 2.6.9.1.4
destroy_navigator CSU 2.10.2.3.2
destroy_overlay CSU 2.10.2.1.32
destroy_overlay_list_item CSU 2.10.2.1.33
destroy_pilot CSU 2.6.4.2.4
destroy_point_cm CSU 2.10.2.1.25
destroy_predicates CSU 2.10.1.3.2
destroy_removed_vehicles CSU 2.8.2.1.14
destroy_rem_veh_id_item CSU 2.8.2.1.15
destroy_reporter CSU 2.8.2.1.2
destroy_route CSU 2.10.2.4.4
destroy_routepoint CSU 2.10.2.4.5
destroy_shellings CSU 2.8.2.1.16
destroy_shell_summary CSU 2.8.2.1.17
destroy_spotter CSU 2.8.2.3.2
destroy_targeting CSU 2.6.9.3.3
destroy_tickable CSU 2.2.3.2
destroy_turret CSU 2.6.9.6.2
destroy_vehicle CSU 2.9.2.2.2
destroy_vehicle_cluster CSU 2.8.2.1.11
destroy_vehicle_clusters CSU 2.8.2.1.10
destroy_veh_id_item CSU 2.8.2.1.13
destroy_veh_id_list CSU 2.8.2.1.12
destroy_weapon_systems CSU 2.6.9.8.3
destroy_zone_cm CSU 2.10.2.1.27
detectable CSU 2.6.5.1.10
detection.c CSC 2.6.5.1
detection.c CSC 2.6.8.2
detection_remove_vehicles CSU 2.6.5.1.4
detection_save_weight CSU 2.6.5.1.12
detection_show CSU 2.6.5.1.5
detection_show_type CSU 2.6.5.1.6
detection_tick CSU 2.6.5.1.11
detect_building_on_path_tick CSU 2.6.3.1.10
detect_collision_tick CSU 2.6.3.1.13
detect_imminent_collision_tick CSU 2.6.3.1.8
df_damage.c CSC 2.6.8.6

diffraction_diff CSU 2.6.7.1.30
diffv CSU 2.14.1.2.9
dir_to_tan_point CSU 2.6.4.2.10
dispatch.c CSC 2.14.1.1
distance CSU 2.12.1.11.1
distance_around_path CSU 2.12.1.11.16
DoCommand CSU 2.5.1.7.1
DoEscape CSU 2.5.1.7.3
DoHelp CSU 2.5.1.7.2
dotv CSU 2.14.1.2.8
Do_Hash CSU 2.1.1.3.5
do_heap_collect CSU 2.14.2.1.18
do_heap_statistics CSU 2.14.2.1.17
do_heap_verify CSU 2.14.2.1.16
do_send CSU 2.15.1.3
driver.c CSC 2.6.3.2
driver.h CSU 2.6.3.3
driver_change_speed_cmd CSU 2.6.3.2.33
driver_execute_overlay CSU 2.6.3.2.17
driver_executing_immediate_command CSU 2.6.3.2.15
driver_face_direction_cmd CSU 2.6.3.2.30
driver_follow_leader CSU 2.6.3.2.19
driver_follow_vehicle_cmd CSU 2.6.3.2.28
driver_forget_about_forps CSU 2.6.3.2.21
driver_goto_point_cmd CSU 2.6.3.2.29
driver_halt_cmd CSU 2.6.3.2.27
driver_mission_completed CSU 2.6.3.2.16
driver_remove_vehicles CSU 2.6.3.2.3
driver_resume_cmd CSU 2.6.3.2.32
driver_resume_from_collision CSU 2.6.3.2.26
driver_resume_mission_cmd CSU 2.6.3.2.31
driver_set_direction CSU 2.6.3.2.24
driver_set_direction_cmd CSU 2.6.3.2.34
driver_set_leader_mis CSU 2.6.3.2.18
driver_set_route CSU 2.6.3.2.22
driver_set_routedirection CSU 2.6.3.2.25
driver_set_route_direction_cmd CSU 2.6.3.2.35
driver_set_speed CSU 2.6.3.2.23
driver_show CSU 2.6.3.2.43
driver_simulator_in_command CSU 2.6.3.2.12
driver_stop_mission CSU 2.6.3.2.20
driver_tick CSU 2.6.3.2.4
echelon.c CSC 2.6.8.5

edge_glw_miss CSU 2.6.7.1.37
edge_z_miss CSU 2.6.7.1.38
eliminate_duplicates CSU 2.10.2.4.9
eliminate_vehicles 2.2.4.4
EmptyLine CSU 2.5.1.4.10
enable_predicate CSU 2.10.1.3.11
enable_predicates CSU 2.10.1.3.12
EndOfLine CSU 2.5.1.4.14
enemy_disposition CSU 2.10.1.3.8
enqueue_on_rcvq CSU 2.3.2.10
entity.c CSC 2.9.2.1
entity_fill_in_appearance_data CSU 2.9.2.1.6
entity_fill_in_position_data CSU 2.9.2.1.5
entity_show CSU 2.9.2.1.4
exagerate_bridges CSU 2.10.2.4.10
execute_overlay CSU 2.14.1.1.8
exit_all_sbx_conns CSU 2.4.3.2.2
exit_handler CSU 2.1.3.1.21
expand_points CSU 2.12.1.17.5
expand_road_route CSU 2.10.2.4.6
extend_bridge CSU 2.12.1.10.3
extend_crossing CSU 2.12.1.10.1
extend_crossings.c CSC 2.12.1.10
extend_intersection CSU 2.12.1.10.2
extend_segment CSU 2.12.1.10.4
External Functions CSC 2.13.2
EXTERNAL INTERFACES 1.2
fake_resupply CSU 2.14.1.1.12
FieldGet CSU 2.5.1.5.3
fill_generic_cis_report CSU 2.10.1.1.18
fill_in_appearance_data CSU 2.14.1.1.1
fill_in_echelon_data CSU 2.14.1.1.2
fill_in_position_data CSU 2.14.1.1.3
fill_sbx_opfor_header CSU 2.4.3.2.18
final_relax_points CSU 2.12.1.11.12
find_a_quad CSU 2.12.1.5.1
find_closer_crossing CSU 2.12.1.11.7
find_closest_building CSU 2.12.1.17.3
find_direction_at_crossing CSU 2.12.1.11.3
find_extra_contour_info CSU 2.12.1.16.8
find_first_vector CSU 2.12.1.11.5
find_next_point CSU 2.12.1.11.8
find_river_bend_points CSU 2.12.1.12.10

find_river_points CSU 2.12.1.12.2
find_route CSU 2.12.1.8.1
find_route_core CSU 2.12.1.8.2
find_segment_cross_points CSU 2.12.1.10.5
find_suitable_crossing_route CSU 2.12.1.8.4
find_tag CSU 2.1.1.4.3
find_tag_sorted CSU 2.1.1.4.7
firestatus_to_string CSU 2.6.9.3.4
fire_missile_at_target CSU 2.6.9.1.6
fire_weapon_at_target CSU 2.6.9.8.21
fix_coords CSU 2.6.7.1.28
flyingveh.c CSC 2.6.4.6
flyingveh.h CSU 2.6.4.7
fly_missile CSU 2.6.9.1.11
fly_round CSU 2.6.9.8.22
fmat_to_rmat CSU 2.14.3.5.16
fncl.c CSC 2.2.1.1
followvehicle CSU 2.6.3.2.14
follow_lake_around CSU 2.12.1.12.8
follow_water_segments CSU 2.12.1.8.3
forceID_to_string CSU 2.9.2.1.3
formation.c CSC 2.6.8.4
formation.c CSC 2.8.1.4
Forward1Char CSU 2.5.1.4.15
ForwardChar CSU 2.5.1.4.24
fragment_route CSU 2.10.2.4.12
Free CSU 2.5.1.9.2
free_crossings CSU 2.12.1.11.20
free_function CSU 2.2.1.3.2
free_list CSU 2.12.1.7.4
free_stack CSU 2.1.1.1.3
free_water_list CSU 2.12.1.4.2
ft_float CSU 2.14.1.2.12
ft_int CSU 2.14.1.2.11
ft_symbol CSU 2.14.1.2.13
ft_table CSU 2.14.1.2.14
ft_untagged_table CSU 2.14.1.2.15
fvec_copy CSU 2.14.3.5.17
fvec_to_rvec CSU 2.14.3.5.13
gasp CSU 2.1.3.1.1
generate_a_deactivate CSU 2.3.1.1
generate_indirect_fire_packet CSU 2.3.1.4
generate_in_place_route CSU 2.10.2.4.8

generate_route_from_route_msg CSU 2.10.2.4.7
generate_status_report CSU 2.14.1.1.11
generate_weapon_hit CSU 2.6.9.8.24
generate_weapon_miss CSU 2.6.9.8.25
geometry.c CSC 2.14.3.6
get_args CSU 2.2.1.1.3
get_a_leader CSU 2.8.1.1.10
get_building_indecies CSU 2.12.1.17.4
get_control_measure CSU 2.10.2.1.3
get_formation_layout CSU 2.8.1.4.1
get_guises CSU 2.6.1.1.8
get_impact_and_trajectory CSU 2.6.4.5.1
get_i_and_t_from_normal CSU 2.6.4.5.2
get_leader_state CSU 2.6.3.2.13
get_local_address CSU 2.4.2.1.6
get_me_a_random_fraction CSU 2.14.3.7.1
get_mid_pt CSU 2.6.7.1.9
get_millisecond_time CSU 2.14.3.3.2
get_next_circle_point CSU 2.6.4.2.23
get_object_name CSU 2.6.7.1.55
get_overlay CSU 2.10.2.1.2
get_quads_in_region CSU 2.12.1.11.13
get_quads_in_rt_point_region CSU 2.12.1.11.14
get_quads_in_vector_region CSU 2.12.1.11.15
get_quad_nodes CSU 2.12.1.5.2
get_quad_nodes_internal CSU 2.12.1.5.3
get_relative_vehicle_agility CSU 2.6.9.8.20
get_remote_address CSU 2.4.2.1.7
get_sbx_from_port CSU 2.4.3.2.5
get_sender_of_last_packet CSU 2.3.2.15
get_soil_type CSU 2.14.1.2.1
get_station_info CSU 2.8.1.4.10
get_string CSU 2.1.1.3.3
get_symbol CSU 2.1.1.3.2
get_symbol_value CSU 2.1.1.3.6
get_turret CSU 2.8.1.4.2
get_unit_direction CSU 2.6.3.2.11
get_water_indecies CSU 2.12.1.4.1
GLOBAL CSC 2.13
Global Variables CSC 2.13.1
globals.h CSU 2.13.1.1
GobbleWord CSU 2.5.1.4.16
go_away CSU 2.14.1.1.6

grid_locword CSU 2.6.7.1.27
Ground Maneuver CSC 2.6.3
groundveh.c CSC 2.6.3.4
groundveh_close_enough CSU 2.6.3.4.6
groundveh_mobility_kill CSU 2.6.3.4.5
groundveh_move CSU 2.6.3.4.3
groundveh_tick CSU 2.6.3.4.4
ground_impact_to_sbx CSU 2.4.3.2.24
ground_level CSU 2.6.4.2.8
gunner.c CSC 2.6.9.5
gunner_round_flying CSU 2.6.9.5.6
gunner_tick CSU 2.6.9.5.7
hasty_attack_needed CSU 2.10.1.3.3
has_this_packet_been_acked CSU 2.4.2.3.14
heading_to_compass CSU 2.8.2.1.5
heap_allocate CSU 2.14.2.1.1
heap_allocate_distributed CSU 2.14.2.1.3
heap_calloc CSU 2.14.2.1.2
heap_create CSU 2.14.2.2.2
heap_deallocate CSU 2.14.2.1.4
heap_destroy CSU 2.14.2.2.3
heap_hang CSU 2.14.2.1.14
heap_inconsistent_action CSU 2.14.2.1.13
heap_print_statistics CSU 2.14.2.1.10
heap_verify CSU 2.14.2.1.12
helo.c CSC 2.6.4.1
helo_tick CSU 2.6.4.1.1
hitmodel.c CSC 2.6.8.3
host_local_vehicle_predicate CSU 2.9.3.1.9
host_loopback_function CSU 2.3.2.6
hull_to_world_from_direction CSU 2.6.9.1.1
hull_to_world_from_orientation CSU 2.6.4.6.8
icon_from_object_type CSU 2.4.3.1.9
identify_exercise CSU 2.5.2.1.18
identify_version CSU 2.1.3.2.1
idle_tick CSU 2.6.4.2.60
if_damage.c CSC 2.6.8.7
impact.c CSC 2.6.4.5
impact_weapon CSU 2.6.9.8.23
indirect_fire_to_sbx CSU 2.4.3.2.23
induce_roll CSU 2.6.4.6.11
induce_tail_spin CSU 2.6.4.6.10
inflate_rect CSU 2.14.3.6.4

INITIALIZATION CSC	2.1.
Initialization CSC	2.1.3
initialize_heap CSU	2.14.2.2.1
initialize_weapon_priority_list CSU	2.6.9.8.13
InitParser CSU	2.5.1.4.2
init_clocks CSU	2.14.3.3.1
init_free_list CSU	2.14.2.1.5
init_global_symbols CSU	2.1.2.1.1
init_grid_entry_list CSU	2.9.3.1.11
init_grid_tables CSU	2.9.3.1.10
init_mappings CSU	2.4.3.1.1
init_pilot_state_machine CSU	2.6.4.2.3
init_rect CSU	2.14.3.6.3
init_safobj_table CSU	2.2.2.1
init_static_matricies CSU	2.6.4.6.9
init_stuff CSU	2.1.3.1.3
init_symbol_table CSU	2.1.1.3.1
init_target_items CSU	2.6.9.3.13
init_target_list CSU	2.6.9.3.6
init_terrain_stuff CSU	2.1.3.1.5
InsertChar CSU	2.5.1.4.23
insert_function CSU	2.2.1.3.5
insert_grid_entry CSU	2.9.3.1.15
insert_periodic_function CSU	2.2.1.3.4
interesting_packet_test CSU	2.3.2.11
interior_angle_between_vectors CSU	2.14.3.5.8
INTERNAL STRUCTURE	1.3
interpolate CSU	2.6.7.1.56
interpolate.c CSC	2.6.8.1
interpolate_curve CSU	2.6.8.1.1
intersection.c CSC	2.12.1.14
intersection_direction CSU	2.12.1.11.2
intersect_rect CSU	2.14.3.6.2
intervis.c CSC	2.6.5.2
Intervisibility CSC	2.6.5
intervis_can_see_pt_to_pt CSU	2.6.5.2.1
intervis_get_high_ground CSU	2.6.5.2.5
intervis_get_view CSU	2.6.5.2.2
intervis_possibly_visible CSU	2.6.5.2.3
intervis_pt_to_pt CSU	2.6.5.2.4
int_dist CSU	2.8.1.4.5
invoke CSU	2.2.1.2.1
invoke.c CSC	2.2.1.2

invoke_functions_until CSU 2.2.1.5.1
in_object2 CSU 2.6.7.2.2
isolate_connection CSU 2.1.3.1.14
issue_reports CSU 2.8.2.1.43
is_cm_executed_in_list CSU 2.10.2.1.18
is_probably_a_string CSU 2.1.1.4.1
iterator.c CSC 2.9.3.1
iterator.h CSU 2.9.3.2
Iterators CSC 2.9.3
KillForward CSU 2.5.1.4.18
kludge.c CSU 2.5.1.10
lakes_thru CSU 2.12.1.4.12
last_item CSU 2.12.1.7.10
last_item_butlast CSU 2.12.1.7.11
LeftWord CSU 2.5.1.4.20
left_bottom_region CSU 2.12.1.13.4
left_column CSU 2.12.1.13.2
left_edge CSU 2.12.1.13.5
length CSU 2.12.1.7.16
lexer.l CSU 2.1.1.2
libcomm CSC 2.14.4
libcomm.h CSU 2.14.4.3
libdatabase CSC 2.6.8
libdatabase.h CSU 2.6.8.9
libheap CSC 2.14.2
libheap.h CSU 2.14.2.3
libparser CSC 2.5.1
libparser.h CSU 2.5.1.11
libpvis CSC 2.6.7
libquad CSC 2.12.1
libquad.h CSU 2.12.1.18
libreader CSC 2.1.1
libreader.h CSU 2.1.1.5
libsched CSC 2.2.1
libsched.h CSU 2.2.1.6
libudp CSC 2.4.2
libutil CSC 2.14.3
libutil.h CSU 2.14.3.9
line_cross_rect CSU 2.14.3.6.1
line_intersection CSU 2.12.1.14.8
line_intersection_core CSU 2.12.1.14.10
list.c CSC 2.12.1.7
loader.c CSC 2.6.9.10

loader_tick CSU 2.6.9.10.1
LOCAL VEHICLES CSC 2.6
Local Vehicles Main CSC 2.6.1
logistics.c CSC 2.6.6.1
logistics.h CSU 2.6.6.2
logistics_remove_vehicles CSU 2.6.6.1.3
logistics_supply_offer_canceled CSU 2.6.6.1.8
logistics_supply_offer_received CSU 2.6.6.1.9
logistics_tick CSU 2.6.6.1.10
longpt.h CSU 2.13.3.3
LookForward CSU 2.5.1.4.28
lookup_cis_communication_string CSU 2.10.1.1.4
lookup_cis_default_speed CSU 2.10.1.1.3
lookup_cis_enabled_predicates CSU 2.10.1.1.6
lookup_cis_formation CSU 2.10.1.1.7
lookup_cis_movement_method CSU 2.10.1.1.5
lookup_cis_resumable CSU 2.10.1.1.2
lookup_cis_situational CSU 2.10.1.1.1
lookup_vehicle_with_range_check CSU 2.5.2.1.19
Macros and Constants CSC 2.13.3
main CSU 2.1.3.1.19
main CSU 2.14.4.1.17
main CSU 2.15.1.1
main.c CSC 2.1.3.1
maint.c CSC 2.2.1.3
major_detection_increase CSU 2.6.5.1.13
major_spotter_increase CSU 2.8.2.3.10
make_array CSU 2.1.1.1.4
make_create_list CSU 2.11.1.1
make_leader CSU 2.8.1.4.8
make_me_non_blocking CSU 2.4.2.1.10
make_polygon CSU 2.14.3.6.5
make_vector_2d CSU 2.12.1.9.1
make_vector_from_angle_magnitude CSU 2.14.3.5.3
map.c CSC 2.4.3.1
map_echelon_number_to_echelon_symbol CSU 2.4.3.1.4
map_echelon_symbol_to_echelon_number CSU 2.4.3.1.3
map_echelon_type_number_to_echelon_type_symbol CSU 2.4.3.1.6
map_echelon_type_symbol_to_echelon_type_number CSU 2.4.3.1.5
map_echelon_type_symbol_to_icon_number CSU 2.4.3.1.7
map_icon_symbol_to_icon_number CSU 2.4.3.1.2
map_role_sym_to_role_number CSU 2.6.9.3.1
MarkCursor CSU 2.5.1.4.8

mass_stdc.h CSU 2.13.3.4
master_rudp_close CSU 2.4.2.3.6
master_rudp_open CSU 2.4.2.3.4
master_rudp_synch CSU 2.4.2.3.8
master_udp_open CSU 2.4.2.1.11
math.c CSC 2.14.3.5
member CSU 2.12.1.7.6
messages.h CSU 2.4.1.1
minimum_clip CSU 2.12.1.13.1
minimum_clip.c CSC 2.12.1.13
minmax.h CSU 2.13.3.5
misc.c CSC 2.14.1.2
missile.c CSC 2.6.9.1
missile.h CSU 2.6.9.2
missile_deactivate CSU 2.6.9.1.8
missile_ground_impact CSU 2.6.9.1.7
missile_maybe_hit_target CSU 2.6.9.1.10
missile_send_appearance CSU 2.6.9.1.13
missile_set_desired_direction CSU 2.6.9.1.12
missile_show CSU 2.6.9.1.5
missile_state_to_string CSU 2.6.9.1.2
mission_stationpoint CSU 2.6.3.2.9
muzzle_position_in_world_coordinates CSU 2.6.9.8.18
navigator.c CSC 2.10.2.3
navigator_entered_area CSU 2.10.2.3.9
navigator_entered_zone CSU 2.10.2.3.11
navigator_generate_report CSU 2.10.2.3.19
navigator_left_area CSU 2.10.2.3.10
navigator_left_zone CSU 2.10.2.3.12
navigator_passed_line CSU 2.10.2.3.8
navigator_passed_point CSU 2.10.2.3.7
navigator_reset_overlay CSU 2.10.2.3.5
navigator_set_overlay CSU 2.10.2.3.4
navigator_show CSU 2.10.2.3.3
navigator_tick CSU 2.10.2.3.6
nconc CSU 2.12.1.7.9
ncons CSU 2.12.1.7.2
need_decluster CSU 2.8.2.1.23
network_silent CSU 2.1.3.1.11
NextChar CSU 2.5.1.4.7
NextCommand CSU 2.5.1.8.4
next_road_point CSU 2.12.1.15.5
next_route_point CSU 2.6.3.2.7

noa_damp CSU 2.14.1.2.7
normalize_and_rotate CSU 2.12.1.11.4
nthcdr CSU 2.12.1.7.8
objects CSU 2.6.7.1.12
obstacle CSU 2.6.7.1.13
offset_point CSU 2.12.1.12.5
offset_points CSU 2.12.1.12.4
open_io_connections CSU 2.4.3.2.1
open_line_intersection CSU 2.12.1.14.9
OPORDERS CSC 2.1
orbit_velocity CSU 2.6.4.2.12
or_bitfield CSU 2.14.3.4.8
p2_bottom CSU 2.12.1.13.6
Parameters CSC 2.1.2
ParseCommandDone CSU 2.5.1.1.4
ParseConvertNumber CSU 2.5.1.6.7
ParseDoKeywordTable CSU 2.5.1.1.5
ParseError CSU 2.5.1.7.7
ParseEscapeComplete CSU 2.5.1.7.12
ParseFindEndList CSU 2.5.1.7.9
ParseGetConstant CSU 2.5.1.2.1
ParseGetDecimal CSU 2.5.1.6.1
ParseGetFields CSU 2.5.1.5.1
ParseGetHex CSU 2.5.1.6.2
ParseGetOctal CSU 2.5.1.6.3
ParseGetString CSU 2.5.1.6.4
ParseGetToken CSU 2.5.1.7.6
ParseInput CSU 2.5.1.1.1
ParseInSet CSU 2.5.1.7.5
ParseKeyword CSU 2.5.1.1.3
ParseList CSU 2.5.1.1.2
ParseMatch CSU 2.5.1.7.10
ParseMessage CSU 2.5.1.7.8
ParseMustFree CSU 2.5.1.7.4
ParseOptional CSU 2.5.1.6.6
ParsePrint CSU 2.5.1.9.3
ParsePutArg CSU 2.5.1.6.5
Parser Command Processor CSC 2.5.2
PARSER INTERFACE CSC 2.5
parser.c CSC 2.5.2.1
parser.y CSC 2.1.1.1
parser_create CSU 2.5.2.1.3
parser_create_vehicle CSU 2.5.2.1.33

parser_global_reset CSU 2.5.2.1.35
parser_heap_collect CSU 2.5.2.1.39
parser_heap_statistics CSU 2.5.2.1.36
parser_heap_verify CSU 2.5.2.1.37
parser_init CSU 2.5.2.1.1
parser_restore_term CSU 2.5.2.1.2
parser_send_string CSU 2.5.2.1.41
parser_set_targeting_parameters CSU 2.5.2.1.40
ParseTableFind CSU 2.5.1.7.11
par_base.c CSC 2.5.1.1
par_const.c CSC 2.5.1.2
par_copy.c CSC 2.5.1.3
par_edit.c CSC 2.5.1.4
par_fields.c CSC 2.5.1.5
par_hist.c CSC 2.5.1.8
par_lib.c CSC 2.5.1.6
par_unix.c CSC 2.5.1.9
par_util.c CSC 2.5.1.7
patch CSU 2.6.7.1.6
perf.c CSC 2.2.1.4
perf_monitor_off CSU 2.2.1.4.2
perf_monitor_on CSU 2.2.1.4.1
periodic_fncl CSU 2.2.1.1.2
phantom.h CSU 2.13.3.1
phase_two_collision_check CSU 2.6.3.1.15
pilot.c CSC 2.6.4.2
pilot_cancel_immediate CSU 2.6.4.2.71
pilot_change_altitude_im CSU 2.6.4.2.66
pilot_change_speed_im CSU 2.6.4.2.68
pilot_check_state CSU 2.6.4.2.63
pilot_execute_overlay CSU 2.6.4.2.85
pilot_executing_immediate_command CSU 2.6.4.2.72
pilot_face_direction CSU 2.6.4.2.58
pilot_face_direction_im CSU 2.6.4.2.74
pilot_followroute_tick CSU 2.6.4.2.45
pilot_followvehicle CSU 2.6.4.2.57
pilot_follow_leader CSU 2.6.4.2.80
pilot_follow_leader_tick CSU 2.6.4.3.22
pilot_follow_vehicle CSU 2.6.4.2.56
pilot_follow_vehicle_im CSU 2.6.4.2.73
pilot_get_altitude CSU 2.6.4.2.65
pilot_get_asm CSU 2.6.4.2.84
pilot_get_speed CSU 2.6.4.2.67

pilot_gotopoint CSU	2.6.4.2.27
pilot_goto_endpoint CSU	2.6.4.2.28
pilot_goto_point_im CSU	2.6.4.2.69
pilot_hold CSU	2.6.4.2.29
pilot_hold_im CSU	2.6.4.2.76
pilot_hoverattack CSU	2.6.4.2.51
pilot_hoverattack_approach CSU	2.6.4.2.48
pilot_hoverattack_complete CSU	2.6.4.2.54
pilot_hoverattack_im CSU	2.6.4.2.75
pilot_hoverattack_tick CSU	2.6.4.2.55
pilot_hoverhold CSU	2.6.4.2.32
pilot_hoverhold_tick CSU	2.6.4.2.33
pilot_init_followroute CSU	2.6.4.2.44
pilot_init_hoverattack CSU	2.6.4.2.46
pilot_init_hoverhold CSU	2.6.4.2.30
pilot_init_land CSU	2.6.4.2.39
pilot_init_orbithold CSU	2.6.4.2.34
pilot_is_facing_direction CSU	2.6.4.2.16
pilot_landhold_tick CSU	2.6.4.2.43
pilot_landing CSU	2.6.4.2.41
pilot_land_im CSU	2.6.4.2.70
pilot_mission_completed CSU	2.6.4.2.77
pilot_on_same_route CSU	2.6.4.2.15
pilot_orbithold CSU	2.6.4.2.36
pilot_orbit_tick CSU	2.6.4.2.37
pilot_point_at_target CSU	2.6.4.2.50
pilot_racetrackhold CSU	2.6.4.2.38
pilot_remove_vehicles CSU	2.6.4.2.5
pilot_set_leader_mis CSU	2.6.4.2.79
pilot_set_route_mis CSU	2.6.4.2.81
pilot_set_speed_mis CSU	2.6.4.2.83
pilot_show CSU	2.6.4.2.7
pilot_show_machine CSU	2.6.4.2.6
pilot_start_gotopoint CSU	2.6.4.2.26
pilot_start_hoverattack CSU	2.6.4.2.49
pilot_start_hoverattack_approach CSU	2.6.4.2.47
pilot_start_hoverattack_complete CSU	2.6.4.2.53
pilot_start_hoverattack_egress CSU	2.6.4.2.52
pilot_start_hoverhold CSU	2.6.4.2.31
pilot_start_landed CSU	2.6.4.2.42
pilot_start_landing CSU	2.6.4.2.40
pilot_start_orbithold CSU	2.6.4.2.35
pilot_start_takingoff CSU	2.6.4.2.24

pilot_state_to_string CSU 2.6.4.2.1
pilot_stationpoint CSU 2.6.4.2.78
pilot_stop_mission CSU 2.6.4.2.82
pilot_takingoff CSU 2.6.4.2.25
pilot_tick CSU 2.6.4.2.64
pilot_tick2 CSU 2.6.4.2.59
plane.c CSC 2.6.4.4
plane_tick CSU 2.6.4.4.1
point_inside_polygon CSU 2.12.1.14.3
point_in_building CSU 2.12.1.17.2
point_in_bv CSU 2.6.7.2.1
point_in_polygon CSU 2.14.3.6.6
point_line_intersection CSU 2.12.1.14.6
point_segment_intersection CSU 2.12.1.14.7
point_weapon_at_target CSU 2.6.9.5.3
poll_request_to_string CSU 2.4.3.2.6
pop_cis CSU 2.10.1.1.12
pop_resumable_cis CSU 2.10.1.1.13
possible_intersection CSU 2.12.1.14.2
predicates.c CSC 2.10.1.3
predicates.h CSU 2.10.1.4
predicates_show CSU 2.10.1.3.9
predicate_impact_hook CSU 2.10.1.3.7
PreviousCommand CSU 2.5.1.8.3
print.c CSC 2.12.1.6
printdirection CSU 2.6.3.2.40
printimmediate CSU 2.6.3.2.42
printmission CSU 2.6.3.2.39
printroutedirection CSU 2.6.3.2.41
print_block CSU 2.14.2.1.15
print_bridge CSU 2.12.1.3.2
print_building CSU 2.12.1.16.6
print_ch_pt_info CSU 2.6.7.1.23
print_commands_from_sbx CSU 2.1.3.1.8
print_debug_list CSU 2.6.7.1.47
print_edge CSU 2.6.7.1.36
print_end_info CSU 2.6.7.1.52
print_error CSU 2.6.7.1.46
print_fpe_info CSU 2.6.7.1.43
print_guard_info CSU 2.6.7.1.49
print_heap_table CSU 2.14.2.1.11
print_hitpt CSU 2.6.7.1.42
print_lake CSU 2.12.1.3.4

print_lists CSU 2.8.1.4.9
print_matrix CSU 2.14.3.5.1
print_message_position CSU 2.4.3.2.77
print_object_info CSU 2.6.7.1.40
print_opfor_header CSU 2.4.3.2.76
print_patch_info CSU 2.6.7.1.50
print_performance_stats CSU 2.2.1.4.6
print_pvparams CSU 2.6.7.1.44
print_quad_node CSU 2.12.1.6.1
print_reasons_and_clear CSU 2.5.2.1.38
print_river_intersection CSU 2.12.1.2.4
print_river_segment CSU 2.12.1.2.2
print_road_intersection CSU 2.12.1.15.4
print_road_segment CSU 2.12.1.15.2
print_route CSU 2.10.2.4.17
print_routepoint CSU 2.10.2.4.16
print_route_list CSU 2.12.1.11.22
print_short_pvparams CSU 2.6.7.1.45
print_sockaddr CSU 2.4.2.1.8
print_terrain_point CSU 2.6.7.1.57
print_tree CSU 2.12.1.16.2
print_tree_canopy CSU 2.12.1.16.4
print_vector CSU 2.14.3.5.2
print_vector_list CSU 2.12.1.11.23
print_vehicle_ids CSU 2.4.3.2.78
print_vehicle_marking CSU 2.9.2.2.3
proc_switches CSU 2.14.3.1.1
project_point CSU 2.6.4.2.21
protocol_data_process CSU 2.3.2.14
protocol_sim_process CSU 2.3.2.12
protocol_stealth_process CSU 2.3.2.13
prototypes.h CSU 2.13.2.1
pro_sim.c CSC 2.3.1
prune_to_point CSU 2.12.1.12.6
push CSU 2.12.1.7.3
push_cis CSU 2.10.1.1.11
push_vector_2d_on_points CSU 2.12.1.11.17
push_vector_2d_on_route CSU 2.12.1.11.18
pve.h CSU 2.6.7.3
pve_checkvis CSU 2.6.7.1.1
pve_checkvis.c CSC 2.6.7.1
pvis_call.h CSU 2.6.7.4
pv_bv.c CSC 2.6.7.2

p_follower.c CSC	2.6.4.3
p_follower_am_i_flying_coord CSU	2.6.4.3.1
p_follower_arrive_at_same_time CSU	2.6.4.3.5
p_follower_comp_reset_route CSU	2.6.4.3.4
p_follower_find_pnt_in_turn CSU	2.6.4.3.24
p_follower_flip_in_turn CSU	2.6.4.3.23
p_follower_flying_independent CSU	2.6.4.3.12
p_follower_fly_in_coord_pos CSU	2.6.4.3.3
p_follower_followroute_tick CSU	2.6.4.3.13
p_follower_gen_coord_goal_pnt CSU	2.6.4.3.9
p_follower_get_wrld_offset CSU	2.6.4.3.7
p_follower_goto_pnt_coord CSU	2.6.4.3.17
p_follower_hoverattack_tick CSU	2.6.4.3.16
p_follower_hoverhold CSU	2.6.4.3.18
p_follower_hoverhold_tick CSU	2.6.4.3.21
p_follower_land CSU	2.6.4.3.19
p_follower_landhold_tick CSU	2.6.4.3.20
p_follower_leader_passed_point CSU	2.6.4.3.14
p_follower_new_rel_rt_pnt CSU	2.6.4.3.11
p_follower_passed_point CSU	2.6.4.3.15
p_follower_pnt_in_hard_turn CSU	2.6.4.3.27
p_follower_pnt_in_no_turn CSU	2.6.4.3.25
p_follower_pnt_in_shallow_turn CSU	2.6.4.3.26
p_follower_set_desired_vel CSU	2.6.4.3.6
p_follower_set_follow CSU	2.6.4.3.10
p_follower_stop_coord CSU	2.6.4.3.8
p_follow_form_flip CSU	2.6.4.3.2
queue_allocate CSU	2.14.4.2.1
queue_dequeue CSU	2.14.4.2.6
queue_describe CSU	2.14.4.2.11
queue_distribute CSU	2.14.4.2.25
queue_enqueue CSU	2.14.4.2.5
queue_excise CSU	2.14.4.2.7
queue_fix.c CSC	2.14.4.2
queue_flush CSU	2.14.4.2.10
queue_flush_hangers CSU	2.14.4.2.9
queue_length CSU	2.14.4.2.2
queue_traverse_and_apply CSU	2.14.4.2.8
r2 CSU	2.14.1.2.6
radiate_target CSU	2.6.9.8.19
random.c CSC	2.14.3.7
randomize_vector CSU	2.6.4.5.3
range_squared CSU	2.14.3.5.10

rassoc_sym CSU 2.4.3.1.8
reader_read_file CSU 2.1.1.1.7
readjust_overlay CSU 2.14.1.1.9
read_feature CSU 2.12.1.1.4
read_machine_file CSU 2.1.3.1.4
read_quadtree CSU 2.12.1.1.3
read_quadtree_database CSU 2.12.1.1.1
read_terrain_files CSU 2.12.1.1.2
RedisplayLine CSU 2.5.1.4.5
relax_points CSU 2.12.1.11.10
relax_points_aux CSU 2.12.1.11.11
rel_conn.c CSC 2.4.2.3
rel_conn.h CSU 2.4.2.4
REMOTE VEHICLES CSC 2.7
remote.c CSC 2.7.1
remote_change_stealth_controlling_port CSU 2.7.1.16
remote_deactivate CSU 2.7.1.15
remote_fill_echelon_data CSU 2.7.1.7
remote_fill_in_appearance_data CSU 2.7.1.6
remote_fill_in_position_data CSU 2.7.1.8
remote_go_away CSU 2.7.1.3
remote_init_vars CSU 2.7.1.2
remote_new_appearance CSU 2.7.1.12
remote_next_road_point CSU 2.7.1.9
remote_show CSU 2.7.1.4
remote_start_being_watched CSU 2.7.1.10
remote_start_ticking CSU 2.7.1.5
remote_stop_being_watched CSU 2.7.1.11
remote_tick CSU 2.7.1.13
remove_clustered_vehicle CSU 2.8.2.1.26
remove_dead_clustered_vehicles CSU 2.8.2.1.31
remove_duplicates CSU 2.12.1.7.5
remove_executing_unit CSU 2.10.2.1.22
remove_from_free_list CSU 2.14.2.1.8
remove_function CSU 2.2.1.3.3
remove_grid_entry CSU 2.9.3.1.14
remove_removed_vehicle CSU 2.8.2.1.30
remove_shelling_cluster CSU 2.8.2.1.38
remove_vehicles CSU 2.14.1.1.5
reporter.c CSC 2.8.2.1
reporter.h CSU 2.8.2.2
reporter_remove_vehicles CSU 2.8.2.1.3
reporter_show CSU 2.8.2.1.4

reporter_tick CSU 2.8.2.1.41
Reports CSC 2.8.2
report_cluster_vehicle_type CSU 2.8.2.1.45
report_edge_hit CSU 2.6.7.1.35
report_error_from_tdb_once CSU 2.14.1.2.4
report_hit CSU 2.6.7.1.22
report_ivis_network CSU 2.8.2.1.44
report_last_hit CSU 2.6.7.1.25
report_tree_block CSU 2.6.7.1.17
report_vis_change CSU 2.6.7.1.54
reset_predicates CSU 2.10.1.3.10
RestoreCursor CSU 2.5.1.4.9
resuply_check_ok CSU 2.6.6.1.5
Resupply CSC 2.6.6
resupply_bld_ammo_needs CSU 2.6.6.1.6
reverse CSU 2.12.1.7.15
re_assign_routepoints CSU 2.10.2.4.13
RightWord CSU 2.5.1.4.21
rivers.c CSC 2.12.1.2
rmat_to_fmat CSU 2.14.3.5.15
roads.c CSC 2.12.1.15
round_real_to_int CSU 2.14.3.5.11
route.c CSC 2.10.2.4
route.h CSU 2.10.2.5
routepoint_distance CSU 2.10.2.4.1
rt_point_to_vector CSU 2.12.1.11.21
rudp_ack_received CSU 2.4.2.3.15
rudp_close CSU 2.4.2.3.5
rudp_discard_all_buffers CSU 2.4.2.3.10
rudp_disconnect CSU 2.4.2.3.9
rudp_init CSU 2.4.2.3.1
rudp_open CSU 2.4.2.3.3
rudp_post_message CSU 2.4.2.3.12
rudp_read_message CSU 2.4.2.3.11
rudp_retransmit CSU 2.4.2.3.17
rudp_retransmit_buf CSU 2.4.2.3.16
rudp_send CSU 2.4.2.3.13
rudp_send_bare_ack CSU 2.4.2.3.18
rudp_show CSU 2.4.2.3.2
rudp_synch CSU 2.4.2.3.7
rudp_tick CSU 2.4.2.3.19
rvec_to_fvec CSU 2.14.3.5.14
SAF COMMAND INTERFACE CSC 2.4

SAF Command Processor CSC 2.4.3
SAF Command Protocol CSC 2.4.1
SAF Object Operations CSC 2.9.2
SAF Object Structures CSC 2.9.1
SAF OBJECTS CSC 2.9
saf.c CSC 2.2.4
safobj.c CSC 2.2.2
safobj.h CSU 2.9.1.1
saf_complete_reset CSU 2.2.4.1
saf_exit CSU 2.1.3.1.22
SAF_NET_SND_KLUDGE CSU 2.3.2.5
saf_print_help CSU 2.1.3.1.17
saf_remove_vehicle CSU 2.2.4.3
saf_reset_for_workstation CSU 2.2.4.2
saf_vehicle.c CSC 2.6.1.1
saf_vehicle_cancel_overlay CSU 2.6.1.1.47
saf_vehicle_catastrophic_kill CSU 2.6.1.1.25
saf_vehicle_change_speed CSU 2.6.1.1.53
saf_vehicle_checkpoint_state CSU 2.6.1.1.16
saf_vehicle_collision_over CSU 2.6.1.1.42
saf_vehicle_doing_collision_stuff CSU 2.6.1.1.43
saf_vehicle_drain_supplies CSU 2.6.1.1.26
saf_vehicle_est_position CSU 2.6.1.1.58
saf_vehicle_execute_overlay CSU 2.6.1.1.46
saf_vehicle_face_direction CSU 2.6.1.1.57
saf_vehicle_fake_ resupply CSU 2.6.1.1.33
saf_vehicle_fill_in_appearance_data CSU 2.6.1.1.13
saf_vehicle_fill_in_echelon_data CSU 2.6.1.1.14
saf_vehicle_fill_in_position_data CSU 2.6.1.1.15
saf_vehicle_firepower_kill CSU 2.6.1.1.22
saf_vehicle_follow_vehicle CSU 2.6.1.1.54
saf_vehicle_generate_status_report CSU 2.6.1.1.34
saf_vehicle_goto_point CSU 2.6.1.1.55
saf_vehicle_go_away CSU 2.6.1.1.2
saf_vehicle_halt CSU 2.6.1.1.52
saf_vehicle_indirect_fire CSU 2.6.1.1.38
saf_vehicle_mimic_vehicle CSU 2.6.1.1.18
saf_vehicle_mission_completed CSU 2.6.1.1.40
saf_vehicle_mobility_kill CSU 2.6.1.1.23
saf_vehicle_next_event_id CSU 2.6.1.1.28
saf_vehicle_out_of_ammo CSU 2.6.1.1.32
saf_vehicle_out_of_gas CSU 2.6.1.1.31
saf_vehicle_pro_sim CSU 2.6.1.1.39

saf_vehicle_reinit CSU 2.6.1.1.17
saf_vehicle_rejoin_unit CSU 2.6.1.1.45
saf_vehicle_remove_vehicles CSU 2.6.1.1.35
saf_vehicle_reset_station_keeper CSU 2.6.1.1.51
saf_vehicle_resume_mission CSU 2.6.1.1.56
saf_vehicle_send_appearance CSU 2.6.1.1.21
saf_vehicle_set_direction CSU 2.6.1.1.50
saf_vehicle_set_leader CSU 2.6.1.1.5
saf_vehicle_set_marking CSU 2.6.1.1.9
saf_vehicle_set_route CSU 2.6.1.1.48
saf_vehicle_set_speed CSU 2.6.1.1.49
saf_vehicle_set_superior CSU 2.6.1.1.4
saf_vehicle_set_targeting_parameters CSU 2.6.1.1.12
saf_vehicle_show CSU 2.6.1.1.11
saf_vehicle_simulator_in_command CSU 2.6.1.1.44
saf_vehicle_start_ticking CSU 2.6.1.1.3
saf_vehicle_stop_flaming CSU 2.6.1.1.24
saf_vehicle_stop_mimicing CSU 2.6.1.1.19
saf_vehicle_sudden_stop CSU 2.6.1.1.27
saf_vehicle_supplies_needed CSU 2.6.1.1.29
saf_vehicle_supplies_provided CSU 2.6.1.1.30
saf_vehicle_teleport CSU 2.6.1.1.7
saf_vehicle_teleport_to_station CSU 2.6.1.1.6
saf_vehicle_tick CSU 2.6.1.1.20
saf_vehicle_useless CSU 2.6.1.1.41
saf_vehicle_vehicle_impact CSU 2.6.1.1.36
saf_vehicle_vehicle_rammed CSU 2.6.1.1.37
sbx.c CSC 2.4.3.2
sbx.h CSU 2.4.3.3
sbx_add_area CSU 2.4.3.2.46
sbx_add_line CSU 2.4.3.2.48
sbx_add_point CSU 2.4.3.2.45
sbx_add_route CSU 2.4.3.2.49
sbx_add_zone CSU 2.4.3.2.47
sbx_altitude CSU 2.4.3.2.65
sbx_attach_stealth CSU 2.4.3.2.70
sbx_attack CSU 2.4.3.2.69
sbx_change_formation CSU 2.4.3.2.54
sbx_change_speed CSU 2.4.3.2.53
sbx_connection_create_unit_msg CSU 2.4.3.2.29
sbx_connection_disconnect CSU 2.4.3.2.14
sbx_connection_exit CSU 2.4.3.2.11
sbx_connection_generate_arty_msg CSU 2.4.3.2.37

sbx_connection_generate_reset_vehicle CSU 2.4.3.2.31
sbx_connection_init CSU 2.4.3.2.9
sbx_connection_open CSU 2.4.3.2.10
sbx_connection_overloaded CSU 2.4.3.2.16
sbx_connection_process_ground_impact CSU 2.4.3.2.21
sbx_connection_process_indirect_fire CSU 2.4.3.2.22
sbx_connection_process_message CSU 2.4.3.2.26
sbx_connection_process_messages CSU 2.4.3.2.27
sbx_connection_process_vehicle_impact CSU 2.4.3.2.20
sbx_connection_query_sub_state CSU 2.4.3.2.38
sbx_connection_reset_msg CSU 2.4.3.2.30
sbx_connection_send_to_port CSU 2.4.3.2.19
sbx_connection_service_poll_msg CSU 2.4.3.2.33
sbx_connection_set_ivis_parameters CSU 2.4.3.2.40
sbx_connection_set_ivis_xmit_modes CSU 2.4.3.2.39
sbx_connection_show CSU 2.4.3.2.12
sbx_connection_show_top_level_units CSU 2.4.3.2.79
sbx_connection_stat_pulse CSU 2.4.3.2.35
sbx_connection_synch_received CSU 2.4.3.2.13
sbx_connection_tick CSU 2.4.3.2.17
sbx_connection_vehicle_reinit CSU 2.4.3.2.32
sbx_connection_wat_pulse CSU 2.4.3.2.34
sbx_connection_write_buffer CSU 2.4.3.2.15
sbx_delete_cm CSU 2.4.3.2.55
sbx_delete_overlay CSU 2.4.3.2.50
sbx_execute_overlay CSU 2.4.3.2.51
sbx_face_direction CSU 2.4.3.2.61
sbx_follow_vehicle CSU 2.4.3.2.57
sbx_goto_point CSU 2.4.3.2.59
sbx_halt CSU 2.4.3.2.52
sbx_hold CSU 2.4.3.2.67
sbx_land CSU 2.4.3.2.63
sbx_printf CSU 2.4.3.2.8
sbx_rejoin_unit CSU 2.4.3.2.56
sbx_resume_mission CSU 2.4.3.2.60
sbx_resupply CSU 2.4.3.2.71
sbx_set_all_known_vehicles CSU 2.4.3.2.42
sbx_set_specific_known_vehicles CSU 2.4.3.2.43
sbx_set_targeting_parameters CSU 2.4.3.2.62
sbx_set_top_level_known_vehicles CSU 2.4.3.2.44
sbx_simulator_in_command CSU 2.4.3.2.58
sbx_swap_known_vehicles CSU 2.4.3.2.41
ScanBackwards CSU 2.5.1.4.29

ScanForward CSU 2.5.1.4.27
scan_weapon CSU 2.6.9.5.4
sched.c CSC 2.2.1.5
SCHEDULER CSC 2.2
scheduler_init CSU 2.2.1.3.1
search.c CSC 2.12.1.5
segment_inside_polygon CSU 2.12.1.14.5
segment_thru_lake CSU 2.12.1.4.7
segment_thru_river CSU 2.12.1.4.6
segment_thru_water CSU 2.12.1.4.5
seg_intersection CSU 2.12.1.14.1
select_weapon_priority_list CSU 2.6.9.3.18
send_contact_report CSU 2.8.2.1.46
send_shelling_report CSU 2.8.2.1.48
send_spot_report CSU 2.8.2.1.47
send_stealth_gone_msg CSU 2.7.1.14
SetStopPoint CSU 2.5.1.4.6
set_abort_on_error CSU 2.5.2.1.5
set_bit CSU 2.14.3.4.3
set_bitfield CSU 2.14.3.4.7
set_command_printing CSU 2.5.2.1.4
set_critical_performance_level CSU 2.2.1.4.3
set_exercise_id CSU 2.1.3.1.13
set_ground_impact_mode CSU 2.5.2.1.6
set_header_printing CSU 2.5.2.1.8
set_indirect_fire_mode CSU 2.5.2.1.7
set_monitor_period CSU 2.5.2.1.9
set_number_of_fake_remotes CSU 2.1.3.1.16
set_pv_params CSU 2.6.5.2.6
set_simnet_files CSU 2.1.3.1.12
set_speed_dir CSU 2.6.3.2.6
set_symbol_value CSU 2.1.1.3.7
set_targeting_parameters CSU 2.14.1.1.7
set_target_list CSU 2.6.9.3.7
set_target_machine CSU 2.1.3.1.15
set_tdb_to_cache CSU 2.1.3.1.10
set_terrain_dbase CSU 2.1.3.1.9
set_xor CSU 2.12.1.7.13
shell_report_needed CSU 2.8.2.1.40
shell_type_to_string CSU 2.8.2.1.33
shoot_target CSU 2.6.9.5.5
show CSU 2.14.1.1.4
show_applies_to CSU 2.10.2.1.43

show_area_cm CSU 2.10.2.1.40
show_connection CSU 2.5.2.1.16
show_connection_all_sbx CSU 2.4.3.2.4
show_control_measure CSU 2.10.2.1.37
show_control_measures CSU 2.10.2.1.36
show_curves CSU 2.6.8.8.3
show_detection_database CSU 2.6.8.2.1
show_df_damage_database CSU 2.6.8.6.1
show_df_damage_entry CSU 2.6.8.6.6
show_df_damage_object CSU 2.6.8.6.4
show_df_damage_side CSU 2.6.8.6.5
show_df_damage_veh CSU 2.6.8.6.2
show_df_damage_weapon CSU 2.6.8.6.3
show_hitmodel_database CSU 2.6.8.3.1
show_if_damage_database CSU 2.6.8.7.1
show_if_damage_entry CSU 2.6.8.7.5
show_if_damage_veh CSU 2.6.8.7.2
show_if_damage_weapon CSU 2.6.8.7.3
show_if_ranged_damage_entry CSU 2.6.8.7.4
show_line_cm CSU 2.10.2.1.42
show_overlay CSU 2.10.2.1.35
show_overlays CSU 2.10.2.1.34
show_overlays_all_sbx CSU 2.4.3.2.7
show_point_cm CSU 2.10.2.1.39
show_point_list CSU 2.10.2.1.38
show_removed_vehicles CSU 2.8.2.1.28
show_route CSU 2.10.2.4.14
show_route_points CSU 2.10.2.4.15
show_sbx_overlays CSU 2.5.2.1.17
show_shelling_clusters CSU 2.8.2.1.35
show_table_record CSU 2.6.8.8.2
show_vehicle CSU 2.5.2.1.20
show_vehicle_cluster CSU 2.8.2.1.8
show_vehicle_clusters CSU 2.8.2.1.7
show_vehicle_ids CSU 2.5.2.1.11
show_vehicle_id_list CSU 2.8.2.1.9
show_zone_cm CSU 2.10.2.1.41
sigh CSU 2.1.3.1.2
SIMNET INTERFACE CSC 2.3
simnet.c CSC 2.3.2
simnet_exit CSU 2.3.2.7
simnet_getstats CSU 2.3.2.8
simnet_tick CSU 2.3.2.2

simnet_zerostats CSU 2.3.2.9
simple_queue_allocate CSU 2.14.4.1.7
simple_queue_deallocate CSU 2.14.4.1.8
simple_queue_dequeue CSU 2.14.4.1.2
simple_queue_describe CSU 2.14.4.1.9
simple_queue_enqueue CSU 2.14.4.1.1
simple_queue_is_empty CSU 2.14.4.1.5
simple_queue_is_full CSU 2.14.4.1.4
simple_queue_is_this_power_of_two CSU 2.14.4.1.6
simple_queue_length CSU 2.14.4.1.3
sim_undef.h CSU 2.13.3.6
Situation Assessment CSC 2.8.3
skirt_lake CSU 2.12.1.12.7
skirt_river CSU 2.12.1.12.1
skirt_river_bend CSU 2.12.1.12.9
skirt_water.c CSC 2.12.1.12
skyline_error CSU 2.6.7.1.51
sockaddr_init CSU 2.4.2.1.9
sort_damage_table CSU 2.6.8.6.9
sort_form_db CSU 2.8.1.4.12
sort_tag_table CSU 2.1.1.4.5
sort_trim_hits CSU 2.6.7.1.39
sort_water CSU 2.12.1.4.10
splice_out_vehicle_cluster CSU 2.8.2.1.21
spotter.c CSC 2.8.2.3
spotter.h CSU 2.8.2.4
spotter_init_table CSU 2.8.2.3.5
spotter_remove_vehicles CSU 2.8.2.3.6
spotter_save_weight CSU 2.8.2.3.11
spotter_show CSU 2.8.2.3.12
spotter_show_table CSU 2.8.2.3.13
spotter_sum_inferior_composite_tables CSU 2.8.2.3.8
spotter_tick CSU 2.8.2.3.7
spread_increment CSU 2.8.1.4.3
stack_push CSU 2.1.1.1.1
stack_push_array CSU 2.1.1.1.2
startup CSU 2.6.7.1.2
start_avoiding_collision CSU 2.6.3.1.9
start_avoiding_house CSU 2.6.3.1.11
start_disengaging CSU 2.6.3.1.14
start_resupply_of_to CSU 2.6.6.1.4
start_simnet CSU 2.3.2.1
start_ticking CSU 2.2.3.3

stationpoint CSU 2.6.3.2.10
stealth_attach_to CSU 2.5.2.1.31
stealth_set_mimic_off CSU 2.5.2.1.29
stealth_set_mimic_on CSU 2.5.2.1.28
stealth_set_symbols_draw_tick CSU 2.5.2.1.27
stealth_site_host_pair CSU 2.5.2.1.30
stealth_teleport_to CSU 2.5.2.1.32
stop CSU 2.6.3.2.5
stop_ticking CSU 2.2.3.4
store_hit CSU 2.6.7.1.19
stringcopy CSU 2.5.1.3.1
string_for_altitude_type CSU 2.4.3.2.64
string_for_attack_type CSU 2.4.3.2.68
string_for_hold_type CSU 2.4.3.2.66
SUPPORT CSC 2.15
symbol.c CSC 2.1.1.3
symbols.c CSC 2.1.2.1
symbols.h CSU 2.1.2.2
tactical_state.h CSU 2.8.3.1
tags.c CSC 2.1.1.4
tag_error CSU 2.1.1.4.2
targeting.c CSC 2.6.9.3
targeting.h CSU 2.6.9.4
targeting_set_fire_at_pointair CSU 2.6.9.3.8
targeting_set_fire_at_will CSU 2.6.9.3.10
targeting_set_hold_fire CSU 2.6.9.3.9
targeting_set_parameters CSU 2.6.9.3.11
targeting_show CSU 2.6.9.3.5
targeting_tick CSU 2.6.9.3.21
target_in_position CSU 2.6.9.3.17
TARGET_ITEM CSU 2.6.9.3.12
target_priority CSU 2.6.9.3.15
target_type_ok CSU 2.6.9.3.16
tdb_get_gl CSU 2.14.1.2.2
tdb_get_zl CSU 2.14.1.2.3
TERMINOLOGY AND DOCUMENTATION 1.5
TERRAIN CSC 2.12
terrain CSU 2.6.7.1.7
terrain.c CSC 2.12.1.1
test_clutter CSU 2.6.7.1.26
test_maxima CSU 2.6.7.1.5
test_mins CSU 2.6.7.1.4
tickable.c CSC 2.2.3

tickable_note_start_tick CSU 2.2.3.6
time.c CSC 2.14.3.3
toggle_debugging CSU 2.5.2.1.10
tolerance.h CSU 2.13.3.7
top_left_corner CSU 2.12.1.13.3
track_target CSU 2.6.9.5.2
treelines CSU 2.6.7.1.16
trees CSU 2.6.7.1.15
trees.c CSC 2.12.1.16
turret.c CSC 2.6.9.6
turret.h CSU 2.6.9.7
turret_firepower_kill CSU 2.6.9.6.10
turret_interest_dir CSU 2.6.9.6.9
turret_muzzle_position_in_world_coordinates CSU 2.6.9.6.7
turret_point_at_target CSU 2.6.9.6.5
turret_scan CSU 2.6.9.6.6
turret_set_scan_parms CSU 2.6.9.6.8
turret_show CSU 2.6.9.6.3
turret_slew CSU 2.6.9.6.4
TypeFields CSU 2.5.1.5.2
uc CSU 2.5.1.7.13
udp_berkeley.c CSC 2.4.2.1
udp_close CSU 2.4.2.1.2
udp_conn.h CSU 2.4.2.2
udp_open CSU 2.4.2.1.1
udp_read CSU 2.4.2.1.4
udp_show CSU 2.4.2.1.3
udp_write CSU 2.4.2.1.5
UNITS CSC 2.8
unit_send_pae_data CSU 2.8.1.3.30
upcase CSU 2.14.3.8.1
UpdateLine CSU 2.5.1.4.3
UpdateMoveCursor CSU 2.5.1.4.4
update_clusters CSU 2.8.2.1.32
update_cluster_com CSU 2.8.2.1.22
update_cluster_heading CSU 2.8.2.1.6
update_grid_entry_list CSU 2.9.3.1.12
update_reporter_vehicles CSU 2.8.2.1.42
update_target_list CSU 2.6.9.3.19
UTILITIES CSC 2.14
Utilities CSC 2.14.1
valid_quad CSU 2.12.1.5.4
vec2_add CSU 2.14.3.5.22

vec2_copy CSU 2.14.3.5.29
vec2_cross CSU 2.14.3.5.25
vec2_dot CSU 2.14.3.5.24
vec2_init CSU 2.14.3.5.20
vec2_mag CSU 2.14.3.5.26
vec2_mag2 CSU 2.14.3.5.27
vec2_norm CSU 2.14.3.5.31
vec2_print CSU 2.14.3.5.19
vec2_range_squared CSU 2.14.3.5.30
vec2_rot90 CSU 2.14.3.5.32
vec2_rot90minus CSU 2.14.3.5.33
vec2_rotate CSU 2.6.4.2.22
vec2_scale CSU 2.14.3.5.28
vec2_set CSU 2.14.3.5.21
vec2_sub CSU 2.14.3.5.23
vec2_veh2world CSU 2.6.3.2.37
vec2_world2veh CSU 2.6.3.2.38
vector_2d.c CSC 2.12.1.9
vector_is_first CSU 2.12.1.11.6
vector_z_rotate CSU 2.14.3.5.12
vec_distance_2d CSU 2.12.1.9.4
vec_normalize_2d CSU 2.12.1.9.2
vec_rotate_2d CSU 2.12.1.9.3
vec_set CSU 2.6.3.2.36
vehicle.c CSC 2.9.2.2
vehicle_bong CSU 2.3.1.5
vehicle_bong_do CSU 2.5.2.1.26
vehicle_catastrophe CSU 2.5.2.1.21
vehicle_component_name CSU 2.6.8.6.8
vehicle_deactivate CSU 2.9.2.2.8
vehicle_defuel CSU 2.5.2.1.24
vehicle_facing_point CSU 2.6.4.2.17
vehicle_fake_resupply CSU 2.5.2.1.23
vehicle_fill_in_appearance_data CSU 2.9.2.2.6
vehicle_fill_in_position_data CSU 2.9.2.2.5
vehicle_impact_to_sbx CSU 2.4.3.2.25
vehicle_iterator_next CSU 2.9.3.1.2
vehicle_iterator_once_next CSU 2.9.3.1.3
vehicle_iterator_reset CSU 2.9.3.1.1
vehicle_kill CSU 2.3.1.2
vehicle_kill_remote CSU 2.3.1.3
vehicle_list_append CSU 2.8.2.1.18
vehicle_manager_count CSU 2.9.3.1.5

vehicle_manager_count_force CSU 2.9.3.1.6
vehicle_manager_count_hosts CSU 2.9.3.1.8
vehicle_manager_count_sites CSU 2.9.3.1.7
vehicle_manager_print CSU 2.9.3.1.4
vehicle_ping CSU 2.3.1.6
vehicle_ping_do CSU 2.5.2.1.25
vehicle_planar_distance_squared CSU 2.9.2.2.7
vehicle_reappeared CSU 2.8.2.1.27
vehicle_resupply CSU 2.5.2.1.22
vehicle_show CSU 2.9.2.2.4
vehicle_status_string CSU 2.6.1.1.10
vehicle_wall_name CSU 2.6.8.6.7
veh_storage.h CSU 2.9.1.2
vel2point CSU 2.6.3.2.8
version.c CSC 2.1.3.2
vis_code CSU 2.6.7.1.53
wait_until CSU 2.2.1.5.2
warn_about_point_space CSU 2.12.1.1.6
water_avoidance.c CSC 2.12.1.8
water_check CSU 2.12.1.4.4
water_check.c CSC 2.12.1.4
water_segments_thru CSU 2.12.1.4.11
water_thru CSU 2.12.1.4.9
water_utilities.c CSC 2.12.1.11
waypoint_vel CSU 2.6.4.2.11
Weapons CSC 2.6.9
weapons.c CSC 2.6.9.8
weapons.h CSU 2.6.9.9
weapon_configure CSU 2.6.9.8.4
weapon_deselect CSU 2.6.9.8.15
weapon_load CSU 2.6.9.8.16
weapon_priority_list_show CSU 2.6.9.8.7
weapon_select CSU 2.6.9.8.14
weapon_show CSU 2.6.9.8.6
weapon_state_to_string CSU 2.6.9.8.1
weapon_systems_checkpoint CSU 2.6.9.8.9
weapon_systems_rearm CSU 2.6.9.8.8
weapon_systems_reinit CSU 2.6.9.8.10
weapon_systems_show CSU 2.6.9.8.5
weapon_unload CSU 2.6.9.8.17
which_event_ring CSU 2.2.1.3.6
which_side CSU 2.14.3.5.9
within_delta CSU 2.14.1.2.10

write_buffer_all_sbx CSU 2.4.3.2.3
xy_dir_and_range CSU 2.6.4.2.19
yyerror CSU 2.1.1.1.6
zero_perf_stats CSU 2.2.1.4.4
z_velocity CSU 2.6.4.2.13
_queue_hanger_hang_this CSU 2.14.4.2.3
_queue_hanger_unhang CSU 2.14.4.2.4