

**AD-A244 972**



1

**SOFTWARE DESIGN DOCUMENT  
SAF Workstation CSCI (6)**

Volume 1 of 2 Sections 1.0 - 2.4.3.4.86

June, 1991

**DTIC**  
**S FLECTE D**  
JAN 09 1992  
**D**

**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

**92-00247**



92 00247

**APPROVED FOR PUBLIC RELEASE  
DISTRIBUTION UNLIMITED**

**SOFTWARE DESIGN DOCUMENT  
SAF Workstation CSCI (6)**

Volume 1 of 2 Sections 1.0 - 2.4.3.4.86

**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

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

**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 June 1991	3. REPORT TYPE AND DATES COVERED Software Design Document	
4. TITLE AND SUBTITLE Software Design Document SAF Workstation CSCI (6)			5. FUNDING NUMBERS  Contract Numbers: MDA972-89-C-0060 MDA972-89-C-0061	
6. AUTHOR(S) Author not specified.			8. PERFORMING ORGANIZATION REPORT NUMBER  Advanced Simulation #: 9109	
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			10. SPONSORING/MONITORING AGENCY REPORT NUMBER DARPA Report Number: None.	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency (DARPA) 3701 North Fairfax Drive Arlington, VA 22203-1714				
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) Workstation Computer Software Configuration Item (CSCI number 6) 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 Workstation CSCI (CSCI 6).			15. NUMBER OF PAGES	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified		20. LIMITATION OF ABSTRACT Same as report.

## Table of Contents

1	INTRODUCTION: SAF WORKSTATION CSCI.....	1
1.1	BACKGROUND.....	1
1.2	EXTERNAL INTERFACES .....	1
1.3	INTERNAL STRUCTURE.....	2
1.4	CONFIGURATION AND CONFIGURATION MANAGEMENT.....	3
1.5	TERMINOLOGY AND DOCUMENTATION .....	3
1.5.1	Glossary.....	3
1.5.2	Related Documentation.....	4
1.6	MISCELLANEOUS .....	5
1.6.1	Automatically Generated Definition Cross-Reference.....	5
1.6.2	Auxiliary Functions.....	6
1.6.3	Commented-Out Code.....	7
2	CSCI FUNCTIONS AND INTERFACES .....	8
2.1	USER PROCESS CSC.....	8
2.1.1	CSU ui>processes.lisp.....	8
2.1.1.1	DYING-PROCESS .....	9
2.1.1.2	MAKE-OPFOR-SUB-PROCESS-FUNCTION-1.....	9
2.1.1.3	MAKE-OPFOR-SUB-PROCESS-FUNCTION .....	9
2.1.1.4	COM-CHECK-OPFOR-PROCESSES.....	10
2.1.1.5	COM-SAF-CHECK-OPFOR-PROCESSES .....	10
2.1.1.6	OPFOR-SUB-PROCESS-REPORTS .....	10
2.1.1.7	SECONDS-AGO.....	10
2.1.1.8	OPFOR-SUB-PROCESS.....	11
2.1.1.9	'(OPFOR-SUB-PROCESS PROCESS) .....	11
2.1.1.10	*ALL-OPFOR-SUB-PROCESSES* .....	11
2.1.1.11	(METHOD MAKE-INSTANCE OPFOR-SUB- PROCESS AFTER).....	11
2.1.1.12	(METHOD DISABLE OPFOR-SUB-PROCESS).....	12
2.1.1.13	(METHOD ENABLE OPFOR-SUB-PROCESS).....	12
2.1.1.14	(METHOD MURDER OPFOR-SUB-PROCESS).....	12
2.1.1.15	(METHOD REMEMBER OPFOR-SUB- PROCESS).....	12
2.1.1.16	(METHOD REPORT OPFOR-SUB-PROCESS).....	13
2.1.1.17	*RUDP-PROCESS-LAST-CYCLE*.....	13



	2.1.1.18	NETWORK-PROCESS-WAKE-UP .....	13
	2.1.1.19	PROCESS-RUDP-PACKETS .....	14
	2.1.1.20	MAKE-RUDP-PROCESS .....	14
	2.1.1.21	MAKE-UPDATE-PROCESS.....	14
2.1.2	CSU ui>menus.lisp.....		15
	2.1.2.1	(METHOD MULTIPLE-CHOICE-ALL-SHOW MULTIPLE-CHOICE-MIXIN) .....	15
	2.1.2.2	(METHOD MULTIPLE-CHOICE-ALL-HIDE MULTIPLE-CHOICE-MIXIN) .....	15
	2.1.2.3	*TERRAIN-MENU* .....	15
	2.1.2.4	MAYBE-MAKE-TERRAIN-MENU .....	16
	2.1.2.5	HANDLE-TERRAIN-MENU .....	16
2.1.3	CSU ui>mouse-interface.lisp.....		16
	2.1.3.1	('*NEW-INTERFACE-FLG* CONSIDER- FLIPPING).....	17
	2.1.3.2	*NEW-INTERFACE-FLG*.....	17
	2.1.3.3	MOUSE-FLIP-SCREEN.....	17
	2.1.3.4	FIND-MOUSE .....	18
	2.1.3.5	CONSIDER-FLIPPING .....	18
	2.1.3.6	JUMP-TO-B&W-SCREEN.....	19
	2.1.3.7	JUMP-TO-COLOR-SCREEN.....	19
	2.1.3.8	*COLOR-SCREEN-MENU*.....	19
	2.1.3.9	COLOR-SCREEN-MENU.....	19
	2.1.3.10	CLEAR-UNITS .....	20
	2.1.3.11	CLEAR-UNITS-AND-OVERLAYS.....	20
	2.1.3.12	CLEAR-OVERLAYS.....	21
	2.1.3.13	*NEW-INTERFACE-PROCESS*.....	21
	2.1.3.14	*NIP-FORMS*.....	21
	2.1.3.15	PUSH-NIP-FORM-IF-NECESSARY .....	22
	2.1.3.16	EXECUTE-IN-NEW-INTERFACE.....	22
	2.1.3.17	NEW-INTERFACE-PROCESS-FUNCTION .....	22
2.1.4	CSU ui>frame.lisp.....		22
	2.1.4.1	STANDARD-MARGINS .....	23
	2.1.4.2	PVD.....	23
	2.1.4.3	PVD.....	24
	2.1.4.4	DO-NOTHING-COMMAND-LOOP .....	25
	2.1.4.5	MAKE-PVD-FRAME .....	25

2.1.4.6	SET-UP-PVD-SCALE .....	25
2.1.4.7	EXPOSE-PVD.....	26
2.1.4.8	Expose PVD .....	26
2.1.4.9	CLEAR-SAF-HISTORY.....	27
2.1.4.10	Clear SAF History .....	27
2.1.4.11	(SET-HIGHLIGHTED-PRESENTATION MAP- WINDOW).....	27
2.1.4.12	(WHO-LINE-DOCUMENTATION-STRING MAP-WINDOW).....	27
2.1.4.13	SAF.....	28
2.1.4.14	(METHOD MAKE-INSTANCE SAF AFTER).....	29
2.1.4.15	(METHOD TOP-LEVEL SAF).....	30
2.1.4.16	(METHOD GET-RUDP-PROCESS PROGRAM- FRAME).....	31
2.1.4.17	(METHOD GET-UPDATE-PROCESS PROGRAM-FRAME).....	31
2.1.5	CSU ui>commands.lisp.....	31
2.1.5.1	DEFINE-PVD-MENU-COMMAND .....	31
2.1.5.2	(COM-ZOOM-IN Zoom In ).....	32
2.1.5.3	(COM-PAN Pan ) .....	32
2.1.5.4	GET-ELEVATION.....	32
2.1.5.5	(COM-ZOOM-OUT Zoom Out ) .....	33
2.1.5.6	RESCALE-PVD-FROM-MENU.....	33
2.1.5.7	(COM-RESCALE Map Scale ) .....	33
2.1.5.8	(COM-REFRESH Refresh ) .....	34
2.1.5.9	(COM-TERRAIN-OPTIONS Terrain Options ).....	34
2.1.5.10	PAN-TO-POINT .....	34
2.1.5.11	PARSE-COORDS.....	34
2.1.5.12	COM-PAN-TO-POINT.....	35
2.1.5.13	COM-UNIT-OPS .....	35
2.1.5.14	COM-BATTALION-OPS .....	35
2.1.5.15	COM-REFRESH-UNIT-DISPLAY .....	35
2.1.5.16	COM-CLEAR-MESSAGE-LOG.....	36
2.1.5.17	COM-CLEAR.....	36
2.1.5.18	COM-SET-VIEWPORT.....	36
2.1.5.19	COM-BOMB-BUTTON.....	37
2.1.5.20	COM-SAF-SET-BOMB-PARAMETERS.....	37

2.1.5.21	COM-ROBO-COP-CONTROL.....	37
2.1.5.22	COM-SET-OPFOR-PARAMETERS .....	37
2.1.5.23	COM-SAVE-SCENARIO.....	38
2.1.5.24	COM-DELETE-SCENARIOS.....	38
2.1.5.25	COM-DELETE-EXERCISES .....	38
2.1.5.26	COM-DELETE-OVERLAYS .....	38
2.1.5.27	COM-STORE-SCENARIO .....	39
2.1.5.28	SUPERIOR-CONTEXT .....	39
2.1.5.29	ROOT-INPUT-CONTEXT .....	39
2.1.5.30	PVD-COMMAND-MENU.....	39
2.2	COMMANDER CSC.....	40
2.2.1	Task Organization CSC .....	40
2.2.1.1	CSU ui>task-org.lisp.....	41
2.2.1.1.1	WORKSTATION-BATTALION.....	41
2.2.1.1.2	MOUSE-WORKSTATION- BATTALION .....	41
2.2.1.1.3	RUN-BATTALION-OPS .....	41
2.2.1.1.4	DISPLAY-WORKSTATION- BATTALION .....	42
2.2.1.1.5	DISPLAY-FOR-TASK-ORG.....	42
2.2.1.1.6	INFERIORS-FOR-TASK-ORG .....	42
2.2.1.1.7	HIGHLIGHT-ON-TASK-ORG.....	43
2.2.1.1.8	TASK-ORG-PANE.....	43
2.2.1.1.9	(SET-HIGHLIGHTED- PRESENTATION TASK-ORG- PANE).....	43
2.2.1.1.10	(METHOD SET- HIGHLIGHTED- PRESENTATION TASK-ORG- PANE AFTER).....	44
2.2.1.1.11	DISPLAY-TASK-ORG .....	44
2.2.1.1.12	(METHOD DRAW-TASK- ORGANIZATION TASK-ORG- PANE).....	44
2.2.1.1.13	(METHOD DRAW-TASK- ORGANIZATION TASK-ORG- PANE AFTER).....	45
2.2.1.1.14	TASK-ORG-PANE.....	45
2.2.2	Operations Order (OPORD) CSC .....	45
2.2.2.1	CSU ui>opord.lisp.....	46

2.2.2.1.1	*OPORD-MODE*.....	46
2.2.2.1.2	*ENABLED-FONT* .....	46
2.2.2.1.3	*DISABLED-FONT*.....	46
2.2.2.1.4	*PREVIOUS-BUTTON-BOX*.....	47
2.2.2.1.5	OPORD-BUTTON.....	47
2.2.2.1.6	(METHOD FONT OPORD- BUTTON).....	47
2.2.2.1.7	(METHOD HIGHLIGHT OPORD-BUTTON).....	47
2.2.2.1.8	OPORD-BUTTON.....	48
2.2.2.1.9	PARAGRAPH.....	48
2.2.2.1.10	MAKE-PARAGRAPH.....	48
2.2.2.1.11	(METHOD DISPLAY PARAGRAPH).....	48
2.2.2.1.12	SUBPARAGRAPH.....	49
2.2.2.1.13	MAKE-SUBPARAGRAPH.....	49
2.2.2.1.14	(METHOD DISPLAY SUBPARAGRAPH).....	49
2.2.2.1.15	PARAGRAPH.....	49
2.2.2.1.16	SUBPARAGRAPH.....	50
2.2.2.1.17	*PARAGRAPH-DATA*.....	50
2.2.2.1.18	OPS-BUTTON.....	50
2.2.2.1.19	MAKE-OPS-BUTTON.....	51
2.2.2.1.20	(METHOD DISPLAY OPS- BUTTON).....	51
2.2.2.1.21	OPS-BUTTON.....	51
2.2.2.1.22	OPS-BUTTON.....	52
2.2.2.1.23	*OPERATIONS-BUTTONS*.....	52
2.2.2.1.24	(METHOD DISPLAY- PARAGRAPHS SAF).....	52
2.2.2.1.25	(METHOD DISPLAY- OPERATIONS SAF).....	52
2.2.2.1.26	(METHOD DISPLAY-OPORD- CHOICES SAF).....	53
2.2.2.1.27	(COM-SELECT- SUBPARAGRAPH).....	53
2.2.2.1.28	SELECT-SUBPARAGRAPH.....	53
2.2.2.1.29	(COM-SELECT-BUTTON).....	53
2.2.2.1.30	SELECT-OPS-BUTTON.....	54

2.2.2.1.31	OPORD.....	54
2.2.2.2	CSU objects>intervention.lisp.....	54
2.2.2.2.1	INTERVENE.....	55
2.2.2.2.2	(METHOD INTERVENE SIMNET-AGENT OTHERWISE).....	55
2.2.2.2.3	(METHOD INTERVENE SIMNET-AGENT RULES-OF- ENGAGEMENT).....	55
2.2.2.2.4	(METHOD INTERVENE SIMNET-AGENT FACE- DIRECTION).....	55
2.2.2.2.5	(METHOD INTERVENE SIMNET-AGENT HALT).....	56
2.2.2.2.6	(METHOD INTERVENE SIMNET-AGENT HOLD).....	56
2.2.2.2.7	(METHOD INTERVENE SIMNET-AGENT ENROUTE- MOVEMENT).....	56
2.2.2.2.8	(METHOD INTERVENE SIMNET-AGENT SPEED).....	57
2.2.2.2.9	(METHOD INTERVENE SIMNET-AGENT ALTITUDE).....	57
2.2.2.2.10	(METHOD INTERVENE SIMNET-AGENT FOLLOW- VEHICLE).....	57
2.2.2.2.11	(METHOD INTERVENE SIMNET-AGENT COMMAND- FROM-SIMULATOR).....	58
2.2.2.2.12	(METHOD INTERVENE SIMNET-AGENT GO-TO LOCATION).....	58
2.2.2.2.13	(METHOD INTERVENE SIMNET-AGENT RESUPPLY).....	59
2.2.2.2.14	(METHOD INTERVENE SIMNET-AGENT LAND).....	59
2.2.2.2.15	(METHOD INTERVENE SIMNET-AGENT ATTACK).....	60
2.2.2.2.16	(METHOD INTERVENE SIMNET-AGENT RESUME).....	60
2.2.2.2.17	(METHOD INTERVENE SIMNET-AGENT RESUME- ALL-SUBORDINATES).....	61

	2.2.2.2.18	(METHOD INTERVENE SIMNET-AGENT REJOIN- UNIT) .....	61
	2.2.2.2.19	(METHOD INTERVENE SIMNET-AGENT FORMATION) .....	61
2.2.2.3	CSU ui>subordinate-tasking.lisp .....		62
	2.2.2.3.1	*TOP-LEVEL-TASKING* .....	62
	2.2.2.3.2	OVERLAY-IS-MODIFIED .....	62
	2.2.2.3.3	(COMPILE LOAD EVAL) .....	63
	2.2.2.3.4	(METHOD SET- HIGHLIGHTED- PRESENTATION SUB-TASK- PANE AFTER) .....	63
	2.2.2.3.5	*FRAG-ORDER-COUNT* .....	63
	2.2.2.3.6	COUNT-FRAGO .....	64
	2.2.2.3.7	FRAGO-COUNT .....	64
	2.2.2.3.8	RESET-FRAGO-COUNT .....	64
	2.2.2.3.9	PRINT-FRAGO-COUNT .....	64
	2.2.2.3.10	SUBORDINATE-UNIT- TASKING .....	65
	2.2.2.3.11	(METHOD CLEAR-STATE SUBORDINATE-UNIT- TASKING) .....	65
	2.2.2.3.12	(METHOD SAVE-SCROLL- STATE SUBORDINATE-UNIT- TASKING) .....	65
	2.2.2.3.13	(METHOD DISPLAY-TASKING- TABLE SUBORDINATE-UNIT- TASKING) .....	66
	2.2.2.3.14	(METHOD DISPLAY-TITLE SUBORDINATE-UNIT- TASKING) .....	66
	2.2.2.3.15	UNIT-TASK-OVERLAY .....	66
	2.2.2.3.16	UNIT-TASK-UNIT .....	67
	2.2.2.3.17	COMBAT-INSTRUCTION-SET .....	67
	2.2.2.3.18	UNIT-TASK .....	68
	2.2.2.3.19	(METHOD MAKE-INSTANCE UNIT-TASK AFTER) .....	68
	2.2.2.3.20	SUB-TASK .....	69
	2.2.2.3.21	(METHOD MAKE-INSTANCE SUB-TASK AFTER) .....	69

2.2.2.3.22	(METHOD CIS-NAME SUB-TASK).....	69
2.2.2.3.23	(METHOD EXECUTE-SUB-TASK SUB-TASK) .....	70
2.2.2.3.24	(METHOD REEXECUTE-SUB-TASK SUB-TASK) .....	70
2.2.2.3.25	MAKE-UNIT-LIST.....	70
2.2.2.3.26	BUILD-UNIT-TASKING-STRUCTURE .....	71
2.2.2.3.27	MERGE-UNIT-TASKING.....	71
2.2.2.3.28	(METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK).....	72
2.2.2.3.29	(COM-CANCEL MENU-ACCELERATOR T) .....	72
2.2.2.3.30	(COM-DONE MENU-ACCELERATOR T) .....	72
2.2.2.3.31	(COM-WARN-OVERLAY MENU-ACCELERATOR T) .....	73
2.2.2.3.32	(COM-EXECUTE-OVERLAY MENU-ACCELERATOR T) .....	73
2.2.2.3.33	(COM-ISSUE-FRAG-ORDER MENU-ACCELERATOR T) .....	73
2.2.2.3.34	(COM-CHOOSE-OVERLAY).....	73
2.2.2.3.35	SELECT-OVERLAY .....	74
2.2.2.3.36	(COM-CHANGE-SUB-TASK).....	74
2.2.2.3.37	SELECT-SUB-TASK.....	74
2.2.2.3.38	(METHOD DISPLAY-SUB-TASKING SUB-TASK).....	74
2.2.2.3.39	(METHOD DISPLAY-OVERLAY-TASKING UNIT-TASK).....	75
2.2.2.3.40	DISPLAY-SUBORDINATE-TASKING-TABLE .....	75
2.2.2.3.41	SUBORDINATE-TASK.....	75
2.2.2.3.42	CLEAR-TOP-LEVEL-TASKING .....	76
2.2.2.3.43	RESET-ALL-OVERLAYS-AND-TASKS .....	76
2.2.2.3.44	SUBORDINATE-UNIT-TASKING .....	76
2.2.2.4	CSU cm>overlay.lisp.....	77
2.2.2.4.1	OVERLAY .....	77

2.2.2.4.2	OVERLAY?.....	78
2.2.2.4.3	(METHOD MAKE-INSTANCE OVERLAY AFTER).....	78
2.2.2.4.4	(METHOD KILL OVERLAY).....	79
2.2.2.4.5	(METHOD PRINT-SELF OVERLAY).....	79
2.2.2.4.6	(METHOD REVIEW-DATA OVERLAY).....	79
2.2.2.4.7	(METHOD REFRESH OVERLAY).....	79
2.2.2.4.8	(METHOD DRAW OVERLAY).....	80
2.2.2.4.9	(METHOD ERASE OVERLAY).....	80
2.2.2.4.10	(METHOD ADD-NEW- CONTROL-MEASURE OVERLAY).....	80
2.2.2.4.11	(METHOD ADD-CONTROL- MEASURE OVERLAY).....	81
2.2.2.4.12	(METHOD DELETE-CONTROL- MEASURE OVERLAY).....	81
2.2.2.4.13	(METHOD DELETE-ALL- CONTROL-MEASURES OVERLAY).....	81
2.2.2.4.14	*CM-DELETE-MENU*.....	81
2.2.2.4.15	*CM-DELETE-MENU-COLOR*.....	82
2.2.2.4.16	GET-DELETE-CM-MENU.....	82
2.2.2.4.17	MULTIPLE MENU-DELETE- CMS.....	82
2.2.2.4.18	(METHOD DELETE-SOME- CONTROL-MEASURES OVERLAY).....	82
2.2.2.4.19	(METHOD SEND-OVERLAY- TO-SIMHOST OVERLAY).....	83
2.2.2.4.20	(METHOD CM-NEEDS- UPDATING OVERLAY).....	83
2.2.2.4.21	(METHOD ALL-ROUTES OVERLAY).....	83
2.2.2.4.22	(METHOD OVERLAY-OPS OVERLAY).....	84
2.2.2.4.23	OVERLAY.....	84
2.2.2.4.24	MAKE-OVERLAY.....	85
2.2.2.4.25	REDRAW-OVERLAYS.....	85
2.2.2.4.26	CHOOSE-AN-OVERLAY.....	86



	2.2.2.4.27	SORT-CMS.....	86
2.2.3		Control Measures CSC.....	86
	2.2.3.1	CSU cm>control-measure.lisp .....	87
	2.2.3.1.1	*CONTROL-MEASURE-ID* .....	87
	2.2.3.1.2	UNIQUE-CM-ID.....	88
	2.2.3.1.3	'CONTROL-MEASURE.....	88
	2.2.3.1.4	CONTROL-MEASURE.....	88
	2.2.3.1.5	(METHOD MAKE-INSTANCE CONTROL-MEASURE AFTER).....	89
	2.2.3.1.6	(METHOD PRINT-SELF CONTROL-MEASURE).....	90
	2.2.3.1.7	(METHOD REFRESH CONTROL-MEASURE).....	90
	2.2.3.1.8	(METHOD ROUTEP CONTROL- MEASURE).....	90
	2.2.3.1.9	(DRAW CONTROL-MEASURE) .....	90
	2.2.3.1.10	(ERASE CONTROL-MEASURE).....	90
	2.2.3.1.11	(METHOD DRAW-NAME CONTROL-MEASURE).....	91
	2.2.3.1.12	(METHOD ERASE-NAME CONTROL-MEASURE).....	91
	2.2.3.1.13	(REVIEW-DATA CONTROL- MEASURE).....	91
	2.2.3.1.14	(MOVE-POINT CONTROL- MEASURE).....	92
	2.2.3.1.15	(DELETE-POINT CONTROL- MEASURE).....	92
	2.2.3.1.16	(INSERT-POINT-AFTER CONTROL-MEASURE).....	92
	2.2.3.1.17	(INSERT-POINT-BEFORE CONTROL-MEASURE).....	92
	2.2.3.1.18	(METHOD ADD-CM-TO- OVERLAY CONTROL- MEASURE).....	92
	2.2.3.1.19	CONTROL-MEASURE- BEHAVIOR.....	93
	2.2.3.1.20	(METHOD PRINT-SELF CONTROL-MEASURE- BEHAVIOR).....	93
	2.2.3.1.21	CONTROL-MEASURE.....	94
	2.2.3.1.22	REVERSE-XY .....	95

	2.2.3.1.23	UNIT .....	96
	2.2.3.1.24	LOCAL-UNIT.....	97
	2.2.3.1.25	*PREV-UNITS* .....	97
	2.2.3.1.26	*APPLIES-TO-UNIT-MENU* .....	97
	2.2.3.1.27	MAKE-APPLIES-TO-UNIT-MENU.....	98
	2.2.3.1.28	MULTIPLE-MENU-CHOOSE-UNITS.....	98
	2.2.3.1.29	CHOOSE-UNITS-FOR-CM.....	98
	2.2.3.1.30	CM-UNIT.....	99
	2.2.3.1.31	REMOVE-UNIT-POINTERS-IN-BEHAVIORS.....	99
	2.2.3.1.32	FORMATION .....	100
	2.2.3.1.33	CM-FORMATION.....	101
	2.2.3.1.34	CIS-FOR-CM.....	101
	2.2.3.1.35	CM-CIS .....	101
	2.2.3.1.36	CM-SPEED .....	102
	2.2.3.1.37	WORLD-COORDS .....	102
	2.2.3.1.38	CONTROL-MEASURE-LABEL.....	102
	2.2.3.1.39	CONTROL-MEASURE-LABEL-GESTURE .....	103
2.2.3.2		CSU cm>control-measure-point.lisp .....	103
	2.2.3.2.1	CONTROL-MEASURE-POINT .....	103
	2.2.3.2.2	(METHOD MAKE-INSTANCE CONTROL-MEASURE-POINT AFTER) .....	104
	2.2.3.2.3	(METHOD PAINT CONTROL-MEASURE-POINT).....	104
	2.2.3.2.4	(METHOD DRAW-AS-FIRST-POINT CONTROL-MEASURE-POINT).....	104
	2.2.3.2.5	(METHOD DRAW CONTROL-MEASURE-POINT).....	105
	2.2.3.2.6	(METHOD ERASE CONTROL-MEASURE-POINT).....	105
	2.2.3.2.7	(METHOD COPY CONTROL-MEASURE-POINT).....	105
	2.2.3.2.8	CONTROL-MEASURE-POINT .....	106
	2.2.3.2.9	CONTROL-MEASURE-POINT .....	106

	2.2.3.2.10	CONTROL-MEASURE- GESTURE .....	106
	2.2.3.2.11	XY-LIST-TO-POINTS.....	107
2.2.3.3		CSU cm>point.lisp .....	107
	2.2.3.3.1	CM-POINT .....	107
	2.2.3.3.2	CM-POINT-BEHAVIOR .....	108
	2.2.3.3.3	(METHOD SEND-BEH-INFO CM-POINT-BEHAVIOR) .....	108
	2.2.3.3.4	(METHOD COPY-BEHAVIOR CM-POINT-BEHAVIOR) .....	109
	2.2.3.3.5	(METHOD MAKE-BEHAVIOR CM-POINT) .....	109
	2.2.3.3.6	(METHOD REVIEW-DATA CM- POINT).....	109
	2.2.3.3.7	(METHOD DRAW CM-POINT).....	110
	2.2.3.3.8	(METHOD ERASE CM-POINT) .....	110
	2.2.3.3.9	(METHOD MOVE-POINT CM- POINT).....	111
	2.2.3.3.10	(METHOD DELETE-POINT CM- POINT).....	111
	2.2.3.3.11	(METHOD SEND-CM-INFO CM- POINT).....	111
	2.2.3.3.12	(METHOD COPY CM-POINT).....	111
	2.2.3.3.13	(METHOD CM-INTERSECTION CM-POINT) .....	112
	2.2.3.3.14	CM-POINT .....	112
	2.2.3.3.15	CM-POINT .....	113
	2.2.3.3.16	CM-POINT-GESTURE.....	113
	2.2.3.3.17	MAKE-POINT .....	114
2.2.3.4		CSU cm>line.lisp .....	114
	2.2.3.4.1	LINE.....	114
	2.2.3.4.2	LINE-BEHAVIOR.....	115
	2.2.3.4.3	(METHOD SEND-BEH-INFO LINE-BEHAVIOR).....	115
	2.2.3.4.4	(METHOD COPY-BEHAVIOR LINE-BEHAVIOR).....	115
	2.2.3.4.5	(METHOD MAKE-BEHAVIOR LINE).....	116
	2.2.3.4.6	(METHOD MAKE-INSTANCE LINE AFTER).....	116

2.2.3.4.7	(METHOD INITIALIZE-POINTS LINE).....	116
2.2.3.4.8	(METHOD REVIEW-DATA LINE).....	116
2.2.3.4.9	(METHOD PAINT-NAME LINE)....	117
2.2.3.4.10	(DRAW-SEGMENT LINE) .....	117
2.2.3.4.11	(METHOD PAINT LINE) .....	117
2.2.3.4.12	(METHOD DRAW LINE) .....	118
2.2.3.4.13	(METHOD ERASE LINE).....	118
2.2.3.4.14	(METHOD ORTHOGONALIZE LINE).....	118
2.2.3.4.15	(METHOD MOVE-POINT LINE) ....	119
2.2.3.4.16	(METHOD DELETE-POINT LINE).....	119
2.2.3.4.17	(METHOD INSERT-POINT- AFTER LINE).....	120
2.2.3.4.18	(METHOD INSERT-POINT- BEFORE LINE).....	120
2.2.3.4.19	(METHOD SEND-CM-INFO LINE).....	121
2.2.3.4.20	(METHOD MOVE-CONTROL- MEASURE LINE) .....	121
2.2.3.4.21	(METHOD COPY LINE) .....	121
2.2.3.4.22	(METHOD CM-INTERSECTION LINE).....	122
2.2.3.4.23	LINE.....	122
2.2.3.4.24	MAKE-LINE .....	123
2.2.3.5	CSU cm>generic-area.lisp .....	123
2.2.3.5.1	GENERIC-AREA .....	123
2.2.3.5.2	GENERIC-AREA?.....	124
2.2.3.5.3	(METHOD MAKE-INSTANCE GENERIC-AREA AFTER).....	124
2.2.3.5.4	(METHOD INITIALIZE-POINTS GENERIC-AREA).....	124
2.2.3.5.5	(METHOD PAINT-NAME GENERIC-AREA).....	124
2.2.3.5.6	(METHOD PAINT GENERIC- AREA).....	125
2.2.3.5.7	(METHOD DRAW GENERIC- AREA).....	125

	2.2.3.5.8	(METHOD ERASE GENERIC-AREA).....	125
	2.2.3.5.9	(METHOD MOVE-POINT GENERIC-AREA).....	126
	2.2.3.5.10	(METHOD DELETE-POINT GENERIC-AREA).....	126
	2.2.3.5.11	(METHOD INSERT-POINT-AFTER GENERIC-AREA).....	127
	2.2.3.5.12	(METHOD INSERT-POINT-BEFORE GENERIC-AREA).....	127
	2.2.3.5.13	(METHOD ORTHOGONALIZE GENERIC-AREA).....	128
	2.2.3.5.14	(METHOD SEND-CM-INFO GENERIC-AREA).....	128
	2.2.3.5.15	GENERIC-AREA.....	128
2.2.3.6	CSU cm>area.lisp.....		128
	2.2.3.6.1	AREA.....	129
	2.2.3.6.2	AREA-BEHAVIOR.....	129
	2.2.3.6.3	(METHOD MAKE-BEHAVIOR AREA).....	130
	2.2.3.6.4	(METHOD COPY-BEHAVIOR AREA-BEHAVIOR).....	130
	2.2.3.6.5	(METHOD REVIEW-DATA AREA).....	130
	2.2.3.6.6	(METHOD COPY AREA).....	130
	2.2.3.6.7	(METHOD MOVE-CONTROL-MEASURE AREA).....	131
	2.2.3.6.8	(METHOD CM-INTERSECTION AREA).....	131
	2.2.3.6.9	AREA.....	132
	2.2.3.6.10	MAKE-AREA.....	132
2.2.3.7	CSU cm>zone.lisp.....		133
	2.2.3.7.1	ZONE.....	133
	2.2.3.7.2	ZONE-BEHAVIOR.....	133
	2.2.3.7.3	(METHOD COPY-BEHAVIOR ZONE-BEHAVIOR).....	134
	2.2.3.7.4	(METHOD MAKE-BEHAVIOR ZONE).....	134
	2.2.3.7.5	(METHOD REVIEW-DATA ZONE).....	134
	2.2.3.7.6	(METHOD COPY ZONE).....	134

		2.2.3.7.7	(METHOD MOVE-CONTROL- MEASURE ZONE).....	135
		2.2.3.7.8	(METHOD CM-INTERSECTION ZONE).....	135
		2.2.3.7.9	ZONE.....	136
		2.2.3.7.10	MAKE-ZONE .....	136
2.2.4	Routes	CSC.....		137
	2.2.4.1	CSU cm>water-avoidance.lisp.....		137
		2.2.4.1.1	*INTERSECTIONS- SEARCHED*.....	137
		2.2.4.1.2	*QUADS-INDEX-LIST*.....	137
		2.2.4.1.3	FIND-ROUTE-AROUND- WATER .....	138
		2.2.4.1.4	THRU-RIVER-BEND.....	138
		2.2.4.1.5	FIND-ROUTE-CORE .....	139
		2.2.4.1.6	FOLLOW-WATER-SEGMENTS.....	139
		2.2.4.1.7	FIND-WATER- INTERSECTIONS.....	140
		2.2.4.1.8	GET-PAIRS-BY-DIRECTION.....	140
		2.2.4.1.9	FIND-SUITABLE-CROSSING- ROUTE.....	141
		2.2.4.1.10	SET-XOR .....	141
		2.2.4.1.11	EXTEND-CROSSING.....	141
		2.2.4.1.12	EXTEND-INTERSECTION.....	142
		2.2.4.1.13	FIRST-ITEMS.....	142
		2.2.4.1.14	EXTEND-BRIDGE.....	143
		2.2.4.1.15	EXTEND-SEGMENT .....	143
		2.2.4.1.16	INTERSECTION-DIRECTION.....	144
		2.2.4.1.17	NORMALIZE-AND-ROTATE .....	144
		2.2.4.1.18	FIND-FIRST-VECTOR.....	144
		2.2.4.1.19	VECTOR-IS-FIRST-P .....	145
		2.2.4.1.20	FIND-NEXT-POINT .....	145
		2.2.4.1.21	FIND-SEGMENT-CROSS- POINTS .....	145
		2.2.4.1.22	FIND-CLOSER-CROSSING .....	146
		2.2.4.1.23	SKIRT-RIVER.....	146
		2.2.4.1.24	FIND-RIVER-POINTS .....	147
		2.2.4.1.25	ALIGN-POINTS.....	147

	2.2.4.1.26	OFFSET-POINTS .....	147
	2.2.4.1.27	OFFSET-POINT .....	148
	2.2.4.1.28	PRUNE-TO-POINT .....	148
	2.2.4.1.29	CROSSING-LOCATION .....	149
	2.2.4.1.30	RELAX-POINTS .....	149
	2.2.4.1.31	RELAX-POINTS-AUX.....	150
	2.2.4.1.32	FINAL-RELAX-POINTS.....	150
	2.2.4.1.33	FLAT-LIST-TO-POINTS.....	151
	2.2.4.1.34	SKIRT-LAKE .....	151
	2.2.4.1.35	DISTANCE-AROUND-PATH.....	151
	2.2.4.1.36	FOLLOW-LAKE-AROUND .....	152
	2.2.4.1.37	SKIRT-RIVER-BEND .....	152
	2.2.4.1.38	FIND-RIVER-BEND-POINTS .....	153
	2.2.4.1.39	FIND-DIRECTION-AT- CROSSING.....	153
	2.2.4.1.40	GET-QUADS-IN-REGION .....	153
2.2.4.2		CSU cm>water-check.lisp .....	154
	2.2.4.2.1	ANY-WIDE-SEGMENT-THRU- WATER .....	154
	2.2.4.2.2	SEGMENT-THRU-WATER .....	155
	2.2.4.2.3	SEGMENT-THRU-RIVER.....	155
	2.2.4.2.4	*INSIDE-LEVEL* .....	156
	2.2.4.2.5	SEGMENT-THRU-LAKE.....	156
	2.2.4.2.6	POLYGON-INTERSECTION .....	156
	2.2.4.2.7	CHECK-LAKE- INTERSECTIONS.....	157
	2.2.4.2.8	ALL-WIDE-SEGMENTS-THRU- WATER .....	157
	2.2.4.2.9	WATER-THRU .....	158
	2.2.4.2.10	WATER-SEGMENTS-THRU .....	158
	2.2.4.2.11	LAKES-THRU .....	158
	2.2.4.2.12	GET-QUADS-PASSED-THRU.....	159
2.2.4.3		CSU cm>route-point.lisp .....	159
	2.2.4.3.1	ROUTE-POINT .....	160
	2.2.4.3.2	(METHOD COPY ROUTE- POINT).....	160
	2.2.4.3.3	ROUTE-POINT .....	160
	2.2.4.3.4	XY-LIST-TO-ROUTE-POINTS .....	160

2.2.4.4	CSU cm>road-routes.lisp.....	161
2.2.4.4.1	GET-ROAD-ROUTE.....	162
2.2.4.4.2	GET-ROAD-POINT.....	163
2.2.4.4.3	GET-ROAD-SEGMENT-POINT.....	163
2.2.4.4.4	FIND-NEAREST- INTERSECTION.....	164
2.2.4.4.5	FIND-NEAREST-ROAD- SEGMENT.....	164
2.2.4.4.6	GET-NEIGHBOR-QUAD- ROADS.....	165
2.2.4.4.7	CALCULATE-POINT-LINE- INTERSECTION.....	165
2.2.4.4.8	PARALLEL-DISTANCE.....	165
2.2.4.4.9	ROUTE-INTERSECTION.....	166
2.2.4.4.10	WITHIN-CURSOR.....	166
2.2.4.4.11	EXPAND-ROUTE.....	166
2.2.4.4.12	EXPAND-ROAD-ROUTE.....	167
2.2.4.4.13	FIND-ROAD-INTERSECTIONS.....	167
2.2.4.4.14	FIND-SHORTEST-ROUTE.....	168
2.2.4.4.15	CALCULATE-ROUTE- DISTANCE.....	168
2.2.4.4.16	FIND-ROAD-DIRECTION.....	168
2.2.4.4.17	ROAD-SEGMENTS-FROM- INTERSECTIONS.....	169
2.2.4.4.18	DRAW-EXPANDED-ROUTE.....	169
2.2.4.4.19	DRAW-EXPANDED-ROUTE- CORE.....	169
2.2.4.4.20	GET-BRIDGE-ROUTE.....	170
2.2.4.4.21	GET-BRIDGE-POINTS.....	171
2.2.4.4.22	MOUSE-ON-BRIDGE- APPROACH-POINT.....	171
2.2.4.4.23	FIND-NEAREST-BRIDGE.....	172
2.2.4.5	CSU cm>route-finder.lisp.....	172
2.2.4.5.1	FIND-ROUTE.....	172
2.2.4.5.2	FIND-ROUTE.....	172
2.2.4.5.3	EXPAND-FIRST-ROUTE.....	173
2.2.4.5.4	SORT-ROUTE-QUEUE.....	173
2.2.4.5.5	PARTIAL-SORT.....	173
2.2.4.5.6	FIND-SHORTEST.....	174



	2.2.4.5.7	TRIM-REDUNDANCY.....	174
	2.2.4.5.8	DISTANCE-BETWEEN- INTERSECTIONS.....	174
	2.2.4.5.9	EXPAND-ROUTE-INTO- POINTS .....	175
2.2.4.6		CSU cm>route.lisp .....	175
	2.2.4.6.1	*ASK-USER*.....	176
	2.2.4.6.2	ROUTE.....	176
	2.2.4.6.3	CM-ROUTE?.....	178
	2.2.4.6.4	ROUTE-BEHAVIOR.....	178
	2.2.4.6.5	(METHOD COPY-BEHAVIOR ROUTE-BEHAVIOR).....	178
	2.2.4.6.6	(METHOD MAKE-BEHAVIOR ROUTE).....	179
	2.2.4.6.7	(METHOD MAKE-INSTANCE ROUTE AFTER).....	179
	2.2.4.6.8	(METHOD INITIALIZE-POINTS ROUTE).....	179
	2.2.4.6.9	(METHOD REVIEW-DATA ROUTE).....	179
	2.2.4.6.10	(METHOD PAINT-NAME ROUTE).....	180
	2.2.4.6.11	(METHOD PAINT ROUTE) .....	180
	2.2.4.6.12	(METHOD DRAW ROUTE) .....	180
	2.2.4.6.13	(METHOD ERASE ROUTE).....	180
	2.2.4.6.14	(METHOD ORTHOGONALIZE ROUTE).....	181
	2.2.4.6.15	(METHOD MOVE-POINT ROUTE).....	181
	2.2.4.6.16	(METHOD DELETE-POINT ROUTE).....	182
	2.2.4.6.17	(METHOD INSERT-POINT- AFTER ROUTE).....	182
	2.2.4.6.18	(METHOD INSERT-POINT- BEFORE ROUTE).....	183
	2.2.4.6.19	(METHOD CHECK ROUTE).....	184
	2.2.4.6.20	(METHOD CHECK-ROUTE- SEGMENT ROUTE).....	184
	2.2.4.6.21	(METHOD SEND-CM-INFO ROUTE).....	185
	2.2.4.6.22	(METHOD COPY ROUTE) .....	185

	2.2.4.6.23	ROUTE.....	185
	2.2.4.6.24	MAKE-ROUTE .....	187
2.3	BATTLEMASTER CSC .....		188
2.3.1	Battlemaster Interface (BMI) CSC.....		188
2.3.1.1	CSU bmi>bmi-frame.lisp .....		189
2.3.1.1.1	BMI .....		189
2.3.1.1.2	(METHOD ENABLE-MMSHIP- CHANGE BMI).....		189
2.3.1.1.3	(METHOD SET-ENABLE- MMSHIP-CHANGE BMI).....		189
2.3.1.1.4	WORKSTATION-MMSHIP- CHANGEP .....		189
2.3.1.1.5	(METHOD WS-ALIGNMENT BMI).....		190
2.3.1.1.6	(METHOD SET-WS- ALIGNMENT BMI).....		190
2.3.1.1.7	WORKSTATION-ALIGNMENT.....		190
2.3.1.1.8	(METHOD BATTLE-VIEW BMI)....		191
2.3.1.1.9	(METHOD SET-BATTLE-VIEW BMI) .....		191
2.3.1.1.10	WORKSTATION-BATTLE- VIEW .....		191
2.3.1.1.11	(METHOD BATTLE-SCHEME BMI) .....		191
2.3.1.1.12	(METHOD SET-BATTLE- SCHEME BMI) .....		192
2.3.1.1.13	WORKSTATION-BATTLE- SCHEME.....		192
2.3.1.1.14	*DEFAULT-BATTALION- NUMBER* .....		192
2.3.1.1.15	GET-BATTALION-NUMBER.....		193
2.3.1.1.16	(METHOD ACCEPT-BMI- OPTIONS BMI).....		193
2.3.1.1.17	(METHOD AFTER-PROGRAM- FRAME-SELECTION- HANDLER BMI) .....		193
2.3.1.1.18	(METHOD BMI-SANDBOX BMI).....		194
2.3.1.1.19	(METHOD BMI-SET-SANDBOX BMI).....		194
2.3.1.1.20	(METHOD BMI-REMOVE- SANDBOX-OBJECT BMI) .....		194

2.3.1.1.21	(METHOD BMI-CLEAR-SANDBOX BMI) .....	194
2.3.1.1.22	(METHOD BMI-ADD-SANDBOX-OBJECT BMI) .....	195
2.3.1.1.23	(METHOD BMI-AIRPORTS BMI) .....	195
2.3.1.1.24	(METHOD BMI-SET-AIRPORTS BMI) .....	195
2.3.1.1.25	(METHOD BMI-ADD-AIRPORT BMI) .....	195
2.3.1.1.26	(METHOD FIND-AIRPORT BMI) .....	195
2.3.1.1.27	(METHOD DISPLAY-CONNECTION-STATE BMI) .....	196
2.3.1.1.28	(METHOD CREATE-MOCK-UNITS BMI) .....	196
2.3.1.1.29	REALLY-MAKE-SANDBOX-OBJECT .....	196
2.3.1.1.30	RETURN-FORCE-AND-COUNTRY-D-AND-O .....	197
2.3.1.1.31	ALIGNMENT-FROM-FORCE-ID .....	197
2.3.1.1.32	(ACCEPT-TACTICS-AND-TEAM BMI) .....	198
2.3.1.1.33	ACCEPT-PARAMETER-FROM-SEQUENCE .....	198
2.3.1.1.34	ALL-ECHELONS .....	198
2.3.1.1.35	GET-ECHELON-TYPES .....	199
2.3.1.1.36	BMI-FIND-FORMATIONS .....	199
2.3.1.1.37	(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI) .....	199
2.3.1.1.38	FIND-ALL-FWA-ECHELONS .....	201
2.3.1.1.39	(METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI) .....	201
2.3.1.1.40	(METHOD DISPLAY-FWA-PANE BMI) .....	202
2.3.1.1.41	(METHOD DISPLAY-TOTALS-PANE BMI) .....	202
2.3.1.1.42	(METHOD REDISPLAY-TOTALS-PANE BMI) .....	203
2.3.1.1.43	(METHOD REDISPLAY-OPTIONS-PANE BMI) .....	203

2.3.1.2	CSU bmi>commands.lisp.....	203
2.3.1.2.1	BATTLEMASTER-SCREEN-P.....	203
2.3.1.2.2	(COM-SELECT-UNITS MENU- ACCELERATOR Select Units MENU-LEVEL BATTLEMASTER).....	204
2.3.1.2.3	(COM-CLEAR-SELECTIONS MENU-ACCELERATOR Clear Selections MENU-LEVEL BATTLEMASTER).....	204
2.3.1.2.4	(COM-RESTORE-EXERCISE MENU-ACCELERATOR Restore Exercise MENU-LEVEL BATTLEMASTER).....	204
2.3.1.2.5	(COM-SAVE-SELECTIONS MENU-ACCELERATOR Save Selections MENU-LEVEL BATTLEMASTER).....	205
2.3.1.2.6	(COM-LOAD-SELECTIONS MENU-ACCELERATOR Load Selections MENU-LEVEL BATTLEMASTER).....	205
2.3.1.2.7	(COM-CREATE-UNITS MENU- ACCELERATOR Create Units MENU-LEVEL BATTLEMASTER).....	205
2.3.1.2.8	(COM-SHOW-SANDBOX).....	205
2.3.1.2.9	(COM-ADD-AIRCRAFT).....	206
2.3.1.2.10	COM-BATTLEMASTER .....	206
2.3.1.2.11	COM-COMMANDER .....	206
2.3.1.3	CSU bmi>utilities.lisp.....	206
2.3.1.3.1	USER-CHOOSE .....	206
2.3.1.3.2	RETRIEVE-A-SANDBOX .....	207
2.3.1.3.3	CONVERT-UNIT-SIZE .....	207
2.3.1.3.4	CONVERT-ALIGNMENT .....	208
2.3.1.3.5	BMI-MAKE-SANDBOX- OBJECT.....	208
2.3.1.3.6	OPFOR-SYMBOL .....	208
2.3.1.3.7	FIND-GOOD-LOCAL-FILE- SERVER .....	208
2.3.1.3.8	MAYBE-LOAD-FORMATION- DATA.....	209
2.3.1.4	CSU bmi>airport.lisp.....	209

	2.3.1.4.1	AIRPORT-DATA .....	209
	2.3.1.4.2	DRAW-AIRPORT-LOCATION.....	210
	2.3.1.4.3	AIRPORT .....	210
	2.3.1.4.4	(METHOD MAKE-INSTANCE AIRPORT AFTER) .....	210
	2.3.1.4.5	(METHOD DRAW AIRPORT).....	210
	2.3.1.4.6	(METHOD MAKE-FWA- SANDBOX-OBJECT AIRPORT) ...	211
	2.3.1.4.7	MAKE-AIRPORT .....	211
	2.3.1.4.8	MAKE-AIRPORTS .....	211
2.3.1.5		CSU bmi>presentation-types.lisp.....	211
	2.3.1.5.1	NO-CONNECTION .....	212
	2.3.1.5.2	MAKE-CONNECTION .....	212
	2.3.1.5.3	CONNECTION .....	212
	2.3.1.5.4	END-CONNECTION .....	212
	2.3.1.5.5	AIRPORT .....	213
	2.3.1.5.6	ADD-AIRCRAFT .....	213
	2.3.1.5.7	SANDBOX-OBJECT.....	213
	2.3.1.5.8	SANDBOX-OBJECT-GESTURE....	214
	2.3.1.5.9	TACTICS .....	214
	2.3.1.5.10	SIMNET-TEAM.....	215
	2.3.1.5.11	MILS.....	215
	2.3.1.5.12	BATTALION-BUMPER.....	215
	2.3.1.5.13	COMPANY-BUMPER.....	216
	2.3.1.5.14	PLATOON-BUMPER.....	216
2.3.2		Sandbox CSC .....	216
	2.3.2.1	CSU sandbox>sandbox.lisp.....	217
	2.3.2.1.1	SANDBOX .....	217
	2.3.2.1.2	COPY-SANDBOX.....	217
	2.3.2.1.3	DRAW-SANDBOX.....	218
	2.3.2.1.4	ERASE-SANDBOX .....	218
	2.3.2.1.5	STORE-SANDBOX .....	218
	2.3.2.1.6	WRITE-SANDBOX .....	219
	2.3.2.1.7	FORMATION-CACHE-ENTRY.....	219
	2.3.2.1.8	*FORMATION-CACHE* .....	219
	2.3.2.1.9	*DEBUG-FCE* .....	219
	2.3.2.1.10	CACHE-FORMATION-INFO .....	220

	2.3.2.1.11	FIND-FORMATION-INFO .....	220
2.3.2.2		CSU sandbox>sandbox-object.lisp.....	221
	2.3.2.2.1	SANDBOX-OBJECT.....	221
	2.3.2.2.2	COPY-SANDBOX-OBJECT.....	222
	2.3.2.2.3	SANDBOX-OBJECT-ALU.....	222
	2.3.2.2.4	SANDBOX-OBJECT- COUNTRY .....	223
	2.3.2.2.5	DRAW-SANDBOX-OBJECT.....	223
	2.3.2.2.6	ERASE-SANDBOX-OBJECT .....	224
	2.3.2.2.7	DRAW-SANDBOX-UNIT .....	225
2.3.2.3		CSU sandbox>utilities.lisp .....	225
	2.3.2.3.1	ACTIVE-SANDBOXES-AS- MENU-ITEMS .....	226
	2.3.2.3.2	NAMES-OF-DISK- SANDBOXES.....	226
	2.3.2.3.3	SYMBOL-VS-CAR-LIST-TEST.....	226
	2.3.2.3.4	ALL-SANDBOXES-AS-MENU- ITEMS.....	226
	2.3.2.3.5	'GET-LOCATION-AND- BEARING .....	227
	2.3.2.3.6	GET-LOCATION-AND- BEARING .....	227
2.3.3		scenario CSC.....	228
	2.3.3.1	CSU objects>storable-mixin.lisp.....	228
	2.3.3.1.1	STORABLE-MIXIN.....	228
	2.3.3.2	CSU sys>new-storage.lisp .....	229
	2.3.3.2.1	*DBASE-FILE*.....	230
	2.3.3.2.2	COERCE-STRING .....	230
	2.3.3.2.3	MKATOM.....	230
	2.3.3.2.4	GET-INSTANCE-VARIABLES.....	232
	2.3.3.2.5	ITERATED-SYMBOL.....	232
	2.3.3.2.6	RETURN-ITERATED-SYMBOL.....	232
	2.3.3.2.7	GET-VALUE-SUBST.....	233
	2.3.3.2.8	GET-VALUE.....	233
	2.3.3.2.9	REPLACE-VALUE-SUBST .....	233
	2.3.3.2.10	REPLACE-VALUE.....	234
	2.3.3.2.11	SAVE-TOP-LEVEL-AND- INFERIORS .....	234

2.3.3.2.12	MAKE-OBJECT-LIST- RECURSIVE .....	234
2.3.3.2.13	SAVE-IN-DATABASE.....	235
2.3.3.2.14	SAVE-INSTANCE .....	235
2.3.3.2.15	REPLACE-SLOT-VALUE- OBJECTS .....	235
2.3.3.2.16	READ-AND-MAKE- INSTANCES .....	236
2.3.3.2.17	REPLACE-SLOT-VALUE- INSTANCE-NAMES.....	236
2.3.3.2.18	REMOVE-LEFTOVER-SLOT- VALUE-INSTANCE-NAMES.....	237
2.3.3.2.19	REMOVE-LEFTOVER- INSTANCE-NAMES.....	237
2.3.3.2.20	REMOVE-LEFTOVER-DB- INSTANCES .....	237
2.3.3.2.21	*SCENARIO* .....	238
2.3.3.2.22	*SAVE-INSTANCE-FILTER* .....	238
2.3.3.2.23	SAVE-FOR-TASKING-P .....	238
2.3.3.2.24	FILTERED-SAVE-INSTANCE.....	239
2.3.3.2.25	CONCATLIST .....	239
2.3.3.2.26	CONCAT.....	240
2.3.3.2.27	SCENARIO.....	241
2.3.3.2.28	CLOSE-ENOUGH.....	241
2.3.3.2.29	(METHOD ADJUST-VIEWPORT SCENARIO).....	242
2.3.3.2.30	GET-SCREEN-PARAMETERS .....	242
2.3.3.2.31	*OVERLAY-TO-SAVE* .....	242
2.3.3.2.32	NAME-AND-STORE-OVERLAY .....	243
2.3.3.2.33	SAVE-OR-LOAD-OVERLAYS .....	243
2.3.3.2.34	NAME-AND-STORE- SCENARIO.....	243
2.3.3.2.35	REMOVE-DOTS-FROM- STRING.....	244
2.3.3.2.36	STORE-SCENARIO.....	244
2.3.3.2.37	RETURN-SCENARIO-OBJECT- LIST .....	245
2.3.3.2.38	GET-CURRENT-TOP-UNITS .....	245
2.3.3.2.39	*SCENARIO-DIRECTORY* .....	246
2.3.3.2.40	*OVERLAY-DIRECTORY* .....	246

2.3.3.2.41	(METHOD STORE SCENARIO).....	246
2.3.3.2.42	LOAD-OVERLAY .....	247
2.3.3.2.43	LOAD-SCENARIO .....	247
2.3.3.2.44	CREATE-STORED-INSTANCE .....	248
2.3.3.2.45	SET-INFERIORS-PORT-AND- SUPERIOR-ID .....	249
2.3.3.2.46	COPY-RELEVANT-IVS.....	249
2.3.3.2.47	*DELETE-TEXT-FILES- MENU* .....	250
2.3.3.2.48	MULTIPLE-MENU-CHOOSE.....	250
2.3.3.2.49	CHOOSE-SCENARIOS-TO- DELETE.....	250
2.3.3.2.50	CHOOSE-OVERLAYS-TO- DELETE.....	251
2.4	MAP DISPLAY CSC.....	252
2.4.1	Color CSC .....	252
2.4.1.1	CSU sys>update-process.lisp .....	252
2.4.1.1.1	*TERRAIN-TO-DRAW* .....	253
2.4.1.1.2	*TERRAIN-CONTOURS-TO- DRAW* .....	253
2.4.1.1.3	*EFFECTS-ERASE-TIME* .....	254
2.4.1.1.4	*UPDATE-PROCESS-WAIT- TIME* .....	254
2.4.1.1.5	*UPDATE-PROCESS-LAST- CYCLE* .....	254
2.4.1.1.6	*UPDATE-PROCESS-MAX- WAIT-TIME* .....	254
2.4.1.1.7	*TIME-LAST-POLLED* .....	255
2.4.1.1.8	UPDATE-PROCESS-WAKE-UP ....	255
2.4.1.1.9	UPDATE-TOP-LEVEL.....	255
2.4.1.1.10	UPDATE-TOP-LEVEL-AUX .....	256
2.4.1.1.11	PROCESS-USER-COMMAND.....	257
2.4.1.1.12	POLL-COMPLETED .....	257
2.4.1.1.13	PROCESS-NETWORK- COMMAND .....	258
2.4.1.1.14	PROCESS-NEW-MAP- OPTIONS .....	258
2.4.1.1.15	*SOIL-TYPES* .....	259
2.4.1.1.16	DRAW-MAP.....	259



	2.4.1.1.17	DRAW-ANOTHER-TERRAIN- QUAD .....	260
2.4.1.2		CSU ui>frame-utils.lisp .....	260
	2.4.1.2.1	MAP-WINDOW .....	260
	2.4.1.2.2	MAP-WINDOW .....	261
	2.4.1.2.3	MAP-LEGEND .....	261
	2.4.1.2.4	MAP-LEGEND .....	261
	2.4.1.2.5	HIGHLIGHT-BUTTON .....	261
	2.4.1.2.6	HIGHLIGHT-BUTTON-1 .....	262
2.4.2		Terrain Display CSC .....	262
	2.4.2.1	CSU map>clip.lisp .....	263
	2.4.2.1.1	ROTATE-90-C .....	263
	2.4.2.1.2	ROTATE-180-C .....	263
	2.4.2.1.3	ROTATE-270-C .....	264
	2.4.2.1.4	REFLECT-X-MINUS-Y .....	264
	2.4.2.1.5	REFLECT-X-AXIS .....	264
	2.4.2.1.6	*DISPLAY* .....	265
	2.4.2.1.7	CLIP .....	265
	2.4.2.1.8	LEFT-COLUMN .....	266
	2.4.2.1.9	TOP-LEFT-CORNER .....	266
	2.4.2.1.10	LEFT-BOTTOM-REGION .....	266
	2.4.2.1.11	LEFT-EDGE .....	267
	2.4.2.1.12	P2-BOTTOM .....	267
	2.4.2.1.13	CENTER-COLUMN .....	267
	2.4.2.1.14	P2-LEFT-TOP .....	268
	2.4.2.1.15	P2-LEFT .....	268
	2.4.2.1.16	INSIDE .....	268
	2.4.2.2	CSU map>color-map.lisp .....	268
	2.4.2.2.1	*OVERLAY-ALU* .....	269
	2.4.2.2.2	*ERASE-OVERLAY-ALU* .....	269
	2.4.2.2.3	*OVERLAY-ALU* .....	269
	2.4.2.2.4	*ERASE-OVERLAY-ALU* .....	270
	2.4.2.2.5	*SOIL-ALU* .....	272
	2.4.2.2.6	*OBJECT-ALU* .....	272
	2.4.2.2.7	*TREE-ALU* .....	272
	2.4.2.2.8	*SOIL-ROAD-ALU* .....	273

	2.4.2.2.9	*SOIL-RAIL-ALU* .....	273
	2.4.2.2.10	*SOIL-WATER-ALU* .....	274
	2.4.2.2.11	*SOIL-MUCK-ALU* .....	274
	2.4.2.2.12	*LOW-CONTOUR-ALU* .....	274
	2.4.2.2.13	*HIGH-CONTOUR-ALU* .....	275
	2.4.2.2.14	*LEGEND-TEXT-ALU* .....	275
	2.4.2.2.15	'MAKE-AN-ALU .....	276
	2.4.2.2.16	MAKE-AN-ALU .....	276
	2.4.2.2.17	'MAKE-ALU-AND-SET-COLOR-MAP .....	276
	2.4.2.2.18	MAKE-ALU-AND-SET-COLOR-MAP .....	276
	2.4.2.2.19	SETUP-COLOR-ALUS .....	277
	2.4.2.2.20	MAKE-COLOR-ARRAY .....	277
	2.4.2.2.21	MAKE-COLOR-ALUS .....	278
	2.4.2.2.22	SET-COLOR-MAP .....	278
2.4.2.3	CSU map>control.lisp .....		279
	2.4.2.3.1	*UNIT-TYPES* .....	279
	2.4.2.3.2	*AREA-TYPES* .....	279
	2.4.2.3.3	*LINE-TYPES* .....	279
	2.4.2.3.4	*CONTROL-MEASURE-MENU-ITEMS* .....	280
	2.4.2.3.5	*CONTROL-MEASURES* .....	280
	2.4.2.3.6	CONTROL-MEASURE .....	280
	2.4.2.3.7	'CONTROL-MEASURE .....	281
	2.4.2.3.8	CONTROL-MEASURE .....	281
	2.4.2.3.9	'AREA-CONTROL-MEASURE .....	281
	2.4.2.3.10	AREA-CONTROL-MEASURE .....	282
	2.4.2.3.11	'BATTLE-POSITION .....	282
	2.4.2.3.12	BATTLE-POSITION .....	282
	2.4.2.3.13	'LINE-CONTROL-MEASURE .....	282
	2.4.2.3.14	LINE-CONTROL-MEASURE .....	283
	2.4.2.3.15	'UNIT-BOUNDARY .....	283
	2.4.2.3.16	UNIT-BOUNDARY .....	283
	2.4.2.3.17	'ARROW-CONTROL-MEASURE .....	283
	2.4.2.3.18	ARROW-CONTROL-MEASURE .....	284

2.4.2.3.19	(METHOD INIT CONTROL- MEASURE AFTER).....	284
2.4.2.3.20	(METHOD EDIT CONTROL- MEASURE).....	284
2.4.2.3.21	(METHOD DRAW AREA- CONTROL-MEASURE).....	285
2.4.2.3.22	(METHOD ENTER-NEW- CONTROL-MEASURE AREA- CONTROL-MEASURE).....	285
2.4.2.3.23	(METHOD DRAW BATTLE- POSITION AFTER).....	285
2.4.2.3.24	(METHOD ENTER-NEW- CONTROL-MEASURE BATTLE- POSITION).....	286
2.4.2.3.25	(METHOD DRAW LINE- CONTROL-MEASURE).....	286
2.4.2.3.26	(METHOD ENTER-NEW- CONTROL-MEASURE LINE- CONTROL-MEASURE).....	286
2.4.2.3.27	(METHOD DRAW UNIT- BOUNDARY AFTER).....	287
2.4.2.3.28	(METHOD ENTER-NEW- CONTROL-MEASURE UNIT- BOUNDARY).....	287
2.4.2.3.29	(METHOD DRAW ARROW- CONTROL-MEASURE).....	287
2.4.2.3.30	(METHOD ENTER-NEW- CONTROL-MEASURE ARROW- CONTROL-MEASURE).....	288
2.4.2.3.31	AREA-CONTROL-MEASURE .....	288
2.4.2.3.32	BATTLE-POSITION .....	288
2.4.2.3.33	LINE-CONTROL-MEASURE .....	289
2.4.2.3.34	UNIT-BOUNDARY .....	289
2.4.2.3.35	ARROW-CONTROL-MEASURE ...	289
2.4.2.3.36	'WITH-COLOR-MOUSE.....	290
2.4.2.3.37	WITH-COLOR-MOUSE.....	290
2.4.2.3.38	ROTATABLE-RECTANGLE .....	290
2.4.2.3.39	DRAW-ROT-RECT.....	291
2.4.2.3.40	'RUBBER-LINE.....	291
2.4.2.3.41	RUBBER-LINE.....	292
2.4.2.3.42	'SELECT-POLYGON.....	292
2.4.2.3.43	SELECT-POLYGON.....	293

	2.4.2.3.44	'SINGLE-POINT .....	293
	2.4.2.3.45	SINGLE-POINT .....	293
	2.4.2.3.46	'DRAW-UNIT-SYMBOL.....	294
	2.4.2.3.47	DRAW-UNIT-SYMBOL.....	294
	2.4.2.3.48	DRAW-1-SCALLOPED-LINE.....	295
	2.4.2.3.49	DRAW-2-SCALLOPED-LINES .....	296
	2.4.2.3.50	FIND-CENTER-POINT .....	296
	2.4.2.3.51	DRAW-ARROW .....	297
	2.4.2.3.52	'CONTROL-MEASURES- MENU.....	297
	2.4.2.3.53	CONTROL-MEASURES-MENU ....	297
	2.4.2.3.54	'EDIT-CONTROL-MEASURES.....	298
	2.4.2.3.55	EDIT-CONTROL-MEASURES.....	298
	2.4.2.3.56	'DRAW-ALL-CONTROL- MEASURES.....	298
	2.4.2.3.57	DRAW-ALL-CONTROL- MEASURES.....	298
2.4.2.4		CSU map>draw-wide-curve.lisp.....	299
	2.4.2.4.1	(METHOD MAP-DRAW-WIDE- CURVE GRAPHICS-MIXIN).....	299
	2.4.2.4.2	(METHOD MAP-DRAW- TAPERED-WIDE-CURVE GRAPHICS-MIXIN).....	299
2.4.2.5		CSU map>grids.lisp.....	300
	2.4.2.5.1	(METHOD GRID-INC UTM- GRID-MIXIN) .....	300
	2.4.2.5.2	(METHOD LEFT-X-GRID UTM- GRID-MIXIN) .....	300
	2.4.2.5.3	(METHOD RIGHT-X-GRID UTM-GRID-MIXIN).....	300
	2.4.2.5.4	(METHOD SW-GRID-WORLDS UTM-GRID-MIXIN).....	300
	2.4.2.5.5	'DRAW-GRIDS.....	301
	2.4.2.5.6	(METHOD DRAW-GRIDS UTM- GRID-MIXIN) .....	301
2.4.2.6		CSU map>intersection.lisp .....	301
	2.4.2.6.1	COUNT-INTERSECTIONS .....	301
	2.4.2.6.2	'POINT-INSIDE-POLYGON-P .....	302
	2.4.2.6.3	POINT-INSIDE-POLYGON-P .....	302

	2.4.2.6.4	'SEGMENT-INSIDE-POLYGON-P.....	302
	2.4.2.6.5	SEGMENT-INSIDE-POLYGON-P.....	302
	2.4.2.6.6	'SEGMENT-INTERSECTS-POLYGON-P.....	303
	2.4.2.6.7	SEGMENT-INTERSECTS-POLYGON-P.....	303
	2.4.2.6.8	BOUNDING-RECTANGLE .....	303
	2.4.2.6.9	'POSSIBLE-INTERSECTION.....	303
	2.4.2.6.10	POSSIBLE-INTERSECTION.....	304
	2.4.2.6.11	'POINT-SEGMENT-INTERSECTION .....	304
	2.4.2.6.12	POINT-SEGMENT-INTERSECTION .....	304
	2.4.2.6.13	'POINT-LINE-INTERSECTION....	304
	2.4.2.6.14	POINT-LINE-INTERSECTION....	305
2.4.2.7		CSU map>legend.lisp .....	305
	2.4.2.7.1	'LEGEND-WINDOW .....	305
	2.4.2.7.2	LEGEND-WINDOW .....	306
	2.4.2.7.3	(METHOD INIT LEGEND-WINDOW AFTER).....	306
	2.4.2.7.4	(METHOD ERASE LEGEND-WINDOW).....	306
	2.4.2.7.5	(METHOD SET-LEGEND-POSITIONS LEGEND-WINDOW).....	306
	2.4.2.7.6	'DRAW-LEGEND .....	306
	2.4.2.7.7	(METHOD DRAW-LEGEND LEGEND-WINDOW) .....	307
	2.4.2.7.8	DRAW-LEGEND-BOX-AND-LINE.....	308
	2.4.2.7.9	DRAW-LEGEND-SCALE-LINE....	308
	2.4.2.7.10	DRAW-LEGEND-BUILDINGS....	308
	2.4.2.7.11	DRAW-LEGEND-BRIDGE.....	308
	2.4.2.7.12	DRAW-LEGEND-CONTOUR-LINE.....	309
2.4.2.8		CSU map>quadtree-search.lisp .....	309
	2.4.2.8.1	'QUADS-TO-DRAW .....	309
	2.4.2.8.2	QUADS-TO-DRAW .....	309

	2.4.2.8.3	'GET-QUAD-NODES.....	310
	2.4.2.8.4	GET-QUAD-NODES.....	310
	2.4.2.8.5	GET-THIS-NODE.....	310
2.4.2.9		CSU map>scalable-window.lisp.....	311
	2.4.2.9.1	'SCALABLE-WINDOW.....	311
	2.4.2.9.2	SCALABLE-WINDOW.....	311
	2.4.2.9.3	(METHOD INIT SCALABLE- WINDOW AFTER).....	311
	2.4.2.9.4	(METHOD UPDATE SCALABLE-WINDOW).....	311
	2.4.2.9.5	(METHOD CLEAR-COORDS SCALABLE-WINDOW).....	312
	2.4.2.9.6	(METHOD NEW-SCALE- INTERNAL SCALABLE- WINDOW).....	312
	2.4.2.9.7	(METHOD NEW-SCALE SCALABLE-WINDOW).....	312
	2.4.2.9.8	(METHOD NEW-SCALE SCALABLE-WINDOW BEFORE).....	313
	2.4.2.9.9	(METHOD NEW-SCALE SCALABLE-WINDOW AFTER).....	313
	2.4.2.9.10	(METHOD DRAW-REGION SCALABLE-WINDOW).....	313
	2.4.2.9.11	'WINDOW-SCALE.....	313
	2.4.2.9.12	(METHOD WINDOW-SCALE SCALABLE-WINDOW).....	314
	2.4.2.9.13	(METHOD SOUTH-WEST- CORNER SCALABLE- WINDOW).....	314
	2.4.2.9.14	(METHOD SCALED-HEIGHT SCALABLE-WINDOW).....	314
	2.4.2.9.15	(METHOD SCALED-WIDTH SCALABLE-WINDOW).....	314
	2.4.2.9.16	'WORLD-EDGES.....	314
	2.4.2.9.17	(METHOD WORLD-EDGES SCALABLE-WINDOW).....	315
	2.4.2.9.18	'CURRENT-CENTER.....	315
	2.4.2.9.19	(METHOD CURRENT-CENTER SCALABLE-WINDOW).....	315
	2.4.2.9.20	'PAN-TO-NEW-POINT.....	315

2.4.2.9.21	(METHOD PAN-TO-NEW- POINT SCALABLE-WINDOW) .....316
2.4.2.9.22	'RESCALE.....316
2.4.2.9.23	(METHOD RESCALE SCALABLE-WINDOW).....316
2.4.2.9.24	'RESCALE-FROM-MENU.....316
2.4.2.9.25	(METHOD RESCALE-FROM- MENU SCALABLE-WINDOW) .....316
2.4.2.9.26	'ZOOM-TO.....317
2.4.2.9.27	(METHOD ZOOM-TO SCALABLE-WINDOW).....317
2.4.2.9.28	'ZOOM-IN .....318
2.4.2.9.29	(METHOD ZOOM-IN SCALABLE-WINDOW).....318
2.4.2.9.30	'ZOOM-OUT .....318
2.4.2.9.31	(METHOD ZOOM-OUT SCALABLE-WINDOW).....318
2.4.2.9.32	'ZOOM-IN-AROUND-CENTER .....319
2.4.2.9.33	(METHOD ZOOM-IN-AROUND- CENTER SCALABLE- WINDOW).....319
2.4.2.9.34	'ZOOM-OUT-AROUND- CENTER .....320
2.4.2.9.35	(METHOD ZOOM-OUT- AROUND-CENTER SCALABLE-WINDOW).....320
2.4.2.9.36	'ON-TERRAIN-P.....320
2.4.2.9.37	(METHOD ON-TERRAIN-P SCALABLE-WINDOW).....320
2.4.2.9.38	'ON-SCREEN-P .....321
2.4.2.9.39	(METHOD ON-SCREEN-P SCALABLE-WINDOW).....321
2.4.2.9.40	(DRAW-TRIANGLE SCALABLE-WINDOW).....321
2.4.2.9.41	(METHOD MOUSE-TO-WORLD SCALABLE-WINDOW).....321
2.4.2.9.42	(METHOD WORLD-TO-MOUSE SCALABLE-WINDOW).....322
2.4.2.10	CSU map>terrain-vars.lisp .....322
2.4.2.10.1	'(*ROAD-SEGMENT-ARRAY* *ROAD-INTERSECTION-

	ARRAY* *RAIL-SEGMENT- ARRAY* *BRIDGE-ARRAY* .....	322
2.4.2.10.2	*ROAD-SEGMENT-ARRAY* .....	323
2.4.2.10.3	*ROAD-INTERSECTION- ARRAY* .....	323
2.4.2.10.4	*TREES-ARRAY* .....	324
2.4.2.10.5	*CONTOUR-ARRAY* .....	324
2.4.2.10.6	*OBJECT-ARRAY* .....	324
2.4.2.10.7	*CANOPY-ARRAY* .....	325
2.4.2.10.8	*CANOPY-TRIANGLES* .....	325
2.4.2.10.9	*WATER-SEGMENT-ARRAY* .....	325
2.4.2.10.10	*WATER-INTERSECTION- ARRAY* .....	326
2.4.2.10.11	*BRIDGE-ARRAY* .....	326
2.4.2.10.12	*RAIL-SEGMENT-ARRAY* .....	326
2.4.2.10.13	*WATER-AREA-ARRAY* .....	327
2.4.2.10.14	*WATER-AREA-TRIANGLES* .....	327
2.4.2.10.15	*X-ORIGIN* .....	327
2.4.2.10.16	*X-MAXIMUM* .....	328
2.4.2.10.17	*Y-ORIGIN* .....	328
2.4.2.10.18	*Y-MAXIMUM* .....	328
2.4.2.10.19	*COLOR-MAP* .....	328
2.4.2.10.20	*QUAD-TREE* .....	328
2.4.2.10.21	*QUAD-TREE* .....	329
2.4.2.10.22	'(QUAD-TREE-DB-NAME QUAD-TREE-VERSION .....	330
2.4.2.10.23	QUAD-TREE .....	330
2.4.2.10.24	QUAD-TREE-DEFAULT .....	331
2.4.2.10.25	'(QUAD-FEATURES QUAD- NW-NODE QUAD-NE-NODE QUAD-SE-NODE QUAD-SW- NODE) .....	331
2.4.2.10.26	QUAD-NODE .....	331
2.4.2.10.27	'(AREA-ROAD-SEGMENTS AREA-ROAD-INTERSECTIONS ...	332
2.4.2.10.28	FEATURE-NODE .....	332
2.4.2.10.29	*FEATURE-LIST* .....	332
2.4.2.10.30	'(SEGMENT-POINTS SEGMENT-WIDTH SEGMENT-	



	HEIGHT SEGMENT-ELEVATION) .....	333
2.4.2.10.31	SEGMENT.....	333
2.4.2.10.32	SEGMENT-HEIGHT .....	333
2.4.2.10.33	SEGMENT-ELEVATION .....	333
2.4.2.10.34	'(NET-POINTS NET-WIDTH.....	334
2.4.2.10.35	NETWORK-SEGMENT.....	334
2.4.2.10.36	NETWORK-INTERSECTION.....	334
2.4.2.10.37	'(BRIDGE-POINTS BRIDGE-NODE BRIDGE-WIDTH).....	335
2.4.2.10.38	BRIDGE.....	335
2.4.2.11	CSU map>utilities.lisp .....	335
2.4.2.11.1	PIE.....	335
2.4.2.11.2	GRAPHICS-TRANSFORM.....	336
2.4.2.11.3	WITH-INTEGER-CONVERSION-MODE .....	336
2.4.2.11.4	'WITH-MAP-GRAPHICS.....	339
2.4.2.11.5	WITH-MAP-GRAPHICS.....	339
2.4.2.11.6	'WITH-FAST-MAP-GRAPHICS ...	342
2.4.2.11.7	WITH-FAST-MAP-GRAPHICS ...	342
2.4.2.11.8	'SCREEN-TO-WORLD.....	344
2.4.2.11.9	SCREEN-TO-WORLD.....	345
2.4.2.11.10	'WORLD-TO-SCREEN.....	345
2.4.2.11.11	WORLD-TO-SCREEN.....	345
2.4.2.11.12	WITH-ULTRA-FAST-GRAPHICS.....	346
2.4.2.11.13	FAST-WORLD-TO-SCREEN.....	347
2.4.2.11.14	TRANSFORM-POINT.....	347
2.4.2.11.15	'DISTANCE.....	348
2.4.2.11.16	DISTANCE.....	348
2.4.2.11.17	'NEAR.....	349
2.4.2.11.18	NEAR.....	349
2.4.2.11.19	SAFE-ATAN .....	350
2.4.2.12	CSU map>utm-grid-mixin.lisp.....	350
2.4.2.12.1	*ALPHABET-ARRAY* .....	351
2.4.2.12.2	FILL-ALPHABET-ARRAY.....	351
2.4.2.12.3	NIL.....	351

	2.4.2.12.4	CHAR-TO-COORD.....	351
	2.4.2.12.5	COORD-TO-CHAR.....	352
	2.4.2.12.6	UTM-OFFSET .....	352
	2.4.2.12.7	'UTM-GRID-MIXIN.....	352
	2.4.2.12.8	UTM-GRID-MIXIN .....	352
	2.4.2.12.9	(METHOD UPDATE UTM- GRID-MIXIN AFTER) .....	353
	2.4.2.12.10	'SET-ORIGIN-UTM- COORDINATES .....	353
	2.4.2.12.11	(METHOD SET-ORIGIN-UTM- COORDINATES UTM-GRID- MIXIN).....	353
	2.4.2.12.12	'WORLD-TO-UTM.....	353
	2.4.2.12.13	(METHOD WORLD-TO-UTM UTM-GRID-MIXIN).....	354
	2.4.2.12.14	'UTM-TO-WORLD.....	354
	2.4.2.12.15	(METHOD UTM-TO-WORLD UTM-GRID-MIXIN).....	354
	2.4.2.12.16	UTM-GRID-MIXIN.....	354
2.4.2.13		CSU map>vectors.lisp .....	355
	2.4.2.13.1	'VEC-NORMALIZE.....	355
	2.4.2.13.2	VEC-NORMALIZE.....	355
	2.4.2.13.3	'VEC-ROTATE.....	356
	2.4.2.13.4	VEC-ROTATE.....	356
	2.4.2.13.5	'VEC-ADD.....	357
	2.4.2.13.6	VEC-ADD.....	357
	2.4.2.13.7	'VEC-SUB.....	358
	2.4.2.13.8	VEC-SUB.....	358
	2.4.2.13.9	'VEC-SCALE.....	359
	2.4.2.13.10	VEC-SCALE.....	359
	2.4.2.13.11	'VEC-ANGLE.....	360
	2.4.2.13.12	VEC-ANGLE.....	360
	2.4.2.13.13	'FIND-INTER-POINT .....	360
	2.4.2.13.14	FIND-INTER-POINT .....	361
	2.4.2.13.15	DRAW-BRIDGE-SYMBOL.....	361
2.4.2.14		CSU map>zoom-levels.lisp.....	362
	2.4.2.14.1	ZOOM-LEVEL .....	362
	2.4.2.14.2	'*ZOOM-LEVELS*.....	362

2.4.2.14.3	*ZOOM-LEVELS*.....	363
2.4.2.14.4	*CURRENT-ZOOM-LEVEL* .....	363
2.4.2.14.5	*CURRENT-ZOOM-LEVEL* .....	363
2.4.2.14.6	'SCALE-STRING .....	365
2.4.2.14.7	SCALE-STRING .....	365
2.4.2.14.8	'MAJOR-CONTOUR-LINE- INTERVAL.....	366
2.4.2.14.9	MAJOR-CONTOUR-LINE- INTERVAL.....	366
2.4.2.14.10	'MINOR-CONTOUR-LINE- INTERVAL.....	366
2.4.2.14.11	MINOR-CONTOUR-LINE- INTERVAL.....	367
2.4.2.14.12	'CONTOUR-POINT-INTERVAL....	367
2.4.2.14.13	CONTOUR-POINT-INTERVAL....	367
2.4.2.14.14	'DRAW-TREELINES .....	368
2.4.2.14.15	DRAW-TREELINES .....	368
2.4.2.14.16	'DRAW-TREELINE-AS-SPLINE....	368
2.4.2.14.17	DRAW-TREELINE-AS-SPLINE....	368
2.4.2.14.18	'DRAW-ROADS-WITH-WIDTH.....	369
2.4.2.14.19	DRAW-ROADS-WITH-WIDTH.....	369
2.4.2.14.20	'DRAW-WATER-WITH-WIDTH....	369
2.4.2.14.21	DRAW-WATER-WITH-WIDTH....	369
2.4.2.14.22	'DRAW-RAILS-WITH-WIDTH.....	370
2.4.2.14.23	DRAW-RAILS-WITH-WIDTH.....	370
2.4.2.14.24	'CURRENT-SCALE .....	370
2.4.2.14.25	CURRENT-SCALE .....	370
2.4.2.14.26	'CURRENT-ANCHOR-X.....	371
2.4.2.14.27	CURRENT-ANCHOR-X.....	371
2.4.2.14.28	'CURRENT-ANCHOR-Y .....	372
2.4.2.14.29	CURRENT-ANCHOR-Y .....	372
2.4.2.14.30	'LEGEND-SIZE.....	373
2.4.2.14.31	LEGEND-SIZE.....	373
2.4.2.14.32	'LEGEND-LENGTH.....	373
2.4.2.14.33	LEGEND-LENGTH.....	373
2.4.2.14.34	'NEXT-ZOOM-OUT .....	374
2.4.2.14.35	NEXT-ZOOM-OUT .....	374

	2.4.2.14.36	'NEXT-ZOOM-IN .....	374
	2.4.2.14.37	NEXT-ZOOM-IN .....	374
	2.4.2.14.38	MAKE-FT-KNOX-ZOOM- LEVELS.....	375
	2.4.2.14.39	*ZOOM-LEVELS*.....	375
	2.4.2.14.40	*CURRENT-ZOOM-LEVEL* .....	376
	2.4.2.14.41	MAKE-HUNTERLGT-ZOOM- LEVELS.....	378
	2.4.2.15	CSU map>draw-terrain.lisp .....	378
2.4.3		Vehicle and Effects Display CSC .....	379
	2.4.3.1	CSU color-window>color-alus.lisp .....	379
	2.4.3.1.1	('*ERASE-VEHICLES-ALU* *DEFENSE-ALU* *OFFENSE- ALU* *TRIM-ALU* *ERASE- EFFECTS-ALU*.....	380
	2.4.3.1.2	SETUP-COLOR-ALUS.....	380
	2.4.3.1.3	Init Window.....	381
	2.4.3.2	CSU fonts>bluefor-icons.bfd .....	381
	2.4.3.3	CSU fonts>opfor-icons.bfd .....	382
	2.4.3.4	CSU simnet-objects>draw-vehicles.lisp .....	382
	2.4.3.4.1	ERASE-VEHICLE-ALU.....	382
	2.4.3.4.2	WITH-CORRECT-MAP- GRAPHICS.....	383
	2.4.3.4.3	*MIN-IMAGE-SCALE* .....	383
	2.4.3.4.4	DRAW-BOX.....	384
	2.4.3.4.5	DRAW-FILLED-BOX.....	384
	2.4.3.4.6	('DRAW-IMAGE ERASE- IMAGE).....	385
	2.4.3.4.7	DRAW-IMAGE .....	385
	2.4.3.4.8	ERASE-IMAGE.....	386
	2.4.3.4.9	UPDATE-SCALE.....	386
	2.4.3.4.10	IMAGE .....	386
	2.4.3.4.11	(METHOD UPDATE-SCALE IMAGE).....	386
	2.4.3.4.12	(METHOD ERASE-IMAGE IMAGE).....	387
	2.4.3.4.13	(METHOD DRAW-IMAGE IMAGE BEFORE).....	387
	2.4.3.4.14	(METHOD DRAW-IMAGE IMAGE).....	387

2.4.3.4.15	IMAGE .....	387
2.4.3.4.16	HELO-IMAGE .....	388
2.4.3.4.17	(METHOD UPDATE-SCALE HELO-IMAGE) .....	389
2.4.3.4.18	(METHOD DRAW-IMAGE HELO-IMAGE) .....	389
2.4.3.4.19	LOCAL-HELO-IMAGE.....	390
2.4.3.4.20	REMOTE-HELO-IMAGE.....	390
2.4.3.4.21	HELO-IMAGE .....	390
2.4.3.4.22	FIGHTER-IMAGE .....	390
2.4.3.4.23	(METHOD UPDATE-SCALE FIGHTER-IMAGE) .....	391
2.4.3.4.24	(METHOD DRAW-IMAGE FIGHTER-IMAGE) .....	391
2.4.3.4.25	LOCAL-FIGHTER-IMAGE.....	392
2.4.3.4.26	REMOTE-FIGHTER-IMAGE.....	392
2.4.3.4.27	FIGHTER-IMAGE .....	392
2.4.3.4.28	GROUND-VEHICLE-IMAGE.....	392
2.4.3.4.29	(METHOD UPDATE-HULL- SCALE GROUND-VEHICLE- IMAGE) .....	393
2.4.3.4.30	(METHOD UPDATE-TURRET- SCALE GROUND-VEHICLE- IMAGE) .....	393
2.4.3.4.31	(METHOD UPDATE- COMPARTMENT-SCALE GROUND-VEHICLE-IMAGE).....	393
2.4.3.4.32	(METHOD UPDATE-MISSILE- SCALE GROUND-VEHICLE- IMAGE) .....	393
2.4.3.4.33	(METHOD UPDATE-SCALE GROUND-VEHICLE-IMAGE).....	394
2.4.3.4.34	(METHOD DRAW-HULL- IMAGE GROUND-VEHICLE- IMAGE) .....	394
2.4.3.4.35	(METHOD DRAW-TURRET- IMAGE GROUND-VEHICLE- IMAGE) .....	394
2.4.3.4.36	(METHOD DRAW- COMPARTMENT-IMAGE GROUND-VEHICLE-IMAGE).....	394

2.4.3.4.37	(METHOD DRAW-MISSILE- IMAGE GROUND-VEHICLE- IMAGE) .....	395
2.4.3.4.38	(METHOD DRAW-IMAGE GROUND-VEHICLE-IMAGE).....	395
2.4.3.4.39	GROUND-VEHICLE-IMAGE.....	395
2.4.3.4.40	HULL-IMAGE .....	396
2.4.3.4.41	(METHOD UPDATE-HULL- SCALE HULL-IMAGE) .....	396
2.4.3.4.42	(METHOD DRAW-HULL- IMAGE HULL-IMAGE).....	396
2.4.3.4.43	HULL-IMAGE .....	397
2.4.3.4.44	SQ-TURRET-IMAGE.....	397
2.4.3.4.45	(METHOD UPDATE-TURRET- SCALE SQ-TURRET-IMAGE).....	397
2.4.3.4.46	(METHOD DRAW-TURRET- IMAGE SQ-TURRET-IMAGE).....	398
2.4.3.4.47	SQ-TURRET-IMAGE.....	398
2.4.3.4.48	RD-TURRET-IMAGE.....	398
2.4.3.4.49	(METHOD UPDATE-TURRET- SCALE RD-TURRET-IMAGE).....	399
2.4.3.4.50	(METHOD DRAW-TURRET- IMAGE RD-TURRET-IMAGE) .....	399
2.4.3.4.51	RD-TURRET-IMAGE.....	400
2.4.3.4.52	A-COMPARTMENT-IMAGE.....	400
2.4.3.4.53	(METHOD UPDATE- COMPARTMENT-SCALE A- COMPARTMENT-IMAGE).....	400
2.4.3.4.54	(METHOD DRAW- COMPARTMENT-IMAGE A- COMPARTMENT-IMAGE).....	400
2.4.3.4.55	A-COMPARTMENT-IMAGE.....	401
2.4.3.4.56	B-COMPARTMENT-IMAGE.....	401
2.4.3.4.57	(METHOD UPDATE- COMPARTMENT-SCALE B- COMPARTMENT-IMAGE).....	402
2.4.3.4.58	(METHOD DRAW- COMPARTMENT-IMAGE B- COMPARTMENT-IMAGE).....	402
2.4.3.4.59	B-COMPARTMENT-IMAGE.....	402
2.4.3.4.60	MISSILE-IMAGE.....	403

2.4.3.4.61	(METHOD UPDATE-MISSILE-SCALE MISSILE-IMAGE).....	403
2.4.3.4.62	(METHOD DRAW-MISSILE-IMAGE MISSILE-IMAGE) .....	403
2.4.3.4.63	MISSILE-IMAGE.....	404
2.4.3.4.64	TANK-IMAGE.....	404
2.4.3.4.65	TANK-IMAGE.....	404
2.4.3.4.66	MECH-IMAGE.....	404
2.4.3.4.67	MECH-IMAGE.....	405
2.4.3.4.68	AMMO-TRUCK-IMAGE.....	405
2.4.3.4.69	AMMO-TRUCK-IMAGE.....	405
2.4.3.4.70	FUEL-TRUCK-IMAGE .....	406
2.4.3.4.71	FUEL-TRUCK-IMAGE .....	406
2.4.3.4.72	SUPPLY-TRUCK-IMAGE.....	406
2.4.3.4.73	SUPPLY-TRUCK-IMAGE.....	407
2.4.3.4.74	MORTAR-IMAGE.....	407
2.4.3.4.75	MORTAR-IMAGE.....	407
2.4.3.4.76	HOWITZER-IMAGE.....	407
2.4.3.4.77	HOWITZER-IMAGE.....	408
2.4.3.4.78	COMMAND-POST-IMAGE.....	408
2.4.3.4.79	COMMAND-POST-IMAGE.....	408
2.4.3.4.80	UNKNOWN-VEHICLE-IMAGE....	409
2.4.3.4.81	UNKNOWN-VEHICLE-IMAGE....	409
2.4.3.4.82	SMOKE-CLOUD-IMAGE.....	409
2.4.3.4.83	SMOKE-CLOUD-IMAGE.....	409
2.4.3.4.84	FAADS-IMAGE.....	410
2.4.3.4.85	FAADS-IMAGE.....	410
2.4.3.4.86	*IMAGE-ARRAY* .....	410
2.4.3.4.87	*LOCAL-IMAGE-TABLE* .....	411
2.4.3.4.88	*REMOTE-IMAGE-TABLE* .....	411
2.4.3.4.89	INIT-IMAGES .....	411
2.4.3.4.90	Init Images .....	412
2.4.3.4.91	'IMAGE-FOR-VEHICLE.....	412
2.4.3.4.92	IMAGE-FOR-VEHICLE.....	412
2.4.3.5	CSU simnet-objects>new-draw-vehicles.lisp .....	413
2.4.3.5.1	$\pi/8$ .....	413
2.4.3.5.2	$3\pi/8$ .....	413

	2.4.3.5.3	5 $\pi$ /8.....	414
	2.4.3.5.4	7 $\pi$ /8.....	414
	2.4.3.5.5	9 $\pi$ /8.....	414
	2.4.3.5.6	11 $\pi$ /8.....	414
	2.4.3.5.7	13 $\pi$ /8.....	415
	2.4.3.5.8	15 $\pi$ /8.....	415
	2.4.3.5.9	FIND-ICON-ROTATION.....	415
	2.4.3.5.10	*ICON-TABLE*.....	416
	2.4.3.5.11	*ICON-HASH-TABLE*.....	416
	2.4.3.5.12	INIT-VEHICLE-ICON-TABLE.....	416
	2.4.3.5.13	VEHICLE-ICON.....	417
	2.4.3.5.14	DRAW-VEHICLE-ICON.....	417
	2.4.3.5.15	DRAW-VEHICLE.....	417
2.4.3.6		CSU simnet-objects>draw-effects.....	418
	2.4.3.6.1	DRAW-IMPACT.....	418
	2.4.3.6.2	ERASE-IMPACT.....	419
	2.4.3.6.3	AMMO-TYPE-RADIUS.....	419
	2.4.3.6.4	MINE-AMMO-TYPE.....	419
	2.4.3.6.5	HANDLE-ARTY.....	420
	2.4.3.6.6	DRAW-ARTY.....	420
	2.4.3.6.7	ERASE-ELASPED-EFFECTS.....	421
2.4.3.7		CSU simnet-objects>draw-units.....	421
	2.4.3.7.1	*UNIT-ICON-TABLE*.....	421
	2.4.3.7.2	INIT-UNIT-ICON-TABLE.....	421
	2.4.3.7.3	Init Icons.....	422
	2.4.3.7.4	UNIT-ICON.....	422
	2.4.3.7.5	DRAW-UNIT.....	422
	2.4.3.7.6	'(DRAW-PLATOON DRAW- SCOUT-PLATOON DRAW-IVIS- PLATOON DRAW-COMPANY DRAW-BATTALION DRAW- PAIR.....	423
2.5		WORLD STATE CSC.....	424
	2.5.1	CSU simnet-objects>macros.lisp.....	424
		2.5.1.1 ACCESS-ID.....	424
		2.5.1.2 ACCESS-VEHICLE-INSTANCE.....	425
		2.5.1.3 ACCESS-NEW-FLAG.....	425



2.5.1.4	SET-DRAWN-FLAG.....	425
2.5.1.5	SET-NEW-FLAG.....	426
2.5.1.6	LOOKUP-ID.....	426
2.5.1.7	SET-ID .....	426
2.5.1.8	ACCESS-PAINTED-FLAG .....	427
2.5.1.9	ASSOCIATE-VEHICLE-HOLDER .....	427
2.5.1.10	IS-STATUS.....	427
2.5.1.11	DEFINE-ARRAY-ACCESSORS .....	428
2.5.1.12	X-COMP .....	428
2.5.1.13	Y-COMP .....	428
2.5.1.14	Z-COMP.....	429
2.5.1.15	DEFINE-FLAVOR-ARRAY-ACCESSORS.....	429
2.5.2	CSU objects>defobject.lisp.....	429
2.5.2.1	The SAF Object Hierarchy .....	430
2.5.2.2	*ALL-OBJECTS* .....	430
2.5.2.3	GENERATE-OBJECT-CLASS-SLOT- METHODS.....	431
2.5.2.4	DEFINE-PREDICATE-METHOD.....	431
2.5.2.5	DEFOBJECT .....	431
2.5.3	CSU objects>simnet-name-mixin.lisp.....	431
2.5.3.1	SIMNET-NAME-MIXIN .....	432
2.5.3.2	(METHOD UNIT-NAME SIMNET-NAME- MIXIN).....	432
2.5.3.3	(METHOD SET-UNIT-NAME SIMNET-NAME- MIXIN).....	432
2.5.3.4	(METHOD CLEAR-UNIT-NAME SIMNET- NAME-MIXIN).....	432
2.5.3.5	CONVERT-TYPE-FOR-NAME .....	433
2.5.3.6	CONVERT-APPEARANCE-FOR-NAME .....	433
2.5.3.7	MAKE-BATTALION-NAME .....	433
2.5.3.8	(METHOD MAKE-UNIT-NAME SIMNET- NAME-MIXIN).....	434
2.5.3.9	SIMNET-NAME-MIXIN .....	434
2.5.4	CSU objects>gunner.lisp.....	434
2.5.4.1	*MARKSMAN* .....	434
2.5.4.2	*COMPETENT* .....	435
2.5.4.3	*NOVICE* .....	435
2.5.4.4	SET-GLOBAL-FIRE-PARAMETERS .....	436

2.5.4.5	GUNNER.....	436
2.5.4.6	(METHOD GET-GUNNER-PARMS GUNNER).....	436
2.5.4.7	(METHOD SET-GUNNER-PARMS GUNNER)....	436
2.5.4.8	(METHOD SPECIFY-RULES-OF- ENGAGEMENT GUNNER).....	437
2.5.4.9	GUNNER.....	437
2.5.5	CSU objects>simnet-agent.lisp .....	438
2.5.5.1	SIMNET-AGENT.....	438
2.5.5.2	(METHOD MAKE-INSTANCE SIMNET- AGENT AFTER).....	439
2.5.5.3	(METHOD PRINT-SELF SIMNET-AGENT).....	439
2.5.5.4	(METHOD VEHICLEP SIMNET-AGENT).....	439
2.5.5.5	(METHOD COMPOSITE-OBJECT-P SIMNET- AGENT).....	440
2.5.5.6	(METHOD SET-VEHICLE-LOADS SIMNET- AGENT).....	440
2.5.5.7	(METHOD REINIT SIMNET-AGENT).....	440
2.5.5.8	(METHOD MAYBE-REPARSE- SUBORDINATES SIMNET-AGENT).....	440
2.5.5.9	(METHOD GET-SUBORDINATES SIMNET- AGENT).....	441
2.5.5.10	(METHOD SET-SUBORDINATES SIMNET- AGENT).....	441
2.5.5.11	(METHOD GET-SUBORDINATES- INSTANCES SIMNET-AGENT).....	441
2.5.5.12	(METHOD SET-SUBORDINATES- INSTANCES SIMNET-AGENT).....	441
2.5.5.13	GET-SUBORDINATES-INSTANCES.....	442
2.5.5.14	GET-SUBORDINATES .....	442
2.5.5.15	(METHOD GET-ALL-SUBORDINATES SIMNET-AGENT).....	442
2.5.5.16	(GET-SUPERIOR SET-SUPERIOR GET- SUPERIOR-INSTANCE SET-SUPERIOR- INSTANCE GET-ALL-SUPERIORS).....	443
2.5.5.17	(METHOD GET-SUPERIOR SIMNET- AGENT).....	443
2.5.5.18	(METHOD SET-SUPERIOR SIMNET-AGENT)....	443
2.5.5.19	(METHOD GET-SUPERIOR-INSTANCE SIMNET-AGENT).....	443

2.5.5.20	(METHOD SET-SUPERIOR-INSTANCE SIMNET-AGENT).....	444
2.5.5.21	GET-SUPERIOR .....	444
2.5.5.22	GET-SUPERIOR-INSTANCE.....	444
2.5.5.23	(METHOD POSSIBLE-FORMATIONS SIMNET-AGENT).....	444
2.5.5.24	(METHOD POSSIBLE-CISS SIMNET- AGENT).....	445
2.5.5.25	(METHOD HIGHLIGHT SIMNET-AGENT) .....	445
2.5.5.26	(METHOD HIGHLIGHT SIMNET-AGENT BEFORE) .....	446
2.5.5.27	(METHOD ALU SIMNET-AGENT).....	446
2.5.5.28	(METHOD COUNTRY SIMNET-AGENT).....	447
2.5.5.29	(METHOD GET-TEMPLATE SIMNET- AGENT).....	448
2.5.5.30	(ERASE SIMNET-AGENT).....	448
2.5.5.31	(METHOD ERASE SIMNET-AGENT BEFORE) .....	448
2.5.5.32	(DRAW SIMNET-AGENT).....	448
2.5.5.33	(METHOD DRAW SIMNET-AGENT AFTER) .....	449
2.5.5.34	COM-OMNISCIENT-VIEW .....	449
2.5.5.35	COM-SAF OMNISCIENT-VIEW.....	449
2.5.5.36	COM-COMMANDERS-EYE-VIEW .....	450
2.5.5.37	COM-SAF-COMMANDERS-EYE-VIEW.....	450
2.5.5.38	UNHIGHLIGHT-VIEWPORTS.....	450
2.5.5.39	HIGHLIGHT-VIEWPORTS .....	451
2.5.5.40	(METHOD SHOW-INFERIORS SIMNET- AGENT).....	451
2.5.5.41	(METHOD HIDE-INFERIORS SIMNET- AGENT).....	451
2.5.5.42	MOUSE-GESTURE-ITEM-LIST.....	451
2.5.5.43	(METHOD SHOW-VEHICLE-INFO SIMNET- AGENT).....	452
2.5.5.44	(METHOD MOUSE-GESTURE-ITEM-LIST SIMNET-AGENT APPEND) .....	452
2.5.5.45	(METHOD MOUSE-GESTURE-MENU SIMNET-AGENT).....	453
2.5.5.46	(METHOD MOUSE-GESTURE SIMNET- AGENT).....	453
2.5.5.47	GET-A-VEHICLE-TO-FOLLOW.....	453

2.5.5.48	FACE-DIRECTION .....	454
2.5.5.49	*PRINT-CHANGE-STATUS-MESSAGES* .....	454
2.5.5.50	(METHOD UPDATE-POSITION SIMNET-AGENT).....	454
2.5.5.51	(METHOD UPDATE-APPEARANCE SIMNET-AGENT).....	455
2.5.5.52	(METHOD UPDATE-ECHELON SIMNET-AGENT).....	455
2.5.5.53	(METHOD FWA-P SIMNET-AGENT).....	456
2.5.5.54	(METHOD RWA-P SIMNET-AGENT) .....	456
2.5.5.55	(METHOD AIR-VEHICLE-P SIMNET-AGENT).....	456
2.5.5.56	(METHOD GROUND-VEHICLE-P SIMNET-AGENT).....	456
2.5.5.57	(METHOD RESUME-ALL-SUBORDINATES SIMNET-AGENT).....	456
2.5.5.58	(METHOD IMMEDIATE-INTERVENTION SIMNET-AGENT).....	457
2.5.5.59	(METHOD IVIS-CONTROL SIMNET-AGENT) ...	457
2.5.5.60	SIMNET-AGENT.....	457
2.5.5.61	SIMNET-AGENT.....	458
2.5.5.62	MOUSE-UNIT-OPERATIONS.....	459
2.5.6	CSU objects>composite-object.lisp .....	459
2.5.6.1	COMPOSITE-OBJECT .....	460
2.5.6.2	(METHOD SET-CONTINUE-MISSION COMPOSITE-OBJECT) .....	460
2.5.6.3	(METHOD IMMEDIATE-INTERVENTION-CHOICES COMPOSITE-OBJECT).....	460
2.5.6.4	(METHOD MOUSE-GESTURE-ITEM-LIST COMPOSITE-OBJECT APPEND) .....	460
2.5.6.5	COMPOSITE-OBJECT .....	461
2.5.7	CSU objects>vehicle.lisp .....	461
2.5.7.1	VEHICLE .....	461
2.5.7.2	(METHOD DRAW VEHICLE).....	462
2.5.7.3	(METHOD ERASE VEHICLE) .....	462
2.5.7.4	(METHOD MOUSE-GESTURE-ITEM-LIST VEHICLE APPEND).....	463
2.5.7.5	(METHOD IMMEDIATE-INTERVENTION-CHOICES VEHICLE).....	463
2.5.7.6	(METHOD REINIT VEHICLE).....	463

	2.5.7.7	VEHICLE .....	464
2.5.8		CSU simnet-objects>vehicle-tracking.lisp.....	464
	2.5.8.1	MAP-OVER-ALL-VEHICLES.....	465
	2.5.8.2	MAP-OVER-ALL-VEHICLE-HOLDERS .....	465
	2.5.8.3	GET-VEHICLE-HOLDER.....	465
	2.5.8.4	GET-VEHICLE .....	466
	2.5.8.5	PAINTED-P .....	467
	2.5.8.6	MAP-OVER-ALL-VEHICLES.....	468
	2.5.8.7	MAP-PREDICATE-OVER-VEHICLES .....	468
	2.5.8.8	GET-PREDICATE-ARGS.....	468
	2.5.8.9	MAP-OVER-ALL-VEHICLE-HOLDERS .....	469
	2.5.8.10	*TOP-LEVEL-UNITS* .....	469
	2.5.8.11	'(TOP-LEVEL-UNITS LOCAL REMOTE) .....	470
	2.5.8.12	TOP-LEVEL-UNITS .....	470
	2.5.8.13	TOP-LEVEL-UNIT-P .....	470
	2.5.8.14	CLEAR-TOP-LEVEL-UNITS.....	471
	2.5.8.15	REMOVE-TOP-LEVEL-UNIT.....	471
	2.5.8.16	ADD-TOP-LEVEL-UNIT.....	471
	2.5.8.17	MOVE-TOP-LEVEL-UNIT-TO-FRONT .....	472
	2.5.8.18	MOVE-TOP-LEVEL-UNIT-TO-BACK.....	472
	2.5.8.19	LOCAL-TOP-LEVEL-UNIT-POSITION .....	472
	2.5.8.20	INSERT-LOCAL-TOP-LEVEL-UNIT.....	473
	2.5.8.21	MOVE-TOP-LEVEL-UNIT-UP.....	473
	2.5.8.22	MOVE-TOP-LEVEL-UNIT-DOWN.....	473
	2.5.8.23	ALL-CHILDREN .....	474
	2.5.8.24	ALL-LOCAL-VEHICLES .....	474
	2.5.8.25	HANDLE-NAN-ERROR .....	475
	2.5.8.26	REDRAW-VEHICLES.....	475
	2.5.8.27	ERASE-ALL-VEHICLES.....	476
	2.5.8.28	*DISPLAY-UNIT-GRAPH-DELAY* .....	476
	2.5.8.29	DELAYED-DISPLAY-UNIT-GRAPH.....	476
	2.5.8.30	DELAYED-DISPLAY-UNIT-GRAPH-1 .....	477
	2.5.8.31	MAKE-AGENT .....	477
2.5.9		CSU objects>grapher-node.lisp.....	478
	2.5.9.1	GRAPHER-NODE.....	479
	2.5.9.2	GRAPHER-NODE.....	479

2.5.9.3	(METHOD GRAPHER-NODE-INFERIOR-NODES GRAPHER-NODE).....	479
2.5.9.4	(METHOD GRAPHER-NODE-DRAW GRAPHER-NODE).....	480
2.5.9.5	GRAPHER-NODE.....	480
2.5.10	CSU objects>object-grapher.lisp .....	480
2.5.10.1	OBJECT-COMPONENTS.....	481
2.5.10.2	PRESENT-OBJECT.....	481
2.5.10.3	OBJECT-DEPENDENTS.....	481
2.5.10.4	MAKE-OBJECT-GRAPHER-NODE .....	481
2.5.10.5	CLEAR-ALL-GRAPH-NODES.....	482
2.5.10.6	GET-PARENTLESS-OBJECTS.....	482
2.5.10.7	OBJECT-GRAPHER.....	482
2.5.10.8	(METHOD REDISPLAY-GRAPH OBJECT-GRAPHER).....	483
2.5.10.9	OBJECT-GRAPHER-NODE.....	483
2.5.10.10	GRAPHER-NODE-TO-FLAVOR-NAME.....	483
2.5.10.11	OBJECT-GRAPHER-NODE.....	483
2.5.10.12	(COM-GRAPH-OBJECTS MENU-ACCELERATOR Graph Objects).....	484
2.5.10.13	(COM-TOGGLE-INFERIOR-VISIBILITY) .....	484
2.5.10.14	TOGGLE-THIS-NODE.....	484
2.5.10.15	EDIT-OBJECT .....	484
2.6	SAF COMMAND PROTOCOL INTERFACE CSC.....	485
2.6.1	SAF Command Protocol (CP) CSC.....	485
2.6.1.1	CSU network>defstorage.lisp.....	486
2.6.1.1.1	(PROPERTY NET-FLOAT DEFSTORAGE-PROCESSOR).....	486
2.6.1.1.2	(PROPERTY NET-SHORT DEFSTORAGE-PROCESSOR).....	486
2.6.1.1.3	(PROPERTY NET-INT DEFSTORAGE-PROCESSOR).....	487
2.6.1.1.4	(PROPERTY NET-INT DEFSTORAGE-DESCRIBE) .....	487
2.6.1.1.5	DEFSTORAGE-STORE-NET-CHAR-SUBSTRING .....	487
2.6.1.1.6	DEFSTORAGE-MAKE-NET-CHAR-SUBSTRING .....	489
2.6.1.1.7	(PROPERTY NET-CHAR DEFSTORAGE-PROCESSOR).....	491

	2.6.1.1.8	AREF-4-BYTES .....	491
	2.6.1.1.9	(PROPERTY NET-DOUBLE DEFSTORAGE-PROCESSOR).....	492
2.6.1.2	CSU network	>packet-layouts.lisp .....	492
	2.6.1.2.1	NET-FLOAT.....	493
	2.6.1.2.2	NET-DOUBLE.....	493
	2.6.1.2.3	NET-INT.....	493
	2.6.1.2.4	NET-SHORT.....	493
	2.6.1.2.5	OPFOR-HEADER .....	493
	2.6.1.2.6	RUDP-HDR .....	494
	2.6.1.2.7	BURST-DESC .....	494
	2.6.1.2.8	CREATION.....	494
	2.6.1.2.9	MINEFIELD-CREATION.....	494
	2.6.1.2.10	VEHICLE-STATUS.....	495
	2.6.1.2.11	POSITION-DESCRIPTOR .....	495
	2.6.1.2.12	WHERE-ARE-THEY.....	495
	2.6.1.2.13	GROUND-IMPACT.....	495
	2.6.1.2.14	VEHICLE-IMPACT.....	495
	2.6.1.2.15	INDIRECT-FIRE .....	496
	2.6.1.2.16	INTERVISIBILITY.....	496
	2.6.1.2.17	NOTIFY.....	496
	2.6.1.2.18	VEHICLE-DEATH .....	496
	2.6.1.2.19	POSITION-DESC.....	497
	2.6.1.2.20	CREATE-REQUEST.....	497
	2.6.1.2.21	RESET-REQUEST .....	497
	2.6.1.2.22	ARTY-REQUEST.....	497
	2.6.1.2.23	READ-CONFIG-REQUEST .....	497
	2.6.1.2.24	ATTACH-REQUEST.....	498
	2.6.1.2.25	DETACH-REQUEST.....	498
	2.6.1.2.26	POLL-REQUEST.....	498
	2.6.1.2.27	MINEFIELD-REQUEST.....	498
	2.6.1.2.28	RESUPPLY-REQUEST .....	499
	2.6.1.2.29	TELEPORT-REQUEST.....	499
	2.6.1.2.30	TARGETING-REQUEST .....	499
	2.6.1.2.31	MACHINE-STATUS.....	499
	2.6.1.2.32	DISCONNECT-REQUEST.....	499

2.6.1.2.33	QUERY-SUB-STATE- REQUEST .....	500
2.6.1.2.34	STATUS-REPORT .....	500
2.6.1.2.35	SUB-STATE.....	500
2.6.1.2.36	IVIS-CONTACT .....	500
2.6.1.2.37	IVIS-SPOT.....	501
2.6.1.2.38	IVIS-SHELL.....	501
2.6.1.2.39	IVIS-CONTROL-REQUEST.....	501
2.6.1.2.40	IVIS-FINE-CONTROL- REQUEST .....	501
2.6.1.2.41	CONTINUE-MISSION- REQUEST .....	501
2.6.1.2.42	XYPOINT.....	502
2.6.1.2.43	ASSIGN-ROUTE-REQUEST.....	502
2.6.1.2.44	VEHICLE-POSITION- DESCRIPTOR.....	502
2.6.1.2.45	VEHICLE-APPEARANCE- DESCRIPTOR.....	502
2.6.1.2.46	VEHICLE-ECHELON- DESCRIPTOR.....	503
2.6.1.2.47	VEHICLE-PAE.....	503
2.6.1.2.48	VEHICLE-POSITION.....	503
2.6.1.2.49	VEHICLE-POSITION-POLL- COMPLETED .....	503
2.6.1.2.50	VEHICLE-APPEARANCE.....	503
2.6.1.2.51	VEHICLE-ECHELON .....	504
2.6.1.2.52	GENERIC-MESSAGE.....	504
2.6.1.2.53	CM-ID .....	504
2.6.1.2.54	CM-POINT-LIST .....	504
2.6.1.2.55	POINT-REQUEST.....	505
2.6.1.2.56	AREA-REQUEST.....	505
2.6.1.2.57	ZONE-REQUEST.....	505
2.6.1.2.58	LINE-REQUEST.....	505
2.6.1.2.59	ROUTE-PT .....	505
2.6.1.2.60	ROUTE-REQUEST.....	506
2.6.1.2.61	DELETE-OVERLAY-REQUEST .....	506
2.6.1.2.62	EXECUTE-OVERLAY- REQUEST .....	506
2.6.1.2.63	DELETE-CM-REQUEST .....	506



2.6.1.2.64	HALT-REQUEST.....	507
2.6.1.2.65	RESUME-REQUEST .....	507
2.6.1.2.66	HOLD-REQUEST .....	507
2.6.1.2.67	CHANGE-SPEED-REQUEST.....	507
2.6.1.2.68	CHANGE-ALTITUDE- REQUEST .....	507
2.6.1.2.69	CHANGE-FORMATION- REQUEST .....	508
2.6.1.2.70	FOLLOW-VEHICLE-REQUEST.....	508
2.6.1.2.71	SIMULATOR-IN-COMMAND- REQUEST .....	508
2.6.1.2.72	GO-TO-POINT-REQUEST.....	508
2.6.1.2.73	LAND-REQUEST .....	509
2.6.1.2.74	RESUME-MISSION-REQUEST.....	509
2.6.1.2.75	FACE-DIRECTION-REQUEST .....	509
2.6.1.2.76	ENROUTE-MOVEMENT- REQUEST .....	509
2.6.1.2.77	STEALTH-POS .....	509
2.6.1.2.78	ATTACH-STEALTH-REQUEST.....	510
2.6.1.2.79	REJOIN-UNIT-REQUEST.....	510
2.6.1.2.80	ATTACK-REQUEST.....	510
2.6.1.2.81	VEHICLE-LOAD .....	510
2.6.1.2.82	VEHICLE-REINIT-REQUEST.....	511
2.6.1.2.83	CHECK-STATION-REQUEST.....	511
2.6.2.	Connection CSC .....	511
2.6.2.1	CSU network>connection.lisp .....	511
2.6.2.1.1	SIMNET.....	511
2.6.2.1.2	*TARGET-NUMBER-OF- WIRED-PACKET-BUFFERS*.....	512
2.6.2.1.3	INIT-CONN .....	512
2.6.2.1.4	INITIALIZE-CONNECTION .....	512
2.6.2.1.5	SPECIFY-SIMNET-PORT .....	513
2.6.2.1.6	INIT-CONN-1.....	513
2.6.2.1.7	BUSY-WAIT-ON-CONN .....	514
2.6.2.1.8	CONN-P .....	514
2.6.2.1.9	STANDALONEP .....	514
2.6.2.1.10	EXIT-CONN .....	515
2.6.2.1.11	UI-EXIT-CONNECTION .....	515

2.6.2.2	CSU network>ip-tcp-patch.lisp .....	516
2.6.2.2.1	(METHOD GIVE-BACK-BUFFERS UDP-CONN) .....	516
2.6.2.2.2	INHIBIT-FDEFINE-WARNINGS .....	516
2.6.2.2.3	(METHOD PACKET-BUFFER-PANIC UDP-PROTOCOL).....	516
2.6.2.2.4	INHIBIT-FDEFINE-WARNINGS .....	516
2.6.2.3	CSU network>vars.lisp .....	517
2.6.2.3.1	GET-HOSTS-WITH-SIMNET-SERVICE .....	517
2.6.2.3.2	GET-LOCAL-HOST-SAF-PORT.....	517
2.6.2.3.3	COM-SHOW-SAF-PORT .....	518
2.6.2.3.4	COM-SAF-SHOW-PORT .....	518
2.6.2.3.5	SIMULATION-HOST.....	518
2.6.2.3.6	RETRANSMIT_PERIOD.....	519
2.6.2.3.7	TRANSMIT_WARNING_LENGTH.....	519
2.6.2.3.8	MISSION-CONTROL-AWAIT.....	519
2.6.2.3.9	MISSION-CONTROL-NOTIFY.....	519
2.6.2.3.10	MISSION-CONTROL-IMMEDIATE .....	520
2.6.2.3.11	MISSION-CONTROL-ABORT.....	520
2.6.2.3.12	MISSION-CONTROL-NODISTRIBUTE .....	520
2.6.2.3.13	CREATION.....	520
2.6.2.3.14	WHERE-ARE-THEY.....	520
2.6.2.3.15	GROUND-IMPACT.....	521
2.6.2.3.16	VEHICLE-IMPACT .....	521
2.6.2.3.17	INTERVISIBILITY.....	521
2.6.2.3.18	NOTIFY.....	521
2.6.2.3.19	VEHICLE-STATUS.....	522
2.6.2.3.20	INDIRECT-FIRE .....	522
2.6.2.3.21	REGISTER .....	522
2.6.2.3.22	UNREGISTER.....	523
2.6.2.3.23	ACTIVITY-COMPLETE.....	523
2.6.2.3.24	RADIO-STATUS .....	523

2.6.2.3.25	RADIO-MESSAGE.....	523
2.6.2.3.26	MACHINE-STATUS.....	523
2.6.2.3.27	MINEFIELD-CREATION.....	524
2.6.2.3.28	SUB-STATE.....	524
2.6.2.3.29	IVIS-CONTACT.....	524
2.6.2.3.30	IVIS-SPOT.....	524
2.6.2.3.31	IVIS-SHELL.....	525
2.6.2.3.32	VEHICLE-PAE.....	525
2.6.2.3.33	VEHICLE-POSITION.....	525
2.6.2.3.34	VEHICLE-POSITION-POLL- COMPLETED.....	525
2.6.2.3.35	VEHICLE-APPEARANCE.....	525
2.6.2.3.36	VEHICLE-ECHELON.....	526
2.6.2.3.37	GENERIC-MESSAGE.....	526
2.6.2.3.38	STEALTH-POS.....	526
2.6.2.3.39	VEHICLE-LOAD.....	526
2.6.2.3.40	CREATE.....	527
2.6.2.3.41	RESET.....	527
2.6.2.3.42	ARTY.....	527
2.6.2.3.43	READ-CONFIG.....	527
2.6.2.3.44	VEHICLE-REINT.....	528
2.6.2.3.45	RESUME.....	528
2.6.2.3.46	RESUPPLY.....	528
2.6.2.3.47	ATTACH.....	528
2.6.2.3.48	DETACH.....	529
2.6.2.3.49	TELEPORT.....	529
2.6.2.3.50	READ-ACTIVITIES.....	529
2.6.2.3.51	START-ACTIVITY.....	529
2.6.2.3.52	POLL.....	529
2.6.2.3.53	RADIO-COMMAND.....	530
2.6.2.3.54	MINEFIELD.....	530
2.6.2.3.55	DISCONNECT.....	530
2.6.2.3.56	QUERY-SUB-STATE.....	530
2.6.2.3.57	IVIS-CONTROL.....	531
2.6.2.3.58	IVIS-FINE-CONTROL.....	531
2.6.2.3.59	CONTINUE-MISSION.....	531

2.6.2.3.60	ASSIGN-ROUTE.....	532
2.6.2.3.61	POINT.....	532
2.6.2.3.62	AREA.....	535
2.6.2.3.63	ZONE.....	535
2.6.2.3.64	LINE.....	536
2.6.2.3.65	ROUTE.....	536
2.6.2.3.66	DELETE-OVERLAY .....	537
2.6.2.3.67	EXECUTE-OVERLAY.....	538
2.6.2.3.68	HALT.....	538
2.6.2.3.69	CHANGE-SPEED .....	538
2.6.2.3.70	CHANGE-FORMATION.....	539
2.6.2.3.71	DELETE-CM .....	539
2.6.2.3.72	FOLLOW-VEHICLE .....	540
2.6.2.3.73	GO-TO-POINT.....	540
2.6.2.3.74	RESUME-MISSION .....	540
2.6.2.3.75	FACE-DIRECTION .....	540
2.6.2.3.76	TARGETING.....	541
2.6.2.3.77	ATTACH-STEALTH.....	541
2.6.2.3.78	HOLD.....	541
2.6.2.3.79	CHANGE-ALTITUDE.....	542
2.6.2.3.80	ENROUTE-MOVEMENT .....	542
2.6.2.3.81	SIMULATOR-IN-COMMAND .....	542
2.6.2.3.82	REJOIN-UNIT .....	542
2.6.2.3.83	LAND.....	543
2.6.2.3.84	ATTACK.....	543
2.6.2.3.85	CHECK-STATION.....	543
2.6.2.3.86	RUNNING-FIRE-ATTACK.....	543
2.6.2.3.87	POP-UP-ATTACK .....	544
2.6.2.3.88	WHERE-ARE-THEY-REQUEST .....	544
2.6.2.3.89	VEHICLE-STATUS-REQUEST.....	544
2.6.2.3.90	TACTICS-NATO .....	544
2.6.2.3.91	TACTICS-WARSAW .....	545
2.6.2.3.92	OPFOR .....	545
2.6.2.3.93	BLUEFOR .....	546
2.6.2.3.94	RESET-ALL-VEHICLES.....	546
2.6.2.3.95	ARTY-TYPE-GROUND.....	546

2.6.2.3.96	ARTY-TYPE-VEHICLE .....	547
2.6.2.3.97	ARTY-TYPE-DEATH.....	547
2.6.2.3.98	READ-FORMATIONS.....	547
2.6.2.3.99	READ-VEHICLE-PARMS .....	547
2.6.2.3.100	READ-UNIT-CONFIG .....	548
2.6.2.3.101	READ-HITMODELS .....	548
2.6.2.3.102	READ-DAMAGES.....	548
2.6.2.3.103	READ-DETECTIONS.....	548
2.6.2.3.104	RESUPPLY-TYPE-FUEL.....	548
2.6.2.3.105	RESUPPLY-TYPE-AMMO .....	549
2.6.2.3.106	HOLD_FIRE.....	549
2.6.2.3.107	FIRE_AT_WILL .....	549
2.6.2.3.108	FIRE_AT_POSITION.....	549
2.6.2.3.109	FIRE_AT_WHAT_LEADER_ SHOOTS .....	550
2.6.2.3.110	FIRE_AT_DESIGNATED_ TARGETS .....	550
2.6.2.3.111	MAX-WEAPONS.....	550
2.6.2.3.112	*PRETTY-WEAPON-TABLE*.....	550
2.6.2.3.113	DEFINE-SIMNET-WEAPON.....	551
2.6.2.3.114	WEAPON-105MM.....	551
2.6.2.3.115	WEAPON-25MM.....	551
2.6.2.3.116	WEAPON-SAGGER .....	551
2.6.2.3.117	WEAPON-SPIRAL.....	552
2.6.2.3.118	WEAPON-ROCKET .....	552
2.6.2.3.119	WEAPON-BOMB.....	552
2.6.2.3.120	WEAPON-ADA-MISSILE.....	552
2.6.2.3.121	MAX-VEH-TYPES.....	553
2.6.2.3.122	VEH-SPECIAL.....	553
2.6.2.3.123	VEH-MAIN-BATTLE-TANK.....	553
2.6.2.3.124	VEH-PERSONNEL-CARRIER.....	553
2.6.2.3.125	VEH-COMMAND-POST .....	554
2.6.2.3.126	VEH-AMMUNITION-TRUCK.....	554
2.6.2.3.127	VEH-FUEL-TRUCK.....	554
2.6.2.3.128	VEH-SUPPLY-TRUCK .....	554
2.6.2.3.129	VEH-MORTAR-CARRIER.....	555

2.6.2.3.130	VEH-SP-HOWITZER.....	555
2.6.2.3.131	VEH-RECOVERY-VEHICLE.....	555
2.6.2.3.132	VEH-FIST-VEHICLE.....	555
2.6.2.3.133	VEH-ATTACK-HELICOPTER.....	556
2.6.2.3.134	VEH-SCOUT-HELICOPTER.....	556
2.6.2.3.135	VEH-FIGHTER-BOMBER-A.....	556
2.6.2.3.136	VEH-FIGHTER-BOMBER.....	556
2.6.2.3.137	VEH-SMOKE-CLOUD .....	557
2.6.2.3.138	VEH-ANTI-AIRCRAFT .....	557
2.6.2.3.139	VEH-TANKER-AIRCRAFT .....	557
2.6.2.3.140	VEH-AWACS-AIRCRAFT.....	558
2.6.2.3.141	VEH-FIGHTER-BOMBER-B.....	558
2.6.2.3.142	VEH-FIGHTER-BOMBER-C.....	558
2.6.2.3.143	VEH-FIGHTER-BOMBER-D.....	558
2.6.2.3.144	VEH-INTERCEPTOR.....	558
2.6.2.3.145	VEH-MISSILE .....	559
2.6.2.3.146	*PRETTY-TYPE-TABLE* .....	559
2.6.2.3.147	(AREF *PRETTY-TYPE- TABLE* 0).....	559
2.6.2.3.148	HEAT-25.....	559
2.6.2.3.149	HEAT-105 .....	560
2.6.2.3.150	SABOT-25 .....	560
2.6.2.3.151	SABOT-105.....	560
2.6.2.3.152	TOW-2K .....	560
2.6.2.3.153	FAAD-MISSILE .....	560
2.6.2.3.154	HELLFIRE-MISSILE .....	561
2.6.2.3.155	MAVERICK-MISSILE.....	561
2.6.2.3.156	DRAGON-MISSILE.....	561
2.6.2.3.157	BOMB500.....	561
2.6.2.3.158	HE107 .....	562
2.6.2.3.159	HE155 .....	562
2.6.2.3.160	WP107.....	562
2.6.2.3.161	FUZE-POINT-DETONATING .....	562
2.6.2.3.162	FUZE-PROXIMITY.....	563
2.6.2.3.163	LOCAL .....	563
2.6.2.3.164	REMOTE.....	564

	2.6.2.3.165	VEH-TARGET-PERSON.....	564
	2.6.2.3.166	VEH-TARGET-VEH.....	564
	2.6.2.3.167	VEH-SAFETY-FAN-L.....	565
	2.6.2.3.168	VEH-SAFETY-FAN-R.....	565
	2.6.2.3.169	VEH-TARGET-BORE .....	565
	2.6.2.3.170	INDIRECT-FIRE-BURST- HEIGHT .....	565
	2.6.2.3.171	*RELATIVE-DISPLAY* .....	565
	2.6.2.3.172	UNHANDLED-MESSAGE- HALT.....	566
	2.6.2.3.173	PRINT-OUTPUT-COMMANDS .....	566
	2.6.2.3.174	*WAITING-FOR-RESET*.....	566
2.6.2.4		CSU network>top-level.lisp.....	566
	2.6.2.4.1	GET-OPFOR-SUB-PACKET.....	567
	2.6.2.4.2	COMPLETE-C2-RESET.....	567
	2.6.2.4.3	*RESET-WAIT-LIMIT* .....	568
	2.6.2.4.4	RESET-SIM .....	568
2.6.3		Reliable Universal Datagram Protocol (RUDP) CSC.....	569
2.6.3.1		CSU rudp>vars.lisp .....	569
	2.6.3.1.1	*RUDP-AREA*.....	569
	2.6.3.1.2	*SIM-CONN*.....	570
	2.6.3.1.3	*SERVICE-ACCESS-PATH* .....	572
	2.6.3.1.4	*RUDP-TYPE-SYNCH* .....	572
	2.6.3.1.5	*RUDP-TYPE-DATA*.....	572
	2.6.3.1.6	*RUDP-TYPE-ACK* .....	573
	2.6.3.1.7	*PKT*.....	573
	2.6.3.1.8	*PKT-START*.....	575
	2.6.3.1.9	*PKT-END*.....	577
	2.6.3.1.10	*PKT-PTR* .....	579
	2.6.3.1.11	*RUDP-PACKETS- PROCESSED* .....	581
	2.6.3.1.12	*LAST-SEQUENCE-IN*.....	581
	2.6.3.1.13	*NEXT-SEQUENCE-OUT* .....	583
	2.6.3.1.14	*ACK-NEEDED*.....	585
	2.6.3.1.15	*RETRANSMIT-TIMER*.....	587
	2.6.3.1.16	*RETRANSMIT-QUEUE*.....	587
	2.6.3.1.17	*RUDP-OUTPUT-STREAM* .....	588

2.6.3.1.18	*RUDP-OUTPUT-STREAM* .....	588
2.6.3.1.19	*PACKET-REQUEST-QUEUE* .....	589
2.6.3.1.20	*PACKET-IMMEDIATE- QUEUE* .....	589
2.6.3.1.21	*RUDP-RECEIVE-QUEUE* .....	590
2.6.3.1.22	*LAST-PACKET-IN-TIME* .....	590
2.6.3.1.23	*LAST-PACKET-IN- WARNING-STATE* .....	591
2.6.3.1.24	CVV-PRINT-60THS .....	591
2.6.3.1.25	CVV-READ-60THS .....	591
2.6.3.1.26	(GET '60THS 'CHOOSE- VARIABLE-VALUES- KEYWORD).....	591
2.6.3.1.27	*RUDP-OPTIONS* .....	592
2.6.3.1.28	*LAST-PACKET-IN- SHUTDOWN-STATE* .....	592
2.6.3.1.29	*BARE-ACK-PERIOD* .....	592
2.6.3.1.30	*RETRANSMIT-PERIOD* .....	592
2.6.3.1.31	*TRANSMIT-QUEUE- WARNING-LENGTH* .....	593
2.6.3.1.32	*TRANSMIT-QUEUE-ERROR- LENGTH* .....	593
2.6.3.1.33	*MAX-RECEIVE-QUEUE- LENGTH* .....	593
2.6.3.1.34	*DEBUG-RUDP* .....	593
2.6.3.1.35	*IVIS-OPTIONS* .....	594
2.6.3.1.36	*IVIS-OPTIONS* .....	594
2.6.3.1.37	*REAPPEAR-LATENCY* .....	594
2.6.3.1.38	*RANGE-THRESHOLD* .....	595
2.6.3.1.39	*UPDATE-RATE* .....	595
2.6.3.1.40	*CLUSTER-DISTANCE* .....	595
2.6.3.2	CSU rudp>outgoing.lisp.....	596
2.6.3.2.1	PREPEND-RUDP-HEADER.....	596
2.6.3.2.2	TRANSMIT-MSG .....	598
2.6.3.2.3	GET-RUDP-BUFFER .....	600
2.6.3.2.4	FLUSH-RUDP-RETRANSMIT- BUFFERS .....	600
2.6.3.2.5	FLUSH-RUDP-PENDING- TRANSMIT-BUFFERS .....	601



	2.6.3.2.6	DO-ALL-QUEUED-REQUESTS.....	601
	2.6.3.2.7	PROCESS-OUTGOING-RUDP.....	601
	2.6.3.2.8	'(NET-MSG REQUEST IMMEDIATE).....	602
	2.6.3.2.9	NET-MSG .....	602
	2.6.3.2.10	PUT-MSG-IN-RETRANSMIT-QUEUE.....	603
	2.6.3.2.11	DEQUEUE-OUTGOING.....	606
	2.6.3.2.12	CHECK-FOR-RETRANSMIT-OR-ACK .....	606
	2.6.3.2.13	TRANSMIT-SYNCH .....	607
	2.6.3.2.14	TRANSMIT-ACK .....	608
	2.6.3.2.15	RETRANSMIT-QUEUED-PACKET .....	609
	2.6.3.2.16	RETRANSMIT-ALL-QUEUED-PACKETS .....	610
2.6.3.3		CSU rudp>incoming.lisp .....	610
	2.6.3.3.1	FLUSH-RUDP-RECEIVE-BUFFERS .....	611
	2.6.3.3.2	PROCESS-INCOMING-RUDP .....	611
	2.6.3.3.3	POLL-FOR-MESSAGES .....	611
	2.6.3.3.4	PROCESS-INCOMING-RUDP-PACKET .....	612
	2.6.3.3.5	PROCESS-RECEIVED-PACKETS .....	613
	2.6.3.3.6	PROCESS-SIM-PKT.....	613
	2.6.3.3.7	PROCESS-ALL-MSGS-IN-UDP-PKT.....	613
2.6.3.4		CSU rudp>utils.lisp .....	614
	2.6.3.4.1	RUDP-PACKET .....	614
	2.6.3.4.2	GET-RUDP-PACKET .....	614
	2.6.3.4.3	FREE-RUDP-PACKET.....	615
	2.6.3.4.4	RETRANSMIT-QUEUE-ITEM.....	615
	2.6.3.4.5	MAKE-RETRANSMIT-QUEUE-ITEM.....	616
	2.6.3.4.6	RECEIVE-QUEUE-ITEM .....	616
	2.6.3.4.7	MAKE-RECEIVE-QUEUE-ITEM....	616
	2.6.3.4.8	DEBUG-RUDP .....	617
	2.6.3.4.9	RUDP-TRANSMIT-AND-RECEIVE .....	617

	2.6.3.4.10	SIGNAL-RUDP-ERROR.....	617
	2.6.3.4.11	FLUSH-ALL-RUDP-BUFFERS .....	618
2.6.4.	SAF Command Layer CSC.....		618
2.6.4.1	CSU rudp>handle-incoming.lisp.....		619
	2.6.4.1.1	*NUMBER-OF-PACKET-TYPES* .....	619
	2.6.4.1.2	*PACKET-HANDLER-FUNCTION-TABLE* .....	619
	2.6.4.1.3	*PACKET-PRINT-FUNCTION-TABLE* .....	620
	2.6.4.1.4	*PACKET-OPTIONS* .....	621
	2.6.4.1.5	*PRINT-MESSAGES* .....	621
	2.6.4.1.6	SET-HANDLER-FUNCTION .....	622
	2.6.4.1.7	LOOKUP-HANDLER-FUNCTION .....	622
	2.6.4.1.8	SET-PRINT-FUNCTION .....	622
	2.6.4.1.9	LOOKUP-PRINT-FUNCTION.....	623
	2.6.4.1.10	(COMPILE LOAD EVAL) .....	623
	2.6.4.1.11	DEF-PACKET-HANDLER.....	623
	2.6.4.1.12	VEHICLE-IMPACT .....	624
	2.6.4.1.13	GROUND-IMPACT.....	624
	2.6.4.1.14	INDIRECT-FIRE .....	624
	2.6.4.1.15	RESET.....	624
	2.6.4.1.16	MACHINE-STATUS.....	625
	2.6.4.1.17	MINEFIELD-CREATION .....	625
	2.6.4.1.18	SUB-STATE.....	625
	2.6.4.1.19	IVIS-CONTACT.....	625
	2.6.4.1.20	IVIS-SPOT.....	625
	2.6.4.1.21	IVIS-SHELL.....	626
	2.6.4.1.22	VEHICLE-POSITION.....	626
	2.6.4.1.23	VEHICLE-POSITION-POLL-COMPLETED .....	626
	2.6.4.1.24	VEHICLE-APPEARANCE.....	626
	2.6.4.1.25	VEHICLE-ECHELON .....	627
	2.6.4.1.26	VEHICLE-PAE.....	627
	2.6.4.1.27	GENERIC-RADIO-MESSAGE.....	627
	2.6.4.1.28	GENERIC-ERROR-MESSAGE .....	627
	2.6.4.1.29	GENERIC-BEEP-MESSAGE.....	628

	2.6.4.1.30	PRINT-MESSAGE .....	628
	2.6.4.1.31	GENERIC-MESSAGE.....	628
	2.6.4.1.32	*OLD-STEALTH- PARAMETERS* .....	628
	2.6.4.1.33	STEALTH-POS .....	629
	2.6.4.1.34	VEHICLE-LOAD .....	629
2.6.4.2		CSU network>commands.lisp .....	629
	2.6.4.2.1	DEFSEND .....	629
	2.6.4.2.2	SEND-CREATE.....	630
	2.6.4.2.3	SEND-TARGETING.....	631
	2.6.4.2.4	SEND-POLL.....	631
	2.6.4.2.5	SEND-MINEFIELD .....	632
	2.6.4.2.6	SEND-RESET.....	633
	2.6.4.2.7	SEND-ARTY .....	633
	2.6.4.2.8	SEND-READ-CONFIG.....	634
	2.6.4.2.9	SEND-ATTACH .....	635
	2.6.4.2.10	SEND-DETACH .....	635
	2.6.4.2.11	SEND-RESUPPLY.....	636
	2.6.4.2.12	SEND-TELEPORT .....	637
	2.6.4.2.13	MAKE-ORTHOGONAL-LIST.....	637
	2.6.4.2.14	*BOMBS-PER-PACKET* .....	638
	2.6.4.2.15	*AMMO-TYPE* .....	638
	2.6.4.2.16	*FUZE-TYPE* .....	638
	2.6.4.2.17	*ARTY-TYPE* .....	639
	2.6.4.2.18	*ARTY-SPREAD* .....	639
	2.6.4.2.19	BOMB-BUTTON .....	639
	2.6.4.2.20	SEND-DISCONNECT .....	640
	2.6.4.2.21	SET-BOMB-PARAMETERS.....	641
	2.6.4.2.22	SEND-QUERY-SUB-STATE.....	641
	2.6.4.2.23	SEND-IVIS-CONTROL .....	642
	2.6.4.2.24	SEND-IVIS-MESSAGES.....	643
	2.6.4.2.25	SEND-IVIS-FINE-CONTROL.....	643
	2.6.4.2.26	SEND-AN-IVIS-FINE- CONTROL-PACKET .....	644
	2.6.4.2.27	SEND-AN-IVIS-FINE- CONTROL.....	644
	2.6.4.2.28	SEND-CONTINUE-MISSION .....	644

2.6.4.2.29	SEND-ASSIGN-ROUTE .....	645
2.6.4.2.30	SEND-POINT .....	646
2.6.4.2.31	SEND-AREA .....	647
2.6.4.2.32	SEND-ZONE .....	648
2.6.4.2.33	SEND-LINE .....	648
2.6.4.2.34	SEND-ROUTE .....	649
2.6.4.2.35	SEND-DELETE-OVERLAY.....	650
2.6.4.2.36	SEND-EXECUTE-OVERLAY.....	651
2.6.4.2.37	SEND-DELETE-CM.....	652
2.6.4.2.38	SEND-HALT .....	652
2.6.4.2.39	SEND-RESUME .....	653
2.6.4.2.40	SEND-HOLD.....	654
2.6.4.2.41	SEND-CHANGE-SPEED .....	654
2.6.4.2.42	SEND-CHANGE-ALTITUDE .....	655
2.6.4.2.43	SEND-CHANGE-FORMATION .....	656
2.6.4.2.44	SEND-FOLLOW-VEHICLE .....	656
2.6.4.2.45	SEND-SIMULATOR-IN- COMMAND .....	657
2.6.4.2.46	SEND-GO-TO-POINT .....	658
2.6.4.2.47	SEND-LAND.....	658
2.6.4.2.48	SEND-RESUME-MISSION.....	659
2.6.4.2.49	SEND-FACE-DIRECTION.....	660
2.6.4.2.50	SEND-ENROUTE-MOVEMENT.....	660
2.6.4.2.51	SEND-ATTACH-STEALTH.....	661
2.6.4.2.52	ATTACH-STEALTH.....	662
2.6.4.2.53	SEND-REJOIN-UNIT .....	662
2.6.4.2.54	SEND-ATTACK .....	663
2.6.4.2.55	SEND-VEHICLE-REINIT.....	663
2.6.4.2.56	SEND-CHECK-STATION .....	664
2.7	GLOBALS CSC.....	665
2.7.1	CSU sys>constants.lisp.....	665
2.7.1.1	$\pi$ .....	665
2.7.1.2	$2\pi$ .....	666
2.7.1.3	180DEG .....	667
2.7.1.4	3200MIL .....	667
2.7.1.5	-180DEG .....	667

2.7.1.6	-3200MIL .....	667
2.7.1.7	90DEG.....	668
2.7.1.8	1600MIL .....	668
2.7.1.9	-90DEG.....	668
2.7.1.10	-1600MIL .....	669
2.7.1.11	360DEG .....	669
2.7.1.12	6400MIL .....	669
2.7.1.13	RAD-TO-DEG.....	669
2.7.1.14	DEG-TO-RAD.....	670
2.7.1.15	RAD-TO-MIL.....	670
2.7.1.16	MIL-TO-RAD.....	670
2.7.1.17	HALFPI.....	671
2.7.1.18	5-DEG.....	671
2.7.1.19	DEG-TO-MIL.....	671
2.7.1.20	UNKNOWN-HEADING.....	671
2.7.1.21	RUDP_TYPE_SYNCH.....	672
2.7.1.22	RUDP_TYPE_DATA.....	672
2.7.1.23	RUDP_TYPE_ACK.....	674
2.7.1.24	TEAM-NATO.....	674
2.7.1.25	TEAM-WARSAW-PACT.....	674
2.7.1.26	INVISIBLE.....	675
2.7.1.27	PARTLY_VISIBLE.....	675
2.7.1.28	VISIBLE .....	675
2.7.1.29	VEH-IMMOBILE.....	675
2.7.1.30	VEH-CANTFIRE.....	676
2.7.1.31	VEH-DESTROYED.....	676
2.7.1.32	VEH-OUT-OF-GAS.....	676
2.7.1.33	VEH-OUT-OF-AMMO.....	677
2.7.1.34	VEH-LANDED.....	677
2.7.1.35	VEH-RESUPPLYING.....	677
2.7.1.36	VEH-STUCK.....	677
2.7.1.37	MAX-VEHICLES.....	678
2.7.1.38	*MAX-VEHICLE-ID* .....	678
2.7.1.39	VEHICLE-ID-IRRELEVANT.....	678
2.7.2	CSU sys>vars.lisp.....	679
2.7.2.1	*SAF-INTERFACE-OPTIONS* .....	679

2.7.2.2	*SAF-DEBUG-OPTIONS*	679
2.7.2.3	*SAF-CONNECTION-OPTIONS*	679
2.7.2.4	*SAF-APPEARANCE-OPTIONS*	680
2.7.2.5	*DEFAULT-OUTPUT-COORDINATE- SYSTEM*	680
2.7.2.6	*SAF-INITIALIZATION-LIST*	680
2.7.2.7	*POLL-WHERE-ARE-THEY-FLAG*	680
2.7.2.8	*ALL-VEHICLES*	681
2.7.2.9	*VIEW-VEHICLE-ID*	681
2.7.2.10	ALIGNED-FOE	682
2.7.2.11	ALIGNED-OFFENSE	682
2.7.2.12	ALIGNED-DEFENSE	683
2.7.2.13	ALIGNED-FRIEND	684
2.7.2.14	ALIGNED-SCENARIO	684
2.7.2.15	ALIGNED-USSR	684
2.7.2.16	ALIGNED-US	685
2.7.2.17	ALIGNED-MIXED	685
2.7.2.18	COUNTRY-US	686
2.7.2.19	COUNTRY-USSR	686
2.7.2.20	DISTINGUISHED-FORCE	686
2.7.2.21	OTHER-FORCE	686
2.7.2.22	OBSERVER-FORCE	687
2.7.2.23	TARGET-FORCE	687
2.7.2.24	GODS-EYE-VIEW	687
2.7.2.25	NON-GODS-EYE-VIEW	687
2.7.2.26	COMMANDERS-EYE-VIEW	688
2.7.2.28	*FOE-ALLIANCE*	688
2.7.2.29	*TEAM*	689
2.7.2.30	*PVD-FRAME*	689
2.7.2.31	*PVD-DISPLAY*	689
2.7.2.32	*PVD-LEGEND*	692
2.7.2.33	*OFFENSE-ALU*	693
2.7.2.34	*DEFENSE-ALU*	693
2.7.2.35	*ERASE-EFFECTS-ALU*	693
2.7.2.36	*ERASE-VEHICLES-ALU*	694
2.7.2.37	*BOMB-EFFECTS-ALU*	694

2.7.2.38	*TRIM-ALU*	695
2.7.2.39	*WHITE-EFFECTS-ALU*	695
2.7.2.40	*YELLOW-EFFECTS-ALU*	695
2.7.2.41	*OPFOR-FRAME*	696
2.7.2.42	*OPFOR-IO*	697
2.7.2.43	*OPFOR-IO*	697
2.7.2.44	*RADIO-OUTPUT*	699
2.7.2.45	*BMI-PROGRAM*	699
2.7.2.46	*TERRAIN-OPTIONS*	700
2.7.2.47	*BATTLEFIELD-OBJECTS*	701
2.7.2.48	*BATTLEFIELD-OBJECTS*	701
2.7.2.49	*SANDBOX*	701
2.7.2.50	*ACTIVE-SANDBOXES*	701
2.7.2.51	*WHERE-ARE-THEY-POLL-WAIT*	702
2.7.2.52	*WHERE-ARE-THEY-POLL-FREQUENCY*	702
2.7.2.53	*WHERE-ARE-THEY-PAINT-FLAG*	702
2.7.2.54	*PAINT-VEHICLES-AS-ICONS*	703
2.7.2.55	*STOP-UPDATE-PROCESS*	703
2.7.2.56	*INTERFACE-TO-UPDATE-PROCESS- QUEUE*	703
2.7.2.57	*NETWORK-TO-UPDATE-PROCESS- QUEUE*	704
2.7.2.58	*BFLY-TIME-OFFSET*	704
2.7.2.59	*ETIME*	705
2.7.2.61	*DEFAULT-UNIT-GRAPH-DELAY*	705
2.7.2.62	*COS-ARRAY*	705
2.7.2.63	*COS-ARRAY-MAX-INDEX*	706
2.7.2.64	*DRIBBLE-FLG*	706
2.7.2.65	*EXTRA-INFO*	706
2.7.2.66	*NEW-INTERFACE-FLG*	706
2.7.2.67	*EFFECTS-QUEUE*	707
2.7.2.68	*TARGET-TYPES*	707
2.7.2.69	*DISPOSITIONS*	707
2.7.2.70	*IVIS-TO-SIMNET*	708
2.7.2.71	*IVIS-TO-SBX*	708
2.7.2.72	SCENARIO-COUNTER	708
2.7.2.73	*SBX-UNIQUE-UNIT-ID*	708

2.7.2.74	UNIQUE_ID_IRRELEVANT.....	709
2.7.2.76	NEW-SBX-UNIQUE-UNIT-ID.....	709
2.7.2.77	RESET-SBX-UNIQUE-UNIT-ID.....	709
2.7.2.78	*SANDBOX-OBJECTS-ALIST*.....	710
2.7.2.79	ADD-SANDBOX-TO-ALIST .....	710
2.7.2.80	CLEAR-SANDBOX-ALIST.....	710
2.7.2.81	RETURN-AND-REMOVE-SANDBOX-FROM- ALIST .....	710
2.7.2.82	*PRETTY-ALIGNMENT-TABLE*.....	711
2.7.2.83	(AREF *PRETTY-ALIGNMENT-TABLE* ALIGNED-FOE).....	711
2.7.2.84	*MY-CONCEIVED-UNITS* .....	711
2.7.2.85	*ALL-OVERLAYS* .....	711
2.7.2.86	*DB-INSTANCES* .....	712
2.7.2.87	*STEALTH-SITE-NUMBER*.....	713
2.7.2.88	*STEALTH-HOST-NUMBER* .....	713
2.7.2.89	*LAST-UNITS-LENGTH* .....	713
2.7.2.90	*LAST-UNITS-SPEED* .....	714
2.7.2.91	*LAST-UNITS-ALTITUDE*.....	714
2.7.2.92	*BATTLEMASTER-PASSWORD*.....	714
2.7.2.93	HOLD-HOVER.....	714
2.7.2.94	HOLD-ORBIT.....	715
2.7.2.95	HOLD-RACETRACK.....	715
2.7.3	CSU sys>macros.lisp .....	715
2.7.3.1	MATH-TO-COMPASS.....	715
2.7.3.2	RADIANS-COMPASS-TO-RADIANS-MATH.....	716
2.7.3.3	RADIANS-MATH-TO-RADIANS-COMPASS.....	716
2.7.3.4	COMPASS-ANGLE.....	717
2.7.3.5	MATH-ANGLE.....	717
2.7.3.6	RADIANS-COMPASS-TO-MILS .....	717
2.7.3.7	RADIANS-MATH-TO-MILS .....	718
2.7.3.8	MILS-TO-RADIANS-COMPASS .....	718
2.7.3.9	MILS-TO-RADIANS-MATH .....	719
2.7.3.10	CVV-MILS-PRINTER .....	719
2.7.3.11	CVV-MILS-READER.....	720
2.7.3.12	(GET 'MILS 'CHOOSE-VARIABLE-VALUES- KEYWORD).....	720



2.7.3.13	SQ.....	720
2.7.3.14	*!.....	721
2.7.3.15	APPROX-COS .....	721
2.7.3.16	APPROX-SIN.....	721
2.7.3.17	ADD-TO-UPDATE-QUEUE .....	722
2.7.3.18	QUEUE-PUSH-LAST .....	723
2.7.3.19	QUEUE-ERASE-EFFECT.....	723
2.7.3.20	SAY.....	723
2.7.3.21	MAYBE-SAY.....	724
2.7.3.22	TALK.....	725
2.7.3.23	SAY-VARIABLES.....	725
2.7.3.24	SAY-VARS.....	725
2.7.3.25	SAY-FORM.....	726
2.7.3.26	SAY-LET.....	726
2.7.3.27	SAY-LET*.....	726
2.7.3.28	SAY-LET-AUX .....	726
2.7.3.29	WHEN-EIGHT-BIT-COLOR.....	727
2.7.3.30	UNLESS-EIGHT-BIT-COLOR.....	727
2.7.3.31	*BUTTERFLY-LOGIN-NAME* .....	727
2.7.3.32	*BUTTERFLY-PASSWORD*.....	728
2.7.3.33	WITH-AUTOMATIC-LOGIN .....	728
2.7.3.34	WITH-OPEN-FILE-ON-BUTTERFLY.....	729
2.7.3.35	HANDLE-LOGIN .....	730
2.7.3.36	ENQUEUE.....	731
2.7.3.37	DEQUEUE.....	731
2.7.3.38	LAST-ITEM-ON.....	732
2.7.3.39	NEXT-ITEM-OFF .....	732
2.7.3.40	MAPQUEUE .....	732
2.7.3.41	QUEUE-LENGTH.....	733
2.7.3.42	QUEUE-FOR-UPDATE-PROCESS .....	733
2.7.3.43	*NAN* .....	733
2.7.3.44	*BREAK-ON-NANS*.....	733
2.7.3.45	NANP .....	734
2.7.4	CSU sys>reader-macros.lisp.....	734
2.7.4.1	GET-DEFSTRUCT-CONSTRUCTOR-MACRO- INFO.....	734

	2.7.4.2	SANDBOX-READER-MACRO .....	735
	2.7.4.3	#.....	735
	2.7.4.4	DEFSTRUCT-ALL-SLOTS .....	735
	2.7.4.5	DEFSTRUCT-ACCESSOR-PREFIX.....	735
	2.7.4.6	DEFSTRUCT-SLOT-VAL-PAIRS .....	736
	2.7.4.7	SANDBOX-PRINTER.....	736
2.7.5		CSU sys>cl-tv-patches .lisp.....	736
	2.7.5.1	OPFOR-TEMPORARY-CHOOSE-VARIABLE- VALUES-WINDOW .....	737
	2.7.5.2	OPFOR-CHOOSE-VARIABLE-VALUES .....	737
	2.7.5.3	WARP-MOUSE-TO-DONE-BOX.....	737
	2.7.5.4	(METHOD OPFOR-TRIANGULATE-CONVEX- POLYGON GRAPHICS-MIXIN).....	738
	2.7.5.5	OPFOR-MENU-CHOOSE.....	738
2.7.6		CSU sys>zl-tv-patches.lisp.....	738
	2.7.6.1	MOUSE-DEFAULT-HANDLER .....	738
	2.7.6.2	WHO-LINE-NO-WINDOW- DOCUMENTATION .....	739
	2.7.6.3	OPFOR-CHOOSE-VARIABLE-VALUES- PROCESS-MESSAGE.....	739
2.7.7		CSU fonts>character-style-defs.lisp.....	739
	2.7.7.1	*B&W-SCREEN* .....	739
2.7.8		CSU fonts>janus-logos.bfd .....	740
2.7.9		CSU fonts>military-icons.bfd.....	740
2.7.10		CSU ui>parameter-menus.lisp .....	740
	2.7.10.1	*ROBO-COP-CONTROL* .....	740
	2.7.10.2	*ROBO-COP-CONTROL* .....	740
	2.7.10.3	ROBO-COP-CONTROL .....	741
2.7.11		CSU sys>interim-model.lisp .....	741
	2.7.11.1	*OPFOR-FORMATIONS-PATH* .....	741
	2.7.11.2	*BLUEFOR-FORMATIONS-PATH* .....	742
	2.7.11.3	*OPFOR-ECHELONS-PATH* .....	742
	2.7.11.4	*BLUEFOR-ECHELONS-PATH* .....	742
	2.7.11.5	*OPFOR-CIS-PATH* .....	742
	2.7.11.6	*BLUEFOR-CIS-PATH* .....	743
	2.7.11.7	*MAPPINGS-PATH* .....	743
	2.7.11.8	*OPFOR-FORMATIONS* .....	743
	2.7.11.9	*BLUEFOR-FORMATIONS* .....	743

2.7.11.10	*OPFOR-ECHELONS*	744
2.7.11.11	*BLUEFOR-ECHELONS*	744
2.7.11.12	*MAPPINGS-ALIST*	744
2.7.11.13	*OPFOR-CIS-DATA*	745
2.7.11.14	*BLUEFOR-CIS-DATA*	745
2.7.11.15	*HOST-FOR-CONFIG-DATA*	745
2.7.11.16	READ-DATA-FILE	746
2.7.11.17	GET-FORMATION-DATA	746
2.7.11.18	*OPFOR-SYNONYMS*	748
2.7.11.19	*BLUEFOR-SYNONYMS*	748
2.7.11.20	GET-RIGHT-FORMATIONS	748
2.7.11.21	GET-RIGHT-ECHELONS	749
2.7.11.22	GET-RIGHT-CISS	749
2.7.11.23	FIND-FORMATIONS	750
2.7.11.24	GET-TYPES-FOR-ECHELON	750
2.7.11.25	GET-ECHELON-AND-TYPES	750
2.7.11.26	GET-VEHICLE-ECHELONS-AND-TYPES	751
2.7.11.27	CAR-EQL	751
2.7.11.28	REV-ASSOC	751
2.7.11.29	MAP-NUMBER-TO-ICON	752
2.7.11.30	MAP-NUMBER-TO-ECHELON	752
2.7.11.31	MAP-ECHELON-TO-NUMBER	752
2.7.11.32	MAP-ECHELON-TYPE-TO-NUMBER	753
2.7.11.33	MAP-ECHELON-TYPE-TO-ICON	753
2.7.11.34	GET-CIS-KEY	753
2.7.11.35	CISS-FOR-ECHELON	754
2.7.11.36	CISS-FOR-CONTROL-MEASURE	754
2.8	UTILITIES CSC	755
2.8.1	CSU sys>utilities.lisp	755
2.8.1.1	FV	755
2.8.1.2	DELETE-DISPLAYED-PRESENTATION	756
2.8.1.3	MENU-CHOOSE	756
2.8.1.4	'(FORMAT-COORDINATES SC WC)	757
2.8.1.5	FORMAT-COORDINATES	757
2.8.1.6	DRAW-STEALTH	758
2.8.1.7	M/SEC-TO-SPEED	758

2.8.1.8	SPEED-TO-M/SEC .....	759
2.8.2	CSU sys>time.lisp.....	759
2.8.2.1	REL-ETIME-TO-SYMBOLICS-TIME.....	759
2.8.2.2	SYMBOLICS-TIME-TO-BFLY-TIME.....	759
2.8.2.3	REL-ETIME-TO-BFLY-TIME .....	760
2.8.2.4	WALL-TIME-TO-REL-ETIME .....	760
2.8.2.5	TIME-COMPARE.....	760
2.8.2.6	MONTHS-ARRAY .....	760
2.8.2.7	MILITARY-TIME-STRING-FROM-BFLY- NUMBER.....	761
2.8.2.8	MILITARY-TIME-STRING-FROM- UNIVERSAL-TIME.....	761
2.8.2.9	DATE-TIME-GROUP .....	761
2.8.3	CSU sys>dw-presentation-types.lisp .....	761
2.8.3.1	'(TYPE-OR-TOKEN TYPE-OR-NULL TYPE- OR-NO-CHANGE).....	762
2.8.3.2	TYPE-OR-TOKEN .....	762
2.8.3.3	TYPE-OR-NULL .....	763
2.8.3.4	TYPE-OR-NO-CHANGE.....	763
2.9	COMPILATION AND INSTALLATION CSC .....	764
2.9.1	CSU sys>site>saf.system .....	764
2.9.2	CSU sys>site>saf.translations .....	764
2.9.3	CSU sys>site>map.system.....	764
2.9.4	CSU sys>site>map.translations.....	764
2.9.5	CSU saf>sysdcl.lisp.....	765
2.9.5.1	NAME .....	765
2.9.5.2	(FIND-PACKAGE 'DIRT).....	766
2.9.5.3	(FIND-PACKAGE 'MAP).....	766
2.9.5.4	(FIND-PACKAGE 'SAF) .....	767
2.9.5.5	*TERRAIN-INITIALIZATION-LIST* .....	767
2.9.5.6	MAKE-AREA .....	767
2.9.5.7	*BACKGROUND-LISP-INTERACTOR- SCREEN-FRACTION* .....	767
2.9.5.8	(OR (MEMBER EIGHT-BIT-COLOR *FEATURES*)).....	768
2.9.5.9	SAF .....	768
2.9.5.10	NETWORK-COMMS .....	768
2.9.5.11	OBJECTS .....	768
2.9.5.12	CONTROL.....	769

2.9.5.13	UI .....	769
2.9.5.14	SANDBOX .....	769
2.9.5.15	MODEL.....	769
2.9.5.16	BMI.....	770
2.9.6	CSU map>defsystem.lisp.....	770
2.9.6.1	(FIND-PACKAGE 'DIRT).....	770
2.9.6.2	(FIND-PACKAGE 'MAP).....	770
2.9.6.3	MAP .....	770
2.9.7	CSU saf>lispm-init.lisp.....	771
APPENDIX A1: WATER AVOIDANCE ALGORITHM.....		A1-1
A1.1	SKIRTING .....	A1-1
A1.2	LOOKING FOR ENDS .....	A1-3
A1.3	RECURSION.....	A1-5
A1.4	RELAXATION.....	A1-6
A1.5	CLEARANCES .....	A1-8
APPENDIX A2: RUDP -- A RELIABLE UDP NETWORK PROTOCOL FOR SIMHOST/SYMBOLICS COMMUNICATIONS .....		A2-1
A2.1	THE NEED FOR RELIABLE COMMUNICATIONS .....	A2-1
A2.2	THE RUDP PROTOCOL IN OVERVIEW .....	A2-1
A2.3	THE PROTOCOL ALGORITHM.....	A2-1
A2.4	RUDP ON THE SYMBOLICS.....	A2-3
A2.5	RUDP ON THE SIMHOST .....	A2-4
APPENDIX A3: CROSS-REFERENCE GENERATOR.....		A3-1
APPENDIX B: INDEX OF DEFINITIONS .....		B-1
INDEX BY SECTION NUMBER .....		Index-1

## **1 INTRODUCTION: SAF WORKSTATION CSCI**

### **1.1 Background**

The Semi-Automated Forces (SAF) system provides a method of incorporating intelligent, realistic, participants that do not require a vehicle simulation. A Semi-Automated vehicle is virtually indistinguishable from a manned vehicle simulation to all observers on the network. The SAF system provides command and control for up to a battalion of Semi-Automated vehicles.

Since the detailed simulation of many realistic vehicles is a computation-intensive task, the SAF system is typically partitioned across several processing platforms. One platform provides the user interface and another the vehicle simulation. This division is reflected in the SAF CSCI structure.

The SAF Workstation CSCI provides the user interface by which the SAF operator controls and monitors SAF. The Workstation CSCI contains code to transmit orders to and receive reports from the SAF Simhost CSCI which performs the simulation of the SAF vehicles and units. The SAF Workstation CSCI is also capable of saving and restoring exercises.

### **1.2 External Interfaces**

To execute a user request and to continue to update the user display the SAF Workstation must communicate to the SAF Parameter Editor and the SAF Simulation Host (Simhost). The Parameter Editor is used to modify the default settings for vehicle behavior, weapons effectiveness, and vehicle interactions. The communication to the Parameter Editor is at the request of the Workstation user. The Simhost provides the detailed vehicle simulation for each of the entities under the command of the Workstation user. The Simhost also provides the interface to the SIMNET network. Since the Simhost maintains the state of the vehicles in the exercise, it continuously updates the Workstation with information to update the display.

The SAF Workstation CSCI interfaces with the SAF Simulation Host (Simhost) CSCI via the SAF Command Protocol. This protocol is described in an appendix to the SAF Parameter Editor CSCI software design document.

The SAF Workstation CSCI interfaces with the SAF Parameter Editor CSCI via the SAF parameter files. These files are described in an appendix to the SAF Parameter Editor CSCI software design document.

Figure 1.2-1 depicts the interfaces between the SAF system components.

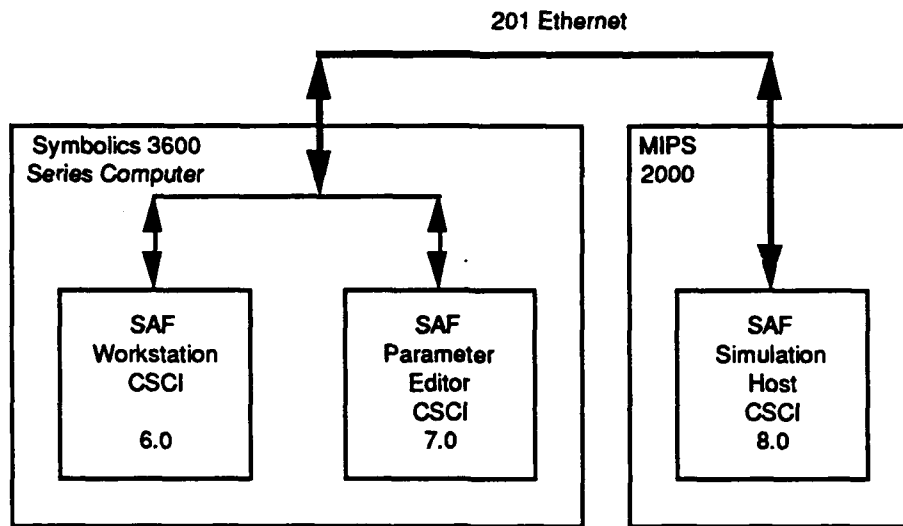


Figure 1.2-1: Software Interface Organization

### 1.3 Internal Structure

The SAF station operator is given battlefield information by means of two monitors. A high resolution color monitor provides a plan view map display of the battlefield showing the battlefield terrain and the state of the units out on the battlefield. This is called the Plan View Display, abbreviated PVD. It also shows the graphical control measures which control the behavior of own units (units controlled by the workstation) and fire activity out on the battlefield. A monochrome monitor provides two different displays corresponding to the two major modes of use of the workstation: battlemaster mode and commander mode.

In battlemaster mode, the monochrome monitor provides commands for selecting, placing, and creating vehicles out on the battlefield while the plan view map display provides a way to position the units. The monochrome monitor also allows some of the global characteristics of the exercise to be set, such as commander's view or omniscient view. Under commander's view, the map display only shows the enemy vehicles which the own vehicles can see. Under omniscient view, the map display shows where everyone is.

In commanders mode, the monochrome monitor displays the task organization of the own units, a message log, and a menu for setting the unit command mode of the workstation. The different command modes allow the SAF operator to either prepare a graphical operations order, issue the operations orders and fragmentary orders to selected units, issue TAC/Es (short immediate fragmentary orders) to units, and request status reports from the units.

The SAF Workstation CSCI is composed of the Computer Software Components (CSCs) depicted in Figure 1.3-1.

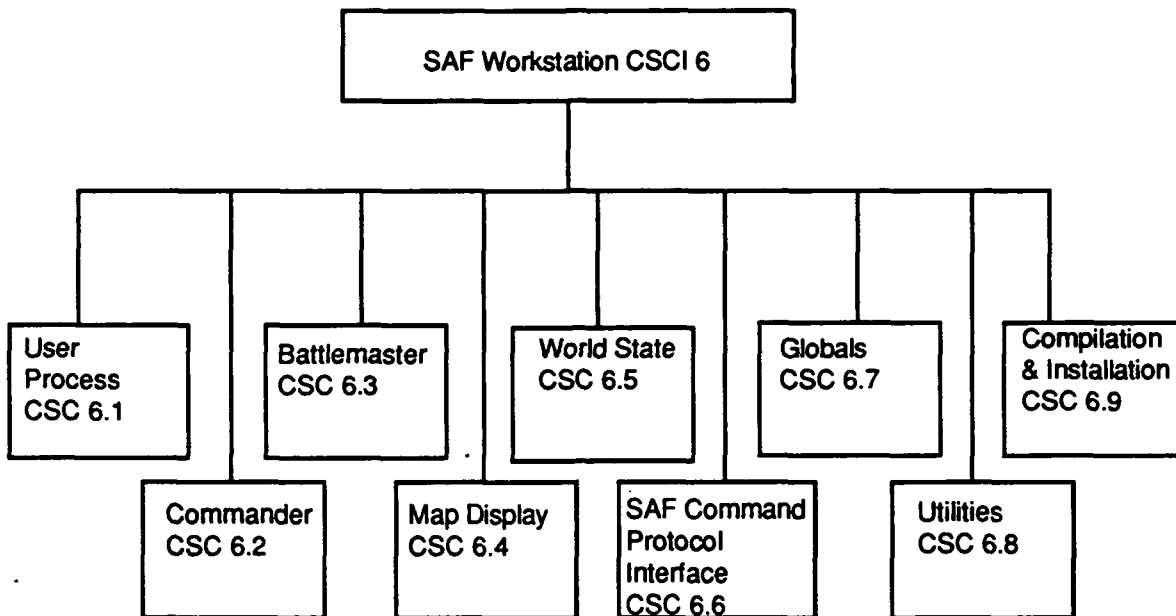


Figure 1.3-1 CSCs of the SAF Workstation CSCI

## 1.4 Configuration and Configuration Management

The SAF Workstation CSCI runs on the SAF Workstation HWCI, which is currently a Symbolics 3650 Imagination, executing under the Genera 7.0 operating system. The SAF Workstation CSCI is written in Symbolics Common Lisp plus the Flavors object-oriented programming system which is included in Genera. A custom enhancement of Flavors, called Objects, is defined and used to automate certain Flavor operations. The Objects system is described in CSU objects>defobject.lisp. A few operating system patches are written in Zeta Lisp; see CSU sys>zl-tv-patches.lisp.

## 1.5 Terminology and Documentation

### 1.5.1 Glossary

ADA	Air Defense Artillery
AI	Artificial Intelligence
alu	Arithmetic Logic Unit (on Symbolics machine)
BMI	SAF Battlemaster Interface
CIS	SAF Combat Instruction Set
CP	Lisp Command Processor
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
CVV	Symbolics Choose Variable Values utility



EGC	Symbolics Ephemeral Garbage Collector
FWA	Fixed Wing Aircraft
HWCI	Hardware Configuration. Item
IFV	Infantry Fighting Vehicle
IP/TCP	Internet Protocol/Transmission Control Protocol
IP/UDP	Internet Protocol/Universal Datagram Protocol
IVIS	SAF Inter-Vehicular Information System
Lisp	List Processing
MIPS	A particular Simhost hardware platform
OPFOR	SAF Opposing Forces
OPORD	SAF Operational Order
PVD	SAF Plan View Display
RUDP	Reliable Universal Datagram Protocol
RWA	Rotary Wing Aircraft
SAF	Semi-Automated Forces
SIMHOST	SIMNET Simulation Host
TAC/E	Tactical/Emergency
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	Universal Datagram Protocol
UTM	Universal Transverse Mercator Coordinate System

### 1.5.2 Related Documentation

Symbolics Genera 7.0 manuals, Symbolics, July 1986

Common Lisp, the Language, 2nd Edition, Guy L. Steele Jr., Digital Press, 1990

Datums, Ellipsoids, Grids, and Grid Reference Systems, Defense Mapping Agency document DMA TM 8358.1

Operational Terms and Symbols, Field Manual FM 101-5-1, US Department of the Army, 21 October 1985

Terrain Reasoning in the Simnet Semi-Automated Forces System, Thomas Stanzone, BBN Systems and Technologies Corporation, Report No. 7140, October 1989

Artificial Intelligence, 2nd Edition, Patrick Henry Winston, Addison Wesley 1984

Object-Oriented Programming in Common Lisp, Sonya E. Keene, Addison Wesley 1989

## 1.6 Miscellaneous

### 1.6.1 Automatically Generated Definition Cross-Reference

SAF Documentation includes an automatically-generated cross reference of definitions. Each cross-reference entry is included as a numbered section in the CSU that contains it. Each entry includes the name of the defined object, the path name of the file where it is defined, its type, arguments, calls, callers, and description. A sample entry is shown below:

#### 2.1.1.3 MAKE-OPFOR-SUB-PROCESS-FUNCTION

Type:	>saf>ui>processes.lisp Function
Arguments:	(NAME BODY-FUNCTION)
Outputs:	
Calls:	MAKE-OPFOR-SUB-PROCESS-FUNCTION-1 >saf>ui>processes.lisp
Called by:	(METHOD MAKE-INSTANCE OPFOR-SUB-PROCESS AFTER) >saf>ui>processes.lisp
Description:	None

The first entry is the name of the defined object, on the same line as the section heading number. On the next line is the path name of the file containing the definition.

The Type entry indicates whether the object is a function, method, parameter, variable, flavor, presentation type, etc.

The Arguments entry lists the arguments of the object if it is a function or method.

The Outputs entry is currently left blank, since there is no generally accepted definition of the output of a Lisp object.

The Calls entry is a list of the definitions called by the object. A parameter or variable is considered to be called if it is referred to or used by the lisp forms of the object being described. Each call is followed by the pathname of the file in which the called object is defined.

The Called By entry is a similar list of the objects which call the object being described; the pathname of the calling object's file is also included.

The Description entry contains the contents of the description string in the lisp object, if it has one. Since most objects were not given description strings when the code was written, most of these entries are "None".

It is important to realize that this cross-reference does not expand each definition to see if it contains other definitions. This means that only top-level Lisp forms are listed. As a result, important forms may be absent from the cross reference. For example, a deffavor form may appear inside an eval-when form, for compilation purposes, as follows:

```
(eval-when (compile load eval)
  (deffavor object-x ... ))
```

In this case, the flavor object-x will not appear in the cross-reference, either as a main entry, or a caller. However, if it is referred to by a main entry, it may appear as a callee. The cross reference will list this entry as a definition of the Lisp form (COMPILE LOAD EVAL), of type EVAL-WHEN, with a nil argument list.

The cross-referencing program was not designed to be complete; it was intended as a basic tool that would provide some useful information. The source code for the program is included in Appendix A3, Cross-Reference Generator.

The Index of Definitions was generated by the same Lisp program that constructed the cross reference. As a result, functions that do not appear in the cross reference will also be missing from the index. The index of definitions is found in Appendix B.

### 1.6.2 Auxiliary Functions

A number of lisp forms used in SAF code define auxiliary functions when called. This feature, an example of the self-modifying capability of lisp, can be confusing because an auxiliary function does not have a form which obviously defines it. For example, the Common Lisp macro *defstruct* automatically creates a constructor function. If a structure has the name *thing*, the constructor function will be given the name *make-thing* by default. However, there will be no lexically apparent definition of *make-thing*, that could be located, for example, by a text search.

The following is a list of all those forms used in the SAF code which create auxiliary functions:

- def-packet-handler
- defflavor (Symbolics)
- define-array-accessors
- define-model-menu-command
- define-program-framework (Symbolics)
- defobject
- defsend
- defstorage (lmfs:defstorage) (Symbolics)
- defstruct (Common Lisp)

Those forms that are part of the Symbolics software are documented in the Symbolics manuals, with the exception of *lmfs:defstorage*, which is discussed in the CSU *network>defstorage.lisp*. The others are part of the SAF software. In most cases, the source code for the form makes it clear what the names of the auxiliary functions will look like.

For example, the macro *def-packet-handler*, defined in *rudp>handle-incoming.lisp*, uses the backquote-comma macro syntax to create the function names *PRINT-XXX-PKT* and *PROCESS-XXX-PKT*, where *XXX* is the print-name of the packet-type argument.

### 1.6.3 Commented-Out Code

Most files in the SAF system contain blocks of lisp code that have been commented out, either with semicolons, or by placing the reader syntax command *#+ignore* in front of the form to be suppressed. In most cases, such code is truly obsolete and should be deleted. This is especially true if the removal is dated several months back.

In some cases, however, the code has a useful function that is not currently needed; this is usually indicated by comments, or explained in this documentation.

In other cases, commented code contains important information on how actual "live" code was constructed. For example, a code fragment obtained from an expert may be included, as a reference, near the form it was used to design. Such situations are usually clear from context; the text should be retained.

## 2 CSCI FUNCTIONS AND INTERFACES

### 2.1 User Process CSC

The User Process CSC contains the code which is run by the user process. This process receives and executes the user's mouse and keyboard commands. An additional process, the mouse warp process, is used to transport the mouse cursor between the color and the monochrome monitor whenever it gets to the left or right edge of the screen. The majority of the commands which the user can issue are invoked by clicking on a mouse sensitive item on either the color or the monochrome monitor. These commands are discussed in the context of the windows and panes in which the mouse sensitive items are displayed. The code in this section also defines the frame for the color and the monochrome display windows and the top level loop which the user process runs to accept user input. Any terrain redisplay commands which the user process receives are queued on the update process which executes them. Any commands which generate messages to the simhost are queued on the RUDP process which transmits them to the Simhost. The SAF workstation has two basic operating modes, the battlemaster mode for initializing the SAF system and the commander mode which controls the forces which the battlemaster has set up. The window displayed on the monochrome monitor is different for the two modes and contains different commands. The User Process CSC contains the following CSUs.

```
ui>processes.lisp csu
ui>menus.lisp csu
ui>mouse-interface.lisp csu
ui>frame.lisp csu
ui>commands.lisp csu
```

#### 2.1.1 CSU ui>processes.lisp

This unit contains the top-level functions for the background processes used in the SAF system. SAF is composed of several processes. The user process is created automatically with the SAF program-framework. This unit defines two background processes: First, the RUDP process, *process-rudp-packets*, which handles communication with the Simhost machine. This task has to have its own subprocess because information requiring immediate attention can arrive over the network at any time, even while the user process is busy. Second, the Update process, *update-top-level*, which handles the redisplay of objects on the PVD. This task was given its own subprocess because screen-painting can be time-consuming, and would lock up the user interface until done, if it was part of the user process. By making update a background process, the user interface stays responsive, even during long screen updates.

This CSU also defines several utilities for handling processes, and a flavor used to represent SAF sub-processes. Processes are described in detail in the Symbolics manuals.

Note: There is another process used by SAF, the mouse-process, but it is a temporary process, created when needed to perform a single mouse-warp action. See CSU ui>mouse-interface.lisp, section 2.1.3.

**2.1.1.1 DYING-PROCESS**

## Definition 1

Type: >saf>ui>processes.lisp  
 Function  
 Arguments: (PROCESS)  
 Outputs:  
 Calls: NAME  
 >saf>sysdcl.lisp  
 \*OPFOR-IO\*  
 >saf>sys>vars.lisp  
 SAY  
 >saf>sys>macros.lisp  
 Called by: MAKE-OPFOR-SUB-PROCESS-FUNCTION-1  
 >saf>ui>processes.lisp  
 Description: None

**2.1.1.2 MAKE-OPFOR-SUB-PROCESS-FUNCTION-1**

## Definition 2

Type: >saf>ui>processes.lisp  
 Macro  
 Arguments: (FUNCTION-NAME BODY)  
 Outputs:  
 Calls: \*OPFOR-FRAME\*  
 >saf>sys>vars.lisp  
 DYING-PROCESS  
 >saf>ui>processes.lisp  
 MAKE-OPFOR-SUB-PROCESS-FUNCTION-1  
 >saf>ui>processes.lisp  
 Called by: MAKE-OPFOR-SUB-PROCESS-FUNCTION  
 >saf>ui>processes.lisp  
 MAKE-OPFOR-SUB-PROCESS-FUNCTION-1  
 >saf>ui>processes.lisp  
 Description: None

**2.1.1.3 MAKE-OPFOR-SUB-PROCESS-FUNCTION**

## Definition 3

Type: >saf>ui>processes.lisp  
 Function  
 Arguments: (NAME BODY-FUNCTION)  
 Outputs:  
 Calls: MAKE-OPFOR-SUB-PROCESS-FUNCTION-1  
 >saf>ui>processes.lisp  
 Called by: (METHOD MAKE-INSTANCE OPFOR-SUB-PROCESS AFTER)  
 >saf>ui>processes.lisp  
 Description: None

**2.1.1.4 COM-CHECK-OPFOR-PROCESSES**

## Definition 4

Type: >saf>ui>processes.lisp  
 CP Command  
 Arguments: ()  
 Outputs:  
 Calls: OPFOR-SUB-PROCESS-REPORTS  
 >saf>ui>processes.lisp  
 Called by: None  
 Description: None

**2.1.1.5 COM-SAF-CHECK-OPFOR-PROCESSES**

## Definition 5

Type: >saf>ui>processes.lisp  
 CP Command  
 Arguments: ()  
 Outputs:  
 Calls: OPFOR-SUB-PROCESS-REPORTS  
 >saf>ui>processes.lisp  
 Called by: None  
 Description: None

**2.1.1.6 OPFOR-SUB-PROCESS-REPORTS**

## Definition 6

Type: >saf>ui>processes.lisp  
 Function  
 Arguments: ()  
 Outputs:  
 Calls: \*OPFOR-FRAME\*  
 >saf>sys>vars.lisp  
 Called by: COM-SAF-CHECK-OPFOR-PROCESSES  
 >saf>ui>processes.lisp  
 COM-CHECK-OPFOR-PROCESSES  
 >saf>ui>processes.lisp  
 Description: None

**2.1.1.7 SECONDS-AGO**

## Definition 7

Type: >saf>ui>processes.lisp  
 Subst  
 Arguments: (UNIVERSAL-TIME)  
 Outputs:  
 Calls: None  
 Called by: (METHOD REPORT OPFOR-SUB-PROCESS)  
 >saf>ui>processes.lisp  
 Description: None

**2.1.1.8 OPFOR-SUB-PROCESS**

Definition 8

Type: >saf>ui>processes.lisp  
 Flavor  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

**2.1.1.9 '(OPFOR-SUB-PROCESS PROCESS)**

Definition 9

Type: >saf>ui>processes.lisp  
 EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

**2.1.1.10 \*ALL-OPFOR-SUB-PROCESSES\***

Definition 10

Type: >saf>ui>processes.lisp  
 Variable  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (METHOD REMEMBER OPFOR-SUB-PROCESS)  
 >saf>ui>processes.lisp  
 (METHOD MURDER OPFOR-SUB-PROCESS)  
 >saf>ui>processes.lisp  
 (METHOD MAKE-INSTANCE OPFOR-SUB-PROCESS AFTER)  
 >saf>ui>processes.lisp  
 Description: bookkeeping of sub-processes

**2.1.1.11 (METHOD MAKE-INSTANCE OPFOR-SUB-PROCESS AFTER)**

Definition 11

Type: >saf>ui>processes.lisp  
 Method  
 Arguments: (&REST IGNORE)  
 Outputs:  
 Calls: MAKE-OPFOR-SUB-PROCESS-FUNCTION  
 >saf>ui>processes.lisp  
 \*ALL-OPFOR-SUB-PROCESSES\*  
 >saf>ui>processes.lisp  
 Called by: None  
 Description: None



**2.1.1.12 (METHOD DISABLE OPFOR-SUB-PROCESS)**

Definition 12

Type: >saf>ui>processes.lisp  
Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.1.1.13 (METHOD ENABLE OPFOR-SUB-PROCESS)**

Definition 13

Type: >saf>ui>processes.lisp  
Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.1.1.14 (METHOD MURDER OPFOR-SUB-PROCESS)**

Definition 14

Type: >saf>ui>processes.lisp  
Method  
Arguments: ()  
Outputs:  
Calls: \*ALL-OPFOR-SUB-PROCESSES\*  
>saf>ui>processes.lisp  
Called by: None  
Description: None

**2.1.1.15 (METHOD REMEMBER OPFOR-SUB-PROCESS)**

Definition 15

Type: >saf>ui>processes.lisp  
Method  
Arguments: ()  
Outputs:  
Calls: \*ALL-OPFOR-SUB-PROCESSES\*  
>saf>ui>processes.lisp  
Called by: None  
Description: None

### 2.1.1.16 (METHOD REPORT OPFOR-SUB-PROCESS)

Definition 16

Type: >saf>ui>processes.lisp  
Method  
Arguments: ()  
Outputs:  
Calls: \*OPFOR-IO\*  
>saf>sys>vars.lisp  
SAY  
>saf>sys>macros.lisp  
SECONDS-AGO  
>saf>ui>processes.lisp  
Called by: None  
Description: None

### 2.1.1.17 \*RUDP-PROCESS-LAST-CYCLE\*

Definition 17

Type: >saf>ui>processes.lisp  
Parameter  
Arguments: ()  
Outputs:  
Calls: None  
Called by: PROCESS-RUDP-PACKETS  
>saf>ui>processes.lisp  
(METHOD MAKE-INSTANCE SAF AFTER)  
>saf>ui>frame.lisp  
MAKE-RUDP-PROCESS  
>saf>ui>processes.lisp  
Description: None

### 2.1.1.18 NETWORK-PROCESS-WAKE-UP

Definition 18

Type: >saf>ui>processes.lisp  
Function  
Arguments: ()  
Outputs:  
Calls: RETRANSMIT\_PERIOD  
>saf>network>vars.lisp  
\*SIM-CONN\*  
>saf>rudp>vars.lisp  
\*RETRANSMIT-TIMER\*  
>saf>rudp>vars.lisp  
\*PACKET-REQUEST-QUEUE\*  
>saf>rudp>vars.lisp  
\*PACKET-IMMEDIATE-QUEUE\*  
>saf>rudp>vars.lisp  
STANDALONEP  
>saf>network>connection.lisp

Called by: PROCESS-RUDP-PACKETS  
 >saf>ui>processes.lisp  
 Description: decides when to make the update process runnable

### 2.1.1.19 PROCESS-RUDP-PACKETS

Definition 19

Type: >saf>ui>processes.lisp  
 Function  
 Arguments: ()  
 Outputs:  
 Calls: \*RADIO-OUTPUT\*  
 >saf>sys>vars.lisp  
 STANDALONEP  
 >saf>network>connection.lisp  
 RUDP-TRANSMIT-AND-RECEIVE  
 >saf>rudp>utils.lisp  
 \*RUDP-PROCESS-LAST-CYCLE\*  
 >saf>ui>processes.lisp  
 NETWORK-PROCESS-WAKE-UP  
 >saf>ui>processes.lisp  
 Called by: (METHOD MAKE-INSTANCE SAF AFTER)  
 >saf>ui>frame.lisp  
 MAKE-RUDP-PROCESS  
 >saf>ui>processes.lisp  
 Description: None

### 2.1.1.20 MAKE-RUDP-PROCESS

Definition 20

Type: >saf>ui>processes.lisp  
 Subst  
 Arguments: ()  
 Outputs:  
 Calls: \*RUDP-PROCESS-LAST-CYCLE\*  
 >saf>ui>processes.lisp  
 PROCESS-RUDP-PACKETS  
 >saf>ui>processes.lisp  
 Called by: (METHOD MAKE-INSTANCE SAF AFTER)  
 >saf>ui>frame.lisp  
 Description: None

### 2.1.1.21 MAKE-UPDATE-PROCESS

Definition 21

Type: >saf>ui>processes.lisp  
 Subst  
 Arguments: ()  
 Outputs:

**Calls:**                    \*UPDATE-PROCESS-LAST-CYCLE\*  
                               >saf>sys>update-process.lisp  
                               UPDATE-TOP-LEVEL  
**Called by:**                >saf>sys>update-process.lisp  
                               (METHOD MAKE-INSTANCE SAF AFTER)  
                               >saf>ui>frame.lisp  
**Description:**             None

### 2.1.2 CSU ui>menus.lisp

This unit contains functions that implement some static menus. These include the "Types of Terrain to Draw" menu, and the Zoom menu. The methods MULTIPLE-CHOICE-ALL-SHOW and MULTIPLE-CHOICE-ALL-HIDE are modifications of the Symbolics method MULTIPLE-CHOICE-DONE of the flavor MULTIPLE-CHOICE-MIXIN. Because these methods directly modify system variables in the tv package, such as choice-box-state, they may not be portable to new Genera versions. The items in the Zoom menu include documentation strings, signaled by the keyword *:documentation*.

#### 2.1.2.1 (METHOD MULTIPLE-CHOICE-ALL-SHOW MULTIPLE-CHOICE-MIXIN)

Definition 1

                              >saf>ui>menus.lisp  
**Type:** Method  
**Arguments:** (CHOSEN THING Y)  
**Outputs:**  
**Calls:** None  
**Called by:** None  
**Description:** None

#### 2.1.2.2 (METHOD MULTIPLE-CHOICE-ALL-HIDE MULTIPLE-CHOICE-MIXIN)

Definition 2

                              >saf>ui>menus.lisp  
**Type:** Method  
**Arguments:** (CHOSEN THING Y)  
**Outputs:**  
**Calls:** None  
**Called by:** None  
**Description:** None

#### 2.1.2.3 \*TERRAIN-MENU\*

Definition 3

                              >saf>ui>menus.lisp  
**Type:** Variable  
**Arguments:** ()  
**Outputs:**

Calls: None  
 Called by: HANDLE-TERRAIN-MENU  
           >saf>ui>menus.lisp  
           MAYBE-MAKE-TERRAIN-MENU  
           >saf>ui>menus.lisp  
 Description: None

#### 2.1.2.4 MAYBE-MAKE-TERRAIN-MENU

Definition 4

          >saf>ui>menus.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: \*NEW-INTERFACE-FLG\*  
           >saf>ui>mouse-interface.lisp  
           \*TERRAIN-MENU\*  
           >saf>ui>menus.lisp  
           \*NEW-INTERFACE-FLG\*  
           >saf>ui>mouse-interface.lisp  
 Called by: HANDLE-TERRAIN-MENU  
           >saf>ui>menus.lisp  
 Description: None

#### 2.1.2.5 HANDLE-TERRAIN-MENU

Definition 5

          >saf>ui>menus.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: \*TERRAIN-OPTIONS\*  
           >saf>sys>vars.lisp  
           \*INTERFACE-TO-UPDATE-PROCESS-QUEUE\*  
           >saf>sys>vars.lisp  
           ADD-TO-UPDATE-QUEUE  
           >saf>sys>macros.lisp  
           \*TERRAIN-MENU\*  
           >saf>ui>menus.lisp  
           MAYBE-MAKE-TERRAIN-MENU  
           >saf>ui>menus.lisp  
 Called by: (METHOD COM-TERRAIN-OPTIONS-INTERNAL PVD)  
           No Source File Record  
 Description: None

#### 2.1.3 CSU ui>mouse-interface.lisp

This unit contains routines to allow the mouse to be driven between the monochrome display and the color display by the user. This "mouse-warp" capability has actually been written into a modified version of the main Symbolics mouse process. (See the file

sys>zl-tv-patches.lisp, section 2.7.6.) This was done to insure that the mouse warp continues to work even when the SAF user is not running SAF. For example, if a SAF user hit Select-L to run some Lisp forms, the mouse warp would still work in Lisp Listener mode. The main loop of the modified mouse process calls the function *consider-flipping*, defined in this unit. If a decision is made to warp the mouse, a separate process is temporarily created which does this, and then terminates. This separate process is necessary because it is illegal to warp the mouse from the mouse process itself; doing so can cause the mouse pointer to disappear.

The function *consider-flipping* calls the function *execute-in-new-interface*, passing it a "jump-to" form that will warp the mouse when evaluated. Then, *execute-in-new-interface* uses the Symbolics function *process-run-function* to start a new process. The new process just calls *new-interface-process-function*, a simple routine that just evaluates any forms on the queue *\*nip-forms\**. Since *execute-in-new-interface* pushes the "jump-to" form onto *\*nip-forms\**, the "jump-to" form will eventually get evaluated by *new-interface-process-function*, warping the mouse.

### 2.1.3.1 '(\*NEW-INTERFACE-FLG\* CONSIDER-FLIPPING)

Definition 1

```
>saf>ui>mouse-interface.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

### 2.1.3.2 \*NEW-INTERFACE-FLG\*

Definition 2

```
>saf>ui>mouse-interface.lisp
Type: Parameter
Arguments: ()
Outputs:
Calls: None
Called by: COLOR-SCREEN-MENU
           >saf>ui>mouse-interface.lisp
           MAYBE-MAKE-TERRAIN-MENU
           >saf>ui>menus.lisp
           MOUSE-DEFAULT-HANDLER
           >saf>sys>zl-tv-patches.lisp
Description: None
```

### 2.1.3.3 MOUSE-FLIP-SCREEN

Definition 3

```
>saf>ui>mouse-interface.lisp
Type: Function
Arguments: (CHAR PANE)
Outputs:
```

Calls: JUMP-TO-B&W-SCREEN  
       >saf>ui>mouse-interface.lisp  
       JUMP-TO-COLOR-SCREEN  
       >saf>ui>mouse-interface.lisp  
 Called by: (METHOD TOP-LEVEL CONFIGURATION-MENU)  
           >saf>interface>formations.lisp  
           (METHOD TOP-LEVEL OBJECT-MENU)  
           >saf>interface>object-menu.lisp  
           (METHOD TOP-LEVEL MODEL-MENU)  
           >saf>interface>model-menu.lisp  
           (METHOD TOP-LEVEL SAF)  
           >saf>ui>frame.lisp  
 Description: None

#### 2.1.3.4 FIND-MOUSE

Definition 4

          >saf>ui>mouse-interface.lisp  
 Type: Function  
 Arguments: (CHAR PANE)  
 Outputs:  
 Calls: None  
 Called by: (METHOD TOP-LEVEL CONFIGURATION-MENU)  
           >saf>interface>formations.lisp  
           (METHOD TOP-LEVEL OBJECT-MENU)  
           >saf>interface>object-menu.lisp  
           (METHOD TOP-LEVEL MODEL-MENU)  
           >saf>interface>model-menu.lisp  
           (METHOD TOP-LEVEL SAF)  
           >saf>ui>frame.lisp  
 Description: None

#### 2.1.3.5 CONSIDER-FLIPPING

Definition 5

          >saf>ui>mouse-interface.lisp  
 Type: Function  
 Arguments: (SCREEN MOUSE-X-COORD)  
 Outputs:  
 Calls: JUMP-TO-B&W-SCREEN  
       >saf>ui>mouse-interface.lisp  
       JUMP-TO-COLOR-SCREEN  
       >saf>ui>mouse-interface.lisp  
       EXECUTE-IN-NEW-INTERFACE  
       >saf>ui>mouse-interface.lisp  
 Called by: MOUSE-DEFAULT-HANDLER  
           >saf>sys>zl-tv-patches.lisp  
 Description: None

**2.1.3.6 JUMP-TO-B&W-SCREEN**

Definition 6

&gt;saf&gt;ui&gt;mouse-interface.lisp

Type: Function

Arguments: (X-WC)

Outputs:

Calls: None

Called by: MOUSE-FLIP-SCREEN

&gt;saf&gt;ui&gt;mouse-interface.lisp

CONSIDER-FLIPPING

&gt;saf&gt;ui&gt;mouse-interface.lisp

Description: None

**2.1.3.7 JUMP-TO-COLOR-SCREEN**

Definition 7

&gt;saf&gt;ui&gt;mouse-interface.lisp

Type: Function

Arguments: (X-WC)

Outputs:

Calls: None

Called by: MOUSE-FLIP-SCREEN

&gt;saf&gt;ui&gt;mouse-interface.lisp

CONSIDER-FLIPPING

&gt;saf&gt;ui&gt;mouse-interface.lisp

Description: None

**2.1.3.8 \*COLOR-SCREEN-MENU\***

Definition 8

&gt;saf&gt;ui&gt;mouse-interface.lisp

Type: Parameter

Arguments: ()

Outputs:

Calls: None

Called by: COLOR-SCREEN-MENU

&gt;saf&gt;ui&gt;mouse-interface.lisp

Description: None

**2.1.3.9 COLOR-SCREEN-MENU**

Definition 9

&gt;saf&gt;ui&gt;mouse-interface.lisp

Type: Function

Arguments: ()

Outputs:



Calls: \*NEW-INTERFACE-FLG\*  
       >saf>ui>mouse-interface.lisp  
       \*NEW-INTERFACE-FLG\*  
       >saf>ui>mouse-interface.lisp  
       \*COLOR-SCREEN-MENU\*  
       >saf>ui>mouse-interface.lisp  
       CLEAR-UNITS  
       >saf>ui>mouse-interface.lisp  
       CLEAR-UNITS-AND-OVERLAYS  
       >saf>ui>mouse-interface.lisp  
       WORKSTATION-BATTLE-VIEW  
       >saf>bmi>bmi-frame.lisp  
 Called by: (PRESENTATION-MOUSE-HANDLER PVD-COMMAND-MENU)  
           No Source File Record  
 Description: Pops up basic plan view display menu

### 2.1.3.10 CLEAR-UNITS

Definition 10

      >saf>ui>mouse-interface.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: \*DB-INSTANCES\*  
       >saf>sys>vars.lisp  
 Called by: CLEAR-UNITS-AND-OVERLAYS  
       >saf>ui>mouse-interface.lisp  
       COLOR-SCREEN-MENU  
       >saf>ui>mouse-interface.lisp  
 Description: None

### 2.1.3.11 CLEAR-UNITS-AND-OVERLAYS

Definition 11

      >saf>ui>mouse-interface.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: CLEAR-UNITS  
       >saf>ui>mouse-interface.lisp  
       CLEAR-OVERLAYS  
       >saf>ui>mouse-interface.lisp  
 Called by: COLOR-SCREEN-MENU  
       >saf>ui>mouse-interface.lisp  
 Description: None

**2.1.3.12 CLEAR-OVERLAYS**

Definition 12

```

>saf>ui>mouse-interface.lisp
Type: Function
Arguments: ()
Outputs:
Calls: *ALL-OVERLAYS*
       >saf>sys>vars.lisp
       SIMNET-AGENT
       >saf>objects>simnet-agent.lisp
       SIMNET-AGENT
       >saf>objects>simnet-agent.lisp
       SIMNET-AGENT
       >saf>objects>simnet-agent.lisp
       UNIT
       >saf>cm>control-measure.lisp
       OVERLAY
       >saf>cm>overlay.lisp
       OVERLAY
       >saf>cm>overlay.lisp
Called by: CLEAR-UNITS-AND-OVERLAYS
           >saf>ui>mouse-interface.lisp
Description: None

```

**2.1.3.13 \*NEW-INTERFACE-PROCESS\***

Definition 13

```

>saf>ui>mouse-interface.lisp
Type: Parameter
Arguments: ()
Outputs:
Calls: None
Called by: EXECUTE-IN-NEW-INTERFACE
           >saf>ui>mouse-interface.lisp
Description: None

```

**2.1.3.14 \*NIP-FORMS\***

Definition 14

```

>saf>ui>mouse-interface.lisp
Type: Parameter
Arguments: ()
Outputs:
Calls: None
Called by: NEW-INTERFACE-PROCESS-FUNCTION
           >saf>ui>mouse-interface.lisp
           PUSH-NIP-FORM-IF-NECESSARY
           >saf>ui>mouse-interface.lisp
Description: None

```

**2.1.3.15 PUSH-NIP-FORM-IF-NECESSARY**

Definition 15

>saf>ui>mouse-interface.lisp  
Type: Function  
Arguments: (FORM)  
Outputs:  
Calls: \*NIP-FORMS\*  
>saf>ui>mouse-interface.lisp  
Called by: EXECUTE-IN-NEW-INTERFACE  
>saf>ui>mouse-interface.lisp  
Description: None

**2.1.3.16 EXECUTE-IN-NEW-INTERFACE**

Definition 16

>saf>ui>mouse-interface.lisp  
Type: Function  
Arguments: (FORM)  
Outputs:  
Calls: NAME  
>saf>sysdcl.lisp  
\*NEW-INTERFACE-PROCESS\*  
>saf>ui>mouse-interface.lisp  
PUSH-NIP-FORM-IF-NECESSARY  
>saf>ui>mouse-interface.lisp  
NEW-INTERFACE-PROCESS-FUNCTION  
>saf>ui>mouse-interface.lisp  
Called by: CONSIDER-FLIPPING  
>saf>ui>mouse-interface.lisp  
Description: None

**2.1.3.17 NEW-INTERFACE-PROCESS-FUNCTION**

Definition 17

>saf>ui>mouse-interface.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*NIP-FORMS\*  
>saf>ui>mouse-interface.lisp  
Called by: EXECUTE-IN-NEW-INTERFACE  
>saf>ui>mouse-interface.lisp  
Description: None

**2.1.4 CSU ui>frame.lisp**

This unit contains the definition of the color and monochrome display windows and the top-level loop to accept user input, including the actual entry point to the SAF code.

These definitions are a straightforward application of the utilities found in the Symbolics dynamic windows package `dw:`, such as `dw:Define-Program-Framework`. This function provides a general, high-level mechanism for user interfaces on the Symbolics machine. The method *top-level* of the program-framework SAF, defined towards the end of the file, is the main entry point to SAF. This is the code that gets executed when the user types `Select-O`.

Note: This unit makes use of "whoppers", powerful method-overloading constructs that are part of the Symbolics flavors system.

The commented-out forms beginning with "`(defun set-color-who-line`" have the effect of allowing the lisp world to be saved after the color screen is created. They work fine, but were commented out because they were not needed.

The function *Do-Nothing-Command-Loop* is used because it is faster than a null method call. The *pvd* program framework doesn't need a top-level method because it is controlled by the *saf* program framework's top-level method.

The form (`compile-flavor-methods pvd`) precompiles the *pvd* program framework, for speed.

#### 2.1.4.1 STANDARD-MARGINS

Definition 1

```
>saf>ui>frame.lisp
Type: Function
Arguments: (LABEL)
Outputs:
Calls: None
Called by: CONFIGURATION-MENU PROGRAM-FRAME-OPTIONS
>saf>patch>saf-6>saf-6-7.lisp
OBJECT-MENU PROGRAM-FRAME-OPTIONS
>saf>patch>saf-6>saf-6-6.lisp
MODEL-MENU PROGRAM-FRAME-OPTIONS
>saf>patch>saf-6>saf-6-5.lisp
SAF PROGRAM-FRAME-OPTIONS
>saf>ui>frame.lisp
Description: None
```

#### 2.1.4.2 PVD

Definition 2

```
>saf>ui>frame.lisp
Type: DEFINE-PROGRAM-FRAMEWORK
Arguments: ()
Outputs:
Calls: None
Called by: SAF
>saf>ui>frame.lisp
LEFT-ON- TERRAIN OPTIONS -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-TERRAIN-OPTIONS-INTERNAL PVD)
```

```

No Source File Record
LEFT-ON- REFRESH -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-REFRESH-INTERNAL PVD)
No Source File Record
LEFT-ON- MAP SCALE -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-RESCALE-INTERNAL PVD)
No Source File Record
LEFT-ON- ZOOM OUT -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-ZOOM-OUT-INTERNAL PVD)
No Source File Record
LEFT-ON- PAN -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-PAN-INTERNAL PVD)
No Source File Record
LEFT-ON- ZOOM IN -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-ZOOM-IN-INTERNAL PVD)
No Source File Record
DEFINE-PVD-MENU-COMMAND
>saf>ui>commands.lisp
MAKE-PVD-FRAME
>saf>ui>frame.lisp
DEFINE-PVD-COMMAND
>saf>ui>frame.lisp

```

Description: None

### 2.1.4.3 PVD

Definition 3

```

>saf>ui>frame.lisp
Type: COMPILE-FLAVOR-METHODS
Arguments: ()
Outputs:
Calls: None
Called by: SAF
>saf>ui>frame.lisp
LEFT-ON- TERRAIN OPTIONS -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-TERRAIN-OPTIONS-INTERNAL PVD)
No Source File Record
LEFT-ON- REFRESH -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-REFRESH-INTERNAL PVD)
No Source File Record
LEFT-ON- MAP SCALE -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-RESCALE-INTERNAL PVD)
No Source File Record
LEFT-ON- ZOOM OUT -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp

```

```
(METHOD COM-ZOOM-OUT-INTERNAL PVD)
No Source File Record
LEFT-ON- PAN -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-PAN-INTERNAL PVD)
No Source File Record
LEFT-ON- ZOOM IN -AT-PVD-PVD-MENU-COMMAND
>saf>ui>commands.lisp
(METHOD COM-ZOOM-IN-INTERNAL PVD)
No Source File Record
DEFINE-PVD-MENU-COMMAND
>saf>ui>commands.lisp
MAKE-PVD-FRAME
>saf>ui>frame.lisp
DEFINE-PVD-COMMAND
>saf>ui>frame.lisp
```

Description: None

#### 2.1.4.4 DO-NOTHING-COMMAND-LOOP

Definition 4

```
>saf>ui>frame.lisp
Type: Function
Arguments: (PROGRAM)
Outputs:
Calls: None
Called by: None
Description: Do Nothing Top Level Command Loop
```

#### 2.1.4.5 MAKE-PVD-FRAME

Definition 5

```
>saf>ui>frame.lisp
Type: Function
Arguments: (IO-BUFFER)
Outputs:
Calls: PVD
>saf>ui>frame.lisp
PVD
>saf>ui>frame.lisp
Called by: (METHOD TOP-LEVEL SAF)
>saf>ui>frame.lisp
Description: None
```

#### 2.1.4.6 SET-UP-PVD-SCALE

Definition 6

```
>saf>ui>frame.lisp
Type: Function
Arguments: (&OPTIONAL (PVD-DISPLAY *PVD-DISPLAY*))
Outputs:
```

Calls: \*ZOOM-LEVELS\*  
       >map>zoom-levels.lisp  
       \*CURRENT-ZOOM-LEVEL\*  
       >map>zoom-levels.lisp  
       CURRENT-SCALE  
       >map>zoom-levels.lisp  
       CURRENT-ANCHOR-X  
       >map>zoom-levels.lisp  
       CURRENT-ANCHOR-Y  
       >map>zoom-levels.lisp  
       \*ZOOM-LEVELS\*  
       >map>zoom-levels.lisp  
       \*CURRENT-ZOOM-LEVEL\*  
       >map>zoom-levels.lisp  
       \*PVD-DISPLAY\*  
       >saf>sys>vars.lisp  
 Called by: EXPOSE-PVD  
           >saf>ui>frame.lisp  
 Description: None

#### 2.1.4.7 EXPOSE-PVD

Definition 7

      >saf>ui>frame.lisp  
 Type: Function  
 Arguments: (&OPTIONAL (PVD-FRAME \*PVD-FRAME\*))  
 Outputs:  
 Calls: \*PVD-FRAME\*  
       >saf>sys>vars.lisp  
       SET-UP-PVD-SCALE  
       >saf>ui>frame.lisp  
 Called by: (INITIALIZATION WARM-INITIALIZATION-LIST Expose PVD)  
           No Source File Record  
           WARM-INITIALIZATION-LIST  
           >rel-7-2>sys>ltop.lisp  
 Description: None

#### 2.1.4.8 Expose PVD

Definition 8

      >saf>ui>frame.lisp  
 Type: ADD-INITIALIZATION  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

**2.1.4.9 CLEAR-SAF-HISTORY**

Definition 9

>saf>ui>frame.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*OPFOR-IO\*  
>saf>sys>vars.lisp  
Called by: (INITIALIZATION BEFORE-COLD-INITIALIZATION-LIST Clear SAF History)  
No Source File Record  
Description: None

**2.1.4.10 Clear SAF History**

Definition 10

>saf>ui>frame.lisp  
Type: ADD-INITIALIZATION  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.1.4.11 (SET-HIGHLIGHTED-PRESENTATION MAP-WINDOW)**

Definition 11

>saf>ui>frame.lisp  
Type: DEFWHOPPER  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.1.4.12 (WHO-LINE-DOCUMENTATION-STRING MAP-WINDOW)**

Definition 12

>saf>ui>frame.lisp  
Type: DEFWHOPPER  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None



**2.1.4.13 SAF**

## Definition 13

```

>saf>ui>frame.lisp
Type: DEFINE-PROGRAM-FRAMEWORK
Arguments: ()
Outputs:
Calls: PVD
       >saf>ui>frame.lisp
       PVD
       >saf>ui>frame.lisp
Called by: GET-FORMATION-DATA
          >saf>sys>interim-model.lisp
          (METHOD TOP-LEVEL CONFIGURATION-MENU)
          >saf>interface>formations.lisp
          (METHOD TOP-LEVEL OBJECT-MENU)
          >saf>interface>object-menu.lisp
          (METHOD TOP-LEVEL MODEL-MENU)
          >saf>interface>model-menu.lisp
          READ-AND-MAKE-INSTANCES
          >saf>sys>new-storage.lisp
          MKATOM
          >saf>sys>new-storage.lisp
          LEFT-ON- CREATE UNITS -AT-BATTLEMASTER-SAF-MENU-COMMAND
          >saf>bmi>commands.lisp
          LEFT-ON- LOAD SELECTIONS -AT-BATTLEMASTER-SAF-MENU-
COMMAND
          >saf>bmi>commands.lisp
          LEFT-ON- SAVE SELECTIONS -AT-BATTLEMASTER-SAF-MENU-
COMMAND
          >saf>bmi>commands.lisp
          LEFT-ON- RESTORE EXERCISE -AT-BATTLEMASTER-SAF-MENU-
COMMAND
          >saf>bmi>commands.lisp
          LEFT-ON- CLEAR SELECTIONS -AT-BATTLEMASTER-SAF-MENU-
COMMAND
          >saf>bmi>commands.lisp
          LEFT-ON- SELECT UNITS -AT-BATTLEMASTER-SAF-MENU-COMMAND
          >saf>bmi>commands.lisp
          DEFINE-SAF-COMMAND
          >saf>ui>frame.lisp
          MAYBE-LOAD-FORMATION-DATA
          >saf>bmi>utilities.lisp
          OPFOR-SYMBOL
          >saf>bmi>utilities.lisp
          CONVERT-ALIGNMENT
          >saf>bmi>utilities.lisp
          CONVERT-UNIT-SIZE
          >saf>bmi>utilities.lisp
          DRAW-SANDBOX-UNIT
          >saf>sandbox>sandbox-object.lisp
          ERASE-SANDBOX-OBJECT
          >saf>sandbox>sandbox-object.lisp
          DRAW-SANDBOX-OBJECT

```

```

>saf>sandbox>sandbox-object.lisp
(METHOD INSERT-POINT-AFTER GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD INITIALIZE-POINTS GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD INSERT-POINT-BEFORE LINE)
>saf>cm>line.lisp
(METHOD INSERT-POINT-AFTER LINE)
>saf>cm>line.lisp
(METHOD INITIALIZE-POINTS LINE)
>saf>cm>line.lisp
(METHOD INSERT-POINT-BEFORE ROUTE)
>saf>cm>route.lisp
(METHOD INSERT-POINT-AFTER ROUTE)
>saf>cm>route.lisp
(METHOD INITIALIZE-POINTS ROUTE)
>saf>cm>route.lisp
NET-MSG
>saf>rudp>outgoing.lisp
DEF-PACKET-HANDLER
>saf>rudp>handle-incoming.lisp
DEFSTRUCT-SLOT-VAL-PAIRS
>saf>sys>reader-macros.lisp
SANDBOX-READER-MACRO
>saf>sys>reader-macros.lisp

```

Description: None

#### 2.1.4.14 (METHOD MAKE-INSTANCE SAF AFTER)

Definition 14

```

>saf>ui>frame.lisp
Type: Method
Arguments: (&REST INIT-ARGS)
Outputs:
Calls: *PVD-FRAME*
>saf>sys>vars.lisp
*OPFOR-FRAME*
>saf>sys>vars.lisp
*SANDBOX*
>saf>sys>vars.lisp
*RUDP-PROCESS-LAST-CYCLE*
>saf>ui>processes.lisp
PROCESS-RUDP-PACKETS
>saf>ui>processes.lisp
MAKE-RUDP-PROCESS
>saf>ui>processes.lisp
MAKE-UPDATE-PROCESS
>saf>ui>processes.lisp

```

**\*UPDATE-PROCESS-LAST-CYCLE\***

>saf>sys>update-process.lisp

UPDATE-TOP-LEVEL

>saf>sys>update-process.lisp

SANDBOX

>saf>sandbox>sandbox.lisp

Called by: None

Description: None

#### 2.1.4.15 (METHOD TOP-LEVEL SAF)

Definition 15

>saf>ui>frame.lisp

Type: Method

Arguments: (&REST ARGS)

Outputs:

Calls: **\*QUAD-TREE\***

>map>terrain-vars.lisp

**\*ZOOM-LEVELS\***

>map>zoom-levels.lisp

**\*CURRENT-ZOOM-LEVEL\***

>map>zoom-levels.lisp

**\*ZOOM-LEVELS\***

>map>zoom-levels.lisp

**\*CURRENT-ZOOM-LEVEL\***

>map>zoom-levels.lisp

**\*PVD-FRAME\***

>saf>sys>vars.lisp

**\*PVD-DISPLAY\***

>saf>sys>vars.lisp

**\*PVD-LEGEND\***

>saf>sys>vars.lisp

**\*OPFOR-FRAME\***

>saf>sys>vars.lisp

**\*OPFOR-IO\***

>saf>sys>vars.lisp

**\*RADIO-OUTPUT\***

>saf>sys>vars.lisp

**\*BMI-PROGRAM\***

>saf>sys>vars.lisp

**\*INTERFACE-TO-UPDATE-PROCESS-QUEUE\***

>saf>sys>vars.lisp

GET-LOCAL-HOST-SAF-PORT

>saf>network>vars.lisp

**\*RUDP-OUTPUT-STREAM\***

>saf>rudp>vars.lisp

**\*RUDP-OUTPUT-STREAM\***

>saf>rudp>vars.lisp

MOUSE-FLIP-SCREEN

>saf>ui>mouse-interface.lisp

FIND-MOUSE

>saf>ui>mouse-interface.lisp

**\*DEFAULT-BATTALION-NUMBER\***

```

>saf>bmi>bmi-frame.lisp
MAKE-PVD-FRAME
>saf>ui>frame.lisp
SETUP-COLOR-ALUS
>saf>color-window>color-alus.lisp

```

Called by: None

Description: None

#### 2.1.4.16 (METHOD GET-RUDP-PROCESS PROGRAM-FRAME)

Definition 16

```

>saf>ui>frame.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

#### 2.1.4.17 (METHOD GET-UPDATE-PROCESS PROGRAM-FRAME)

Definition 17

```

>saf>ui>frame.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

### 2.1.5 CSU ui>commands.lisp

This unit contains the menu commands for the PVD, definitions of command processor commands available to the SAF user, and the presentation action for the PVD command menu. The macro `define-pvd-menu-command` encapsulates the code needed to create new PVD menu commands, and automatically highlights them. This macro is then called to define a number of such commands.

PVD menu commands include zooming, panning, scale selection, refresh, and terrain options. CP commands, including, for example, unit-ops, batallion-ops, clear-message-log, and set-viewport, are described in the SAF User's Guide.

#### 2.1.5.1 DEFINE-PVD-MENU-COMMAND

Definition 1

```

>saf>ui>commands.lisp
Type: Macro
Arguments: ((NAME PRETTY-NAME) &BODY BODY)
Outputs:

```

Calls: HIGHLIGHT-BUTTON  
>saf>ui>frame-utils.lisp  
PVD  
>saf>ui>frame.lisp  
PVD  
>saf>ui>frame.lisp  
DEFINE-PVD-MENU-COMMAND  
>saf>ui>commands.lisp  
Called by: DEFINE-PVD-MENU-COMMAND  
>saf>ui>commands.lisp  
Description: None

#### 2.1.5.2 (COM-ZOOM-IN Zoom In ) Definition 2

>saf>ui>commands.lisp  
Type: DEFINE-PVD-MENU-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.1.5.3 (COM-PAN Pan ) Definition 3

>saf>ui>commands.lisp  
Type: DEFINE-PVD-MENU-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.1.5.4 GET-ELEVATION Definition 4

>saf>ui>commands.lisp  
Type: Function  
Arguments: (X Y)  
Outputs:  
Calls: HEIGHT-AT-POINT  
>map>draw-terrain.lisp  
\*OPFOR-IO\*  
>saf>sys>vars.lisp  
SAY  
>saf>sys>macros.lisp  
FORMAT-COORDINATES  
>saf>sys>utilities.lisp

Called by: (METHOD COM-PAN-INTERNAL PVD)  
No Source File Record  
Description: Determine Elevation

#### 2.1.5.5 (COM-ZOOM-OUT Zoom Out ) Definition 5

>saf>ui>commands.lisp  
Type: DEFINE-PVD-MENU-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.1.5.6 RESCALE-PVD-FROM-MENU Definition 6

>saf>ui>commands.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*ZOOM-LEVELS\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
SCALE-STRING  
>map>zoom-levels.lisp  
\*ZOOM-LEVELS\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: (METHOD COM-RESCALE-INTERNAL PVD)  
No Source File Record  
Description: None

#### 2.1.5.7 (COM-RESCALE Map Scale ) Definition 7

>saf>ui>commands.lisp  
Type: DEFINE-PVD-MENU-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.1.5.8 (COM-PLFRESH Refresh )**

## Definition 8

```
>saf>ui>commands.lisp
Type: DEFINE-PVD-MENU-COMMAND
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.1.5.9 (COM-TERRAIN-OPTIONS Terrain Options )**

## Definition 9

```
>saf>ui>commands.lisp
Type: DEFINE-PVD-MENU-COMMAND
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.1.5.10 PAN-TO-POINT**

## Definition 10

```
>saf>ui>commands.lisp
Type: Function
Arguments: ()
Outputs:
Calls: *OPFOR-IO*
      >saf>sys>vars.lisp
      *INTERFACE-TO-UPDATE-PROCESS-QUEUE*
      >saf>sys>vars.lisp
      ADD-TO-UPDATE-QUEUE
      >saf>sys>macros.lisp
      SAY
      >saf>sys>macros.lisp
      PARSE-COORDS
      >saf>ui>commands.lisp
Called by: COM-PAN-TO-POINT
          >saf>ui>commands.lisp
Description: None
```

**2.1.5.11 PARSE-COORDS**

## Definition 11

```
>saf>ui>commands.lisp
Type: Function
Arguments: (COORD-STRING)
Outputs:
```

Calls: \*PVD-DISPLAY\*  
>saf>sys>vars.lisp  
Called by: PAN-TO-POINT  
>saf>ui>commands.lisp  
(PRESENTATION-FUNCTION WORLD-COORDS PARSER)  
No Source File Record  
Description: None

### 2.1.5.12 COM-PAN-TO-POINT

Definition 12

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: PAN-TO-POINT  
>saf>ui>commands.lisp  
Called by: None  
Description: None

### 2.1.5.13 COM-UNIT-OPS

Definition 13

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

### 2.1.5.14 COM-BATTALION-OPS

Definition 14

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: RUN-BATTALION-OPS  
>saf>ui>task-org.lisp  
Called by: None  
Description: None

### 2.1.5.15 COM-REFRESH-UNIT-DISPLAY

Definition 15

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:



Calls: DISPLAY-TASK-ORG  
>saf>ui>task-org.lisp  
Called by: None  
Description: None

### 2.1.5.16 COM-CLEAR-MESSAGE-LOG

Definition 16

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: \*OPFOR-FRAME\*  
>saf>sys>vars.lisp  
Called by: None  
Description: None

### 2.1.5.17 COM-CLEAR

Definition 17

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: STANDALONEP  
>saf>network>connection.lisp  
COMPLETE-C2-RESET  
>saf>network>top-level.lisp  
RESET-SIM  
>saf>network>top-level.lisp  
Called by: None  
Description: None

### 2.1.5.18 COM-SET-VIEWPORT

Definition 18

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: \*VIEW-VEHICLE-ID\*  
>saf>sys>vars.lisp  
UNHIGHLIGHT-VIEWPORTS  
>saf>objects>simnet-agent.lisp  
HIGHLIGHT-VIEWPORTS  
>saf>objects>simnet-agent.lisp  
Called by: None  
Description: None

**2.1.5.19 COM-BOMB-BUTTON**

Definition 19

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: BOMB-BUTTON  
>saf>network>commands.lisp  
Called by: None  
Description: None

**2.1.5.20 COM-SAF-SET-BOMB-PARAMETERS**

Definition 20

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: SET-BOMB-PARAMETERS  
>saf>network>commands.lisp  
Called by: None  
Description: None

**2.1.5.21 COM-ROBO-COP-CONTROL**

Definition 21

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: ROBO-COP-CONTROL  
>saf>ui>parameter-menus.lisp  
Called by: None  
Description: None

**2.1.5.22 COM-SET-OPFOR-PARAMETERS**

Definition 22

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: ROBO-COP-CONTROL  
>saf>ui>parameter-menus.lisp  
Called by: None  
Description: None

**2.1.5.23 COM-SAVE-SCENARIO**

## Definition 23

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: NAME-AND-STORE-SCENARIO  
>saf>sys>new-storage.lisp  
Called by: None  
Description: None

**2.1.5.24 COM-DELETE-SCENARIOS**

## Definition 24

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: CHOOSE-SCENARIOS-TO-DELETE  
>saf>sys>new-storage.lisp  
Called by: None  
Description: None

**2.1.5.25 COM-DELETE-EXERCISES**

## Definition 25

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: CHOOSE-SCENARIOS-TO-DELETE  
>saf>sys>new-storage.lisp  
Called by: None  
Description: None

**2.1.5.26 COM-DELETE-OVERLAYS**

## Definition 26

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: CHOOSE-OVERLAYS-TO-DELETE  
>saf>sys>new-storage.lisp  
Called by: None  
Description: None

**2.1.5.27 COM-STORE-SCENARIO**

## Definition 27

>saf>ui>commands.lisp  
Type: CP Command  
Arguments: ()  
Outputs:  
Calls: NAME-AND-STORE-SCENARIO  
>saf>sys>new-storage.lisp  
Called by: None  
Description: None

**2.1.5.28 SUPERIOR-CONTEXT**

## Definition 28

>saf>ui>commands.lisp  
Type: Function  
Arguments: (CONTEXT)  
Outputs:  
Calls: None  
Called by: ROOT-INPUT-CONTEXT  
>saf>ui>commands.lisp  
Description: None

**2.1.5.29 ROOT-INPUT-CONTEXT**

## Definition 29

>saf>ui>commands.lisp  
Type: Function  
Arguments: (CONTEXT)  
Outputs:  
Calls: SUPERIOR-CONTEXT  
>saf>ui>commands.lisp  
ROOT-INPUT-CONTEXT  
>saf>ui>commands.lisp  
Called by: (PRESENTATION-MOUSE-HANDLER PVD-COMMAND-MENU)  
No Source File Record  
ROOT-INPUT-CONTEXT  
>saf>ui>commands.lisp  
Description: None

**2.1.5.30 PVD-COMMAND-MENU**

## Definition 30

>saf>ui>commands.lisp  
Type: DEFINE-PRESENTATION-ACTION  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

## 2.2 COMMANDER CSC

The Commander CSC contains the code to allow the SAF operator to control his units and vehicles. The display window on the monochrome monitor in the commander mode contains a task organization pane, a message log pane, an OPORD pane, and a lisp interaction pane. The OPORD pane and the task organization pane are described below. The message log displays any commands which the user has issued to the own units and any reports which have come from the own units. The reports are written to the message log directly by the RUDP process when the messages are received. The commands are written by the user interface process as the commands are sent over to the RUDP process for transmission. The lisp interaction pane is used to type in text commands such as the command to go into battlemaster mode and the password required to enter that mode. Figure 2.2-1 shows the sub-level CSCs of the Commander CSC.

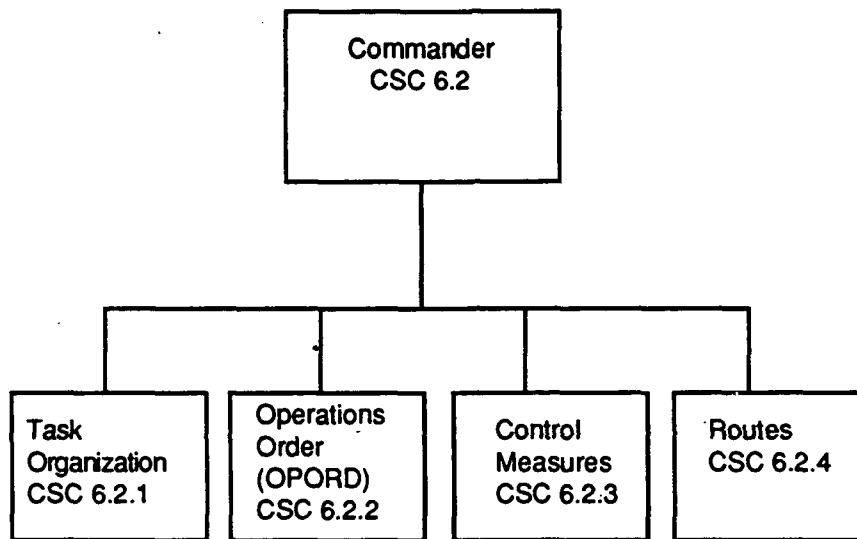


Figure 2.2-1 CSCs of the Commander CSC

### 2.2.1 Task Organization CSC

This CSC contains the code to run the task organization pane. This code is invoked by the user process in response to user commands. The task organization pane displays the hierarchy of the forces commanded by this workstation. The hierarchy is displayed in graphical form using the military symbols for the units. The display of the hierarchy can be truncated at any level. By clicking the middle or right mouse button on a unit, you access a number of menu items which allow you to change the level of the display or the position of the units on the task organization pane. You can also issue commands to the stealth vehicle (used for the out-the-window view) to change its display to that unit's out the window view.

The effect of clicking left on a unit in the task organization display is determined by the current OPORD mode. This mode is selected by clicking left on the appropriate button on the OPORD pane. The modes are TAC/E, status, and subordinate unit tasking. In the TAC/E mode, clicking on a unit allows you to give quick commands to the units, in the

subordinate unit tasking mode it allows you to assign graphical orders to subunits, and in the status mode it requests the unit to send back a status report which will be displayed in the message log. This CSC contains the following CSU:

```
ui>task-org.lisp csu
```

### 2.2.1.1 CSU ui>task-org.lisp

This unit contains the software behind the task-organization pane on the monochrome display. It includes the definition of the *task-org-pane* flavor, and related methods and functions for presentation and display. Battalion presentation and associated functions are also included.

#### 2.2.1.1.1 WORKSTATION-BATTALION

Definition 1

```
>saf>ui>task-org.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: (PRESENTATION-MOUSE-HANDLER MOUSE-WORKSTATION-
BATTALION)
No Source File Record
DISPLAY-WORKSTATION-BATTALION
>saf>ui>task-org.lisp
(PROPERTY WORKSTATION-BATTALION DEFTYPE)
No Source File Record
Description: None
```

#### 2.2.1.1.2 MOUSE-WORKSTATION-BATTALION

Definition 2

```
>saf>ui>task-org.lisp
Type: DEFINE-PRESENTATION-TO-COMMAND-TRANSLATOR
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.2.1.1.3 RUN-BATTALION-OPS

Definition 3

```
>saf>ui>task-org.lisp
Type: Function
Arguments: (BUTTON)
Outputs:
```

Calls: MENU-CHOOSE  
      >saf>sys>utilities.lisp  
      OPORD  
      >saf>ui>opord.lisp  
Called by: COM-BATTALION-OPS  
      >saf>ui>commands.lisp  
Description: None

#### 2.2.1.1.4 DISPLAY-WORKSTATION-BATTALION

Definition 4

      >saf>ui>task-org.lisp  
Type: Function  
Arguments: (PROGRAM STREAM)  
Outputs:  
Calls: WORKSTATION-BATTALION  
      >saf>ui>task-org.lisp  
      GET-BATTALION-NUMBER  
      >saf>bmi>bmi-frame.lisp  
Called by: SAF PROGRAM-FRAME-OPTIONS  
      >saf>ui>frame.lisp  
Description: None

#### 2.2.1.1.5 DISPLAY-FOR-TASK-ORG

Definition 5

      >saf>ui>task-org.lisp  
Type: Function  
Arguments: (UNIT STREAM X Y)  
Outputs:  
Calls: \*UNIT-ICON-TABLE\*  
      >saf>simnet-objects>draw-units.lisp  
      UNIT-ICON  
      >saf>simnet-objects>draw-units.lisp  
Called by: (METHOD DRAW-TASK-ORGANIZATION TASK-ORG-PANE)  
      >saf>ui>task-org.lisp  
Description: None

#### 2.2.1.1.6 INFERIORS-FOR-TASK-ORG

Definition 6

      >saf>ui>task-org.lisp  
Type: Function  
Arguments: (UNIT)  
Outputs:  
Calls: VEH-DESTROYED  
      >saf>sys>constants.lisp  
      VEHICLE-STATUS  
      >saf>network>packet-layouts.lisp  
      IS-STATUS

>saf>simnet-objects>macros.lisp  
GET-SUBORDINATES-INSTANCES  
>saf>objects>simnet-agent.lisp  
Called by: (METHOD DRAW-TASK-ORGANIZATION TASK-ORG-PANE)  
>saf>ui>task-org.lisp  
Description: None

#### 2.2.1.1.7 HIGHLIGHT-ON-TASK-ORG

Definition 7

>saf>ui>task-org.lisp  
Type: Function  
Arguments: (UNIT)  
Outputs:  
Calls: \*OPFOR-FRAME\*  
>saf>sys>vars.lisp  
Called by: HIGHLIGHT-VIEWPORTS  
>saf>objects>simnet-agent.lisp  
UNHIGHLIGHT-VIEWPORTS  
>saf>objects>simnet-agent.lisp  
Description: None

#### 2.2.1.1.8 TASK-ORG-PANE

Definition 8

>saf>ui>task-org.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.2.1.1.9 (SET-HIGHLIGHTED-PRESENTATION TASK-ORG-PANE)

Definition 9

>saf>ui>task-org.lisp  
Type: DEFWHOPPER  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None



**2.2.1.1.10 (METHOD SET-HIGHLIGHTED-PRESENTATION TASK-ORG-PANE AFTER)**

Definition 10

```

>saf>ui>task-org.lisp
Type: Method
Arguments: (PRESENTATION &OPTIONAL DOCUMENTATION SHIFTS MORE-DOCUMENTATION)
Outputs:
Calls: SIMNET-AGENT
       >saf>objects>simnet-agent.lisp
       SIMNET-AGENT
       >saf>objects>simnet-agent.lisp
       SIMNET-AGENT
       >saf>objects>simnet-agent.lisp
Called by: None
Description: None

```

**2.2.1.1.11 DISPLAY-TASK-ORG**

Definition 11

```

>saf>ui>task-org.lisp
Type: Function
Arguments: ()
Outputs:
Calls: *OPFOR-FRAME*
       >saf>sys>vars.lisp
Called by: COM-REFRESH-UNIT-DISPLAY
          >saf>ui>commands.lisp
          DELAYED-DISPLAY-UNIT-GRAPH-1
          >saf>simnet-objects>vehicle-tracking.lisp
          (METHOD HIDE-INFERIORS SIMNET-AGENT)
          >saf>objects>simnet-agent.lisp
          (METHOD SHOW-INFERIORS SIMNET-AGENT)
          >saf>objects>simnet-agent.lisp
          (METHOD MOUSE-GESTURE-ITEM-LIST SIMNET-AGENT APPEND)
          >saf>objects>simnet-agent.lisp
Description: None

```

**2.2.1.1.12 (METHOD DRAW-TASK-ORGANIZATION TASK-ORG-PANE)**

Definition 12

```

>saf>ui>task-org.lisp
Type: Method
Arguments: ()
Outputs:
Calls: LOCAL
       >saf>network>vars.lisp
       TOP-LEVEL-UNITS
       >saf>simnet-objects>vehicle-tracking.lisp

```

DISPLAY-FOR-TASK-ORG  
 >saf>ui>task-org.lisp  
 INFERIORS-FOR-TASK-ORG  
 >saf>ui>task-org.lisp

Called by: None  
 Description: None

#### 2.2.1.1.13 (METHOD DRAW-TASK-ORGANIZATION TASK-ORG-PANE AFTER)

Definition 13

>saf>ui>task-org.lisp  
 Type: Method  
 Arguments: ()  
 Outputs:  
 Calls: HIGHLIGHT-VIEWPORTS  
 >saf>objects>simnet-agent.lisp  
 Called by: None  
 Description: None

#### 2.2.1.1.14 TASK-ORG-PANE

Definition 14

>saf>ui>task-org.lisp  
 Type: COMPILE-FLAVOR-METHODS  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: SAF PROGRAM-FRAME-OPTIONS  
 >saf>ui>frame.lisp  
 Description: None

### 2.2.2 Operations Order (OPORD) CSC

This CSC contains the code to define the mouse buttons which change the OPORD mode plus the code to implement the OPORD commands when one button is selected. The OPORD pane contains buttons to place the command frame into the TAC/E, subordinate unit tasking, operations order, or status mode. In addition, it contains buttons for saving the exercise state or the overlays which have been created. This CSC has the code to present a menu with TAC/E commands and then execute the chosen command. It also has the code to put up the subordinate unit tasking menu and generate orders for subordinated units. It also contains the interface to the overlay code. This CSC contains the following CSUs:

ui>opord.lisp csu  
 objects>intervention.lisp csu  
 ui>subordinate-tasking.lisp csu  
 cm>overlay.lisp csu

**2.2.2.1 CSU ui>opord.lisp**

This unit contains the data-structures and routines that implement the Oporder Metaphor window. It defines flavors for paragraphs, subparagraphs, and operations buttons, along with their associated methods. SAF flavor methods are defined to display paragraphs, operations, and opord choices. SAF commands for selecting subparagraphs and ops-buttons are also defined. Finally, *opord*, the top-level function for the Operations Order system, is defined.

**2.2.2.1.1 \*OPORD-MODE\***

Definition 1

```

>saf>ui>opord.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: OPORD
           >saf>ui>opord.lisp
           (METHOD COM-SELECT-BUTTON-INTERNAL SAF)
           No Source File Record
           (METHOD COM-SELECT-SUBPARAGRAPH-INTERNAL SAF)
           No Source File Record
Description: None

```

**2.2.2.1.2 \*ENABLED-FONT\***

Definition 2

```

>saf>ui>opord.lisp
Type: Parameter
Arguments: ()
Outputs:
Calls: None
Called by: (METHOD FONT OPORD-BUTTON)
           >saf>ui>opord.lisp
Description: None

```

**2.2.2.1.3 \*DISABLED-FONT\***

Definition 3

```

>saf>ui>opord.lisp
Type: Parameter
Arguments: ()
Outputs:
Calls: None
Called by: (METHOD DISPLAY PARAGRAPH)
           >saf>ui>opord.lisp
           (METHOD FONT OPORD-BUTTON)
           >saf>ui>opord.lisp
Description: None

```

**2.2.2.1.4 \*PREVIOUS-BUTTON-BOX\***

Definition 4

>saf>ui>opord.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (METHOD DISPLAY-OPORD-CHOICES SAF)  
>saf>ui>opord.lisp  
(METHOD HIGHLIGHT OPORD-BUTTON)  
>saf>ui>opord.lisp  
Description: None

**2.2.2.1.5 OPORD-BUTTON**

Definition 5

>saf>ui>opord.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.1.6 (METHOD FONT OPORD-BUTTON)**

Definition 6

>saf>ui>opord.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: \*ENABLED-FONT\*  
>saf>ui>opord.lisp  
\*DISABLED-FONT\*  
>saf>ui>opord.lisp  
Called by: None  
Description: None

**2.2.2.1.7 (METHOD HIGHLIGHT OPORD-BUTTON)**

Definition 7

>saf>ui>opord.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: \*OPFOR-FRAME\*  
>saf>sys>vars.lisp  
\*PREVIOUS-BUTTON-BOX\*  
>saf>ui>opord.lisp

Called by: None  
Description: None

### 2.2.2.1.8 OPORD-BUTTON

Definition 8

>saf>ui>opord.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: OPS-BUTTON  
>saf>ui>opord.lisp  
SUBPARAGRAPH  
>saf>ui>opord.lisp  
Description: None

### 2.2.2.1.9 PARAGRAPH

Definition 9

>saf>ui>opord.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

### 2.2.2.1.10 MAKE-PARAGRAPH

Definition 10

>saf>ui>opord.lisp  
Type: Function  
Arguments: (&KEY NAME SUBPARAGRAPHS)  
Outputs:  
Calls: NAME  
>saf>sysdcl.lisp  
PARAGRAPH  
>saf>ui>opord.lisp  
Called by: None  
Description: None

### 2.2.2.1.11 (METHOD DISPLAY PARAGRAPH)

Definition 11

>saf>ui>opord.lisp  
Type: Method  
Arguments: (STREAM COLUMN)  
Outputs:

Calls: \*DISABLED-FONT\*  
>saf>ui>opord.lisp  
Called by: None  
Description: None

#### 2.2.2.1.12 SUBPARAGRAPH

Definition 12

>saf>ui>opord.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: OPORD-BUTTON  
>saf>ui>opord.lisp  
Called by: None  
Description: None

#### 2.2.2.1.13 MAKE-SUBPARAGRAPH

Definition 13

>saf>ui>opord.lisp  
Type: Function  
Arguments: (&KEY NAME KEYWORD ENABLED)  
Outputs:  
Calls: NAME  
>saf>sysdcl.lisp  
SUBPARAGRAPH  
>saf>ui>opord.lisp  
Called by: None  
Description: None

#### 2.2.2.1.14 (METHOD DISPLAY SUBPARAGRAPH)

Definition 14

>saf>ui>opord.lisp  
Type: Method  
Arguments: (STREAM COLUMN)  
Outputs:  
Calls: SUBPARAGRAPH  
>saf>ui>opord.lisp  
Called by: None  
Description: None

#### 2.2.2.1.15 PARAGRAPH

Definition 15

>saf>ui>opord.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:

Calls: None  
Called by: MAKE-PARAGRAPH  
>saf>ui>opord.lisp  
Description: None

#### 2.2.2.1.16 SUBPARAGRAPH

Definition 16

>saf>ui>opord.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (PRESENTATION-MOUSE-HANDLER SELECT-SUBPARAGRAPH)  
No Source File Record  
(METHOD COM-SELECT-SUBPARAGRAPH-PARSER SAF)  
No Source File Record  
(METHOD DISPLAY SUBPARAGRAPH)  
>saf>ui>opord.lisp  
MAKE-SUBPARAGRAPH  
>saf>ui>opord.lisp  
Description: None

#### 2.2.2.1.17 \*PARAGRAPH-DATA\*

Definition 17

>saf>ui>opord.lisp  
Type: Parameter  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (METHOD DISPLAY-PARAGRAPHS SAF)  
>saf>ui>opord.lisp  
Description: None

#### 2.2.2.1.18 OPS-BUTTON

Definition 18

>saf>ui>opord.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: OPORD-BUTTON  
>saf>ui>opord.lisp  
Called by: None  
Description: None

**2.2.2.1.19 MAKE-OPS-BUTTON**

Definition 19

&gt;saf&gt;ui&gt;opord.lisp

Type: Function

Arguments: (&amp;KEY NAME KEYWORD ENABLED)

Outputs:

Calls: NAME

&gt;saf&gt;sysdcl.lisp

OPS-BUTTON

&gt;saf&gt;ui&gt;opord.lisp

OPS-BUTTON

&gt;saf&gt;ui&gt;opord.lisp

Called by: None

Description: None

**2.2.2.1.20 (METHOD DISPLAY OPS-BUTTON)**

Definition 20

&gt;saf&gt;ui&gt;opord.lisp

Type: Method

Arguments: (STREAM COLUMN)

Outputs:

Calls: OPS-BUTTON

&gt;saf&gt;ui&gt;opord.lisp

OPS-BUTTON

&gt;saf&gt;ui&gt;opord.lisp

Called by: None

Description: None

**2.2.2.1.21 OPS-BUTTON**

Definition 21

&gt;saf&gt;ui&gt;opord.lisp

Type: COMPILE-FLAVOR-METHODS

Arguments: ()

Outputs:

Calls: None

Called by: (PRESENTATION-MOUSE-HANDLER SELECT-OPS-BUTTON)

No Source File Record

(METHOD COM-SELECT-BUTTON-PARSER SAF)

No Source File Record

(METHOD DISPLAY OPS-BUTTON)

&gt;saf&gt;ui&gt;opord.lisp

MAKE-OPS-BUTTON

&gt;saf&gt;ui&gt;opord.lisp

Description: None



**2.2.2.1.22 OPS-BUTTON**

Definition 22

```
>saf>ui>opord.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: (PRESENTATION-MOUSE-HANDLER SELECT-OPS-BUTTON)
           No Source File Record
           (METHOD COM-SELECT-BUTTON-PARSER SAF)
           No Source File Record
           (METHOD DISPLAY OPS-BUTTON)
>saf>ui>opord.lisp
MAKE-OPS-BUTTON
>saf>ui>opord.lisp
Description: None
```

**2.2.2.1.23 \*OPERATIONS-BUTTONS\***

Definition 23

```
>saf>ui>opord.lisp
Type: Parameter
Arguments: ()
Outputs:
Calls: None
Called by: (METHOD DISPLAY-OPERATIONS SAF)
           >saf>ui>opord.lisp
Description: None
```

**2.2.2.1.24 (METHOD DISPLAY-PARAGRAPHS SAF)**

Definition 24

```
>saf>ui>opord.lisp
Type: Method
Arguments: (STREAM COLUMN)
Outputs:
Calls: *PARAGRAPH-DATA*
           >saf>ui>opord.lisp
Called by: None
Description: None
```

**2.2.2.1.25 (METHOD DISPLAY-OPERATIONS SAF)**

Definition 25

```
>saf>ui>opord.lisp
Type: Method
Arguments: (STREAM COLUMN)
Outputs:
```

**Calls: \*OPERATIONS-BUTTONS\***

>saf>ui>opord.lisp

Called by: None

Description: None

#### **2.2.2.1.26 (METHOD DISPLAY-OPORD-CHOICES SAF)**

Definition 26

>saf>ui>opord.lisp

Type: Method

Arguments: (STREAM)

Outputs:

**Calls: \*OPFOR-FRAME\***

>saf>sys>vars.lisp

**\*PREVIOUS-BUTTON-BOX\***

>saf>ui>opord.lisp

Called by: None

Description: None

#### **2.2.2.1.27 (COM-SELECT-SUBPARAGRAPH)**

Definition 27

>saf>ui>opord.lisp

Type: DEFINE-SAF-COMMAND

Arguments: ()

Outputs:

Called by: None

Called by: None

Description: None

#### **2.2.2.1.28 SELECT-SUBPARAGRAPH**

Definition 28

>saf>ui>opord.lisp

Type: DEFINE-PRESENTATION-TO-COMMAND-TRANSLATOR

Arguments: ()

Outputs:

Called by: None

Called by: None

Description: None

#### **2.2.2.1.29 (COM-SELECT-BUTTON)**

Definition 29

>saf>ui>opord.lisp

Type: DEFINE-SAF-COMMAND

Arguments: ()

Outputs:

Calls: None  
 Called by: None  
 Description: None

### 2.2.2.1.30 SELECT-OPS-BUTTON

Definition 30

>saf>ui>opord.lisp  
 Type: DEFINE-PRESENTATION-TO-COMMAND-TRANSLATOR  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

### 2.2.2.1.31 OPORD

Definition 31

>saf>ui>opord.lisp  
 Type: Function  
 Arguments: (UNIT)  
 Outputs:  
 Calls: SIMNET-AGENT  
       >saf>objects>simnet-agent.lisp  
       SIMNET-AGENT  
       >saf>objects>simnet-agent.lisp  
       SIMNET-AGENT  
       >saf>objects>simnet-agent.lisp  
       \*OPORD-MODE\*  
       >saf>ui>opord.lisp  
       SUBORDINATE-TASK  
       >saf>ui>subordinate-tasking.lisp  
 Called by: RUN-BATTALION-OPS  
       >saf>ui>task-org.lisp  
       (METHOD MOUSE-GESTURE SIMNET-AGENT)  
       >saf>objects>simnet-agent.lisp  
 Description: None

### 2.2.2.2 CSU objects>intervention.lisp

This unit defines the function *intervene* that implements immediate interventions on *simnet-agent* instances. Because there are 17 different intervention types, *intervene* is defined using the Common Lisp macro *defgeneric*, to make available the Symbolics *:case* keyword. This allows the function to be broken up into separate methods as cases, based on the value of the *intervention-type* argument. This was done simply to make the definition lexically more modular; otherwise it would have been a single 4-page-long form.

Methods for *intervene* define intervention responses for the intervention-types rules-of-engagement, face-direction, halt, hold, enroute-movement, speed, altitude, follow-vehicle, command-from-simulator, go-to-location, resupply, land, attack, resume, resume-all-subordinates, rejoin-unit, and formation.

**2.2.2.2.1 INTERVENE**

## Definition 1

>saf>objects>intervention.lisp  
Type: DEFGENERIC  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (METHOD IMMEDIATE-INTERVENTION SIMNET-AGENT)  
>saf>objects>simnet-agent.lisp  
Description: None

**2.2.2.2.2 (METHOD INTERVENE SIMNET-AGENT OTHERWISE)**

## Definition 2

>saf>objects>intervention.lisp  
Type: Method  
Arguments: (UNMATCHED-CASE)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.2.3 (METHOD INTERVENE SIMNET-AGENT RULES-OF-ENGAGEMENT)**

## Definition 3

>saf>objects>intervention.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.2.4 (METHOD INTERVENE SIMNET-AGENT FACE-DIRECTION)**

## Definition 4

>saf>objects>intervention.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: FACE-DIRECTION  
>saf>objects>simnet-agent.lisp  
FACE-DIRECTION  
>saf>objects>simnet-agent.lisp  
Called by: None  
Description: None

**2.2.2.2.5 (METHOD INTERVENE SIMNET-AGENT HALT)**

## Definition 5

>saf>objects>intervention.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: HALT  
>saf>network>vars.lisp  
NET-MSG  
>saf>rudp>outgoing.lisp  
Called by: None  
Description: None

**2.2.2.2.6 (METHOD INTERVENE SIMNET-AGENT HOLD)**

## Definition 6

>saf>objects>intervention.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: HOLD-HOVER  
>saf>sys>vars.lisp  
HOLD-ORBIT  
>saf>sys>vars.lisp  
MENU-CHOOSE  
>saf>sys>utilities.lisp  
HOLD  
>saf>network>vars.lisp  
NET-MSG  
>saf>rudp>outgoing.lisp  
Called by: None  
Description: None

**2.2.2.2.7 (METHOD INTERVENE SIMNET-AGENT ENROUTE-MOVEMENT)**

## Definition 7

>saf>objects>intervention.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: MENU-CHOOSE  
>saf>sys>utilities.lisp  
ENROUTE-MOVEMENT  
>saf>network>vars.lisp  
NET-MSG  
>saf>rudp>outgoing.lisp  
Called by: None  
Description: None

**2.2.2.2.8 (METHOD INTERVENE SIMNET-AGENT SPEED)**

Definition 8

>saf>objects>intervention.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: \*LAST-UNITS-SPEED\*  
>saf>sys>vars.lisp  
SPEED-TO-M/SEC  
>saf>sys>utilities.lisp  
CHANGE-SPEED  
>saf>network>vars.lisp  
NET-MSG  
>saf>rudp>outgoing.lisp  
Called by: None  
Description: None

**2.2.2.2.9 (METHOD INTERVENE SIMNET-AGENT ALTITUDE)**

Definition 9

>saf>objects>intervention.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: \*LAST-UNITS-LENGTH\*  
>saf>sys>vars.lisp  
\*LAST-UNITS-ALTITUDE\*  
>saf>sys>vars.lisp  
CHANGE-ALTITUDE  
>saf>network>vars.lisp  
NET-MSG  
>saf>rudp>outgoing.lisp  
Called by: None  
Description: None

**2.2.2.2.10 (METHOD INTERVENE SIMNET-AGENT FOLLOW-VEHICLE)**

Definition 10

>saf>objects>intervention.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: VEC-ROTATE  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
SINGLE-POINT  
>map>control.lisp  
\*PVD-DISPLAY\*  
>saf>sys>vars.lisp

```

*LAST-UNITS-LENGTH*
>saf>sys>vars.lisp
FOLLOW-VEHICLE
>saf>network>vars.lisp
NET-MSG
>saf>rudp>outgoing.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp

```

Called by: None

Description: None

#### 2.2.2.2.11 (METHOD INTERVENE SIMNET-AGENT COMMAND-FROM-SIMULATOR)

Definition 11

```

>saf>objects>intervention.lisp
Type: Method
Arguments: ()
Outputs:
Calls: *PVD-DISPLAY*
>saf>sys>vars.lisp
SIMULATOR-IN-COMMAND
>saf>network>vars.lisp
NET-MSG
>saf>rudp>outgoing.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp

```

Called by: None

Description: None

#### 2.2.2.2.12 (METHOD INTERVENE SIMNET-AGENT GO-TO-LOCATION)

Definition 12

```

>saf>objects>intervention.lisp
Type: Method
Arguments: ()
Outputs:
Calls: SINGLE-POINT
>map>control.lisp
*DEFAULT-OUTPUT-COORDINATE-SYSTEM*
>saf>sys>vars.lisp
*PVD-DISPLAY*
>saf>sys>vars.lisp
HOLD-HOVER

```

```

>saf>sys>vars.lisp
HOLD-ORBIT
>saf>sys>vars.lisp
POINT
>saf>interface>model-menu.lisp
GO-TO-POINT
>saf>network>vars.lisp
NET-MSG
>saf>rudp>outgoing.lisp
WORLD-COORDS
>saf>cm>control-measure.lisp
POINT
>saf>interface>model-menu.lisp

```

Called by: None

Description: None

#### 2.2.2.2.13 (METHOD INTERVENE SIMNET-AGENT RESUPPLY)

Definition 13

```

>saf>objects>intervention.lisp
Type: Method
Arguments: ()
Outputs:
Calls: *PVD-DISPLAY*
>saf>sys>vars.lisp
RESUPPLY
>saf>network>vars.lisp
NET-MSG
>saf>rudp>outgoing.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp

```

Called by: None

Description: None

#### 2.2.2.2.14 (METHOD INTERVENE SIMNET-AGENT LAND)

Definition 14

```

>saf>objects>intervention.lisp
Type: Method
Arguments: ()
Outputs:
Calls: SINGLE-POINT
>map>control.lisp
*PVD-DISPLAY*
>saf>sys>vars.lisp
FORMAT-COORDINATES
>saf>sys>utilities.lisp
POINT

```



```
>saf>interface>model-menu.lisp
LAND
>saf>network>vars.lisp
NET-MSG
>saf>rudp>outgoing.lisp
WORLD-COORDS
>saf>cm>control-measure.lisp
POINT
>saf>interface>model-menu.lisp
```

Called by: None

Description: None

#### 2.2.2.2.15 (METHOD INTERVENE SIMNET-AGENT ATTACK)

Definition 15

```
>saf>objects>intervention.lisp
Type: Method
Arguments: ()
Outputs:
Calls: RUBBER-LINE
>map>control.lisp
*PVD-DISPLAY*
>saf>sys>vars.lisp
FORMAT-COORDINATES
>saf>sys>utilities.lisp
ATTACK
>saf>network>vars.lisp
RUNNING-FIRE-ATTACK
>saf>network>vars.lisp
POP-UP-ATTACK
>saf>network>vars.lisp
NET-MSG
>saf>rudp>outgoing.lisp
WORLD-COORDS
>saf>cm>control-measure.lisp
```

Called by: None

Description: None

#### 2.2.2.2.16 (METHOD INTERVENE SIMNET-AGENT RESUME)

Definition 16

```
>saf>objects>intervention.lisp
Type: Method
Arguments: ()
Outputs:
Calls: RESUME
>saf>network>vars.lisp
NET-MSG
>saf>rudp>outgoing.lisp
```

Called by: None

Description: None

**2.2.2.2.17 (METHOD INTERVENE SIMNET-AGENT RESUME-ALL-SUBORDINATES)**

Definition 17

```

>saf>objects>intervention.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

**2.2.2.2.18 (METHOD INTERVENE SIMNET-AGENT REJOIN-UNIT)**

Definition 18

```

>saf>objects>intervention.lisp
Type: Method
Arguments: ()
Outputs:
Calls: RESUME
       >saf>network>vars.lisp
       REJOIN-UNIT
       >saf>network>vars.lisp
       NET-MSG
       >saf>rudp>outgoing.lisp
Called by: None
Description: None

```

**2.2.2.2.19 (METHOD INTERVENE SIMNET-AGENT FORMATION)**

Definition 19

```

>saf>objects>intervention.lisp
Type: Method
Arguments: ()
Outputs:
Calls: LINE
       >saf>cm>line.lisp
       CHANGE-FORMATION
       >saf>network>vars.lisp
       NET-MSG
       >saf>rudp>outgoing.lisp
       FORMATION
       >saf>cm>control-measure.lisp
       FORMATION
       >saf>cm>control-measure.lisp
       LINE
       >saf>cm>line.lisp
       LINE
       >saf>cm>line.lisp
Called by: None
Description: None

```

### 2.2.2.3 CSU ui>subordinate-tasking.lisp

This unit contains the data-structures and code that implement the Subordinate Unit Tasking window. Included are functions to count frag orders, the overall program framework *subordinate-unit-tasking*, associated display methods, presentation types, the objects *unit-task* and *sub-task* and their associated methods, menu items, presentation translators and commands, display code, and an entry-point driving function, called *subordinate-task*, that calls the *subordinate-unit-tasking* program framework. Finally, a form is included that compiles the key flavor-methods.

#### 2.2.2.3.1 \*TOP-LEVEL-TASKING\*

Definition 1

```
>saf>ui>subordinate-tasking.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by:  LOAD-SCENARIO
           >saf>sys>new-storage.lisp
           STORE-SCENARIO
           >saf>sys>new-storage.lisp
           SAVE-FOR-TASKING-P
           >saf>sys>new-storage.lisp
           RESET-ALL-OVERLAYS-AND-TASKS
           >saf>ui>subordinate-tasking.lisp
           CLEAR-TOP-LEVEL-TASKING
           >saf>ui>subordinate-tasking.lisp
           SUBORDINATE-TASK
           >saf>ui>subordinate-tasking.lisp
Description:  None
```

#### 2.2.2.3.2 OVERLAY-IS-MODIFIED

Definition 2

```
>saf>ui>subordinate-tasking.lisp
Type: Function
Arguments: (OVERLAY)
Outputs:
Calls: None
Called by:  (METHOD DISPLAY-OVERLAY-TASKING UNIT-TASK)
           >saf>ui>subordinate-tasking.lisp
Description:  None
```

**2.2.2.3.3 (COMPILE LOAD EVAL)**

Definition 3

>saf>ui>subordinate-tasking.lisp  
 Type: EVAL-WHEN  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

**2.2.2.3.4 (METHOD SET-HIGHLIGHTED-PRESENTATION SUB-TASK-PANE AFTER)**

Definition 4

>saf>ui>subordinate-tasking.lisp  
 Type: Method  
 Arguments: (PRESENTATION &OPTIONAL DOCUMENTATION SHIFTS MORE-DOCUMENTATION)  
 Outputs:  
 Calls: SIMNET-AGENT  
       >saf>objects>simnet-agent.lisp  
       SIMNET-AGENT  
       >saf>objects>simnet-agent.lisp  
       SIMNET-AGENT  
       >saf>objects>simnet-agent.lisp  
       UNIT-TASK-UNIT  
       >saf>ui>subordinate-tasking.lisp  
       UNIT  
       >saf>cm>control-measure.lisp  
 Called by: None  
 Description: None

**2.2.2.3.5 \*FRAG-ORDER-COUNT\***

Definition 5

>saf>ui>subordinate-tasking.lisp  
 Type: Variable  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: PRINT-FRAGO-COUNT  
       >saf>ui>subordinate-tasking.lisp  
       RESET-FRAGO-COUNT  
       >saf>ui>subordinate-tasking.lisp  
       FRAGO-COUNT  
       >saf>ui>subordinate-tasking.lisp  
       COUNT-FRAGO  
       >saf>ui>subordinate-tasking.lisp  
 Description: None

**2.2.2.3.6 COUNT-FRAGO**

## Definition 6

>saf>ui>subordinate-tasking.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*FRAG-ORDER-COUNT\*  
>saf>ui>subordinate-tasking.lisp  
Called by: (METHOD COM-ISSUE-FRAG-ORDER-INTERNAL SUBORDINATE-UNIT-TASKING)  
No Source File Record  
Description: None

**2.2.2.3.7 FRAGO-COUNT**

## Definition 7

>saf>ui>subordinate-tasking.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*FRAG-ORDER-COUNT\*  
>saf>ui>subordinate-tasking.lisp  
Called by: None  
Description: None

**2.2.2.3.8 RESET-FRAGO-COUNT**

## Definition 8

>saf>ui>subordinate-tasking.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*FRAG-ORDER-COUNT\*  
>saf>ui>subordinate-tasking.lisp  
Called by: COMPLETE-C2-RESET  
>saf>network>top-level.lisp  
Description: None

**2.2.2.3.9 PRINT-FRAGO-COUNT**

## Definition 9

>saf>ui>subordinate-tasking.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*RADIO-OUTPUT\*  
>saf>sys>vars.lisp  
\*FRAG-ORDER-COUNT\*  
>saf>ui>subordinate-tasking.lisp

Called by: None  
Description: None

### 2.2.2.3.10 SUBORDINATE-UNIT-TASKING Definition 10

>saf>ui>subordinate-tasking.lisp  
Type: DEFINE-PROGRAM-FRAMEWORK  
Arguments: ()  
Outputs:  
Calls: None  
Called by: LEFT-ON-EXECUTE OVERLAY-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
>saf>patch>saf-6>saf-6-3.lisp  
SUBORDINATE-TASK  
>saf>ui>subordinate-tasking.lisp  
LEFT-ON-ISSUE FRAG ORDER-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
>saf>ui>subordinate-tasking.lisp  
LEFT-ON-WARN OVERLAY-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
>saf>ui>subordinate-tasking.lisp  
LEFT-ON-DONE-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
>saf>ui>subordinate-tasking.lisp  
LEFT-ON-CANCEL-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
>saf>ui>subordinate-tasking.lisp  
DEFINE-SUBORDINATE-UNIT-TASKING-COMMAND  
>saf>ui>subordinate-tasking.lisp  
Description: None

### 2.2.2.3.11 (METHOD CLEAR-STATE SUBORDINATE-UNIT-TASKING) Definition 11

>saf>ui>subordinate-tasking.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

### 2.2.2.3.12 (METHOD SAVE-SCROLL-STATE SUBORDINATE-UNIT-TASKING) Definition 12

>saf>ui>subordinate-tasking.lisp  
Type: Method  
Arguments: ()  
Outputs:

Calls: None  
 Called by: None  
 Description: None

### 2.2.2.3.13 (METHOD DISPLAY-TASKING-TABLE SUBORDINATE-UNIT-TASKING)

Definition 13

>saf>ui>subordinate-tasking.lisp  
 Type: Method  
 Arguments: (STREAM)  
 Outputs:  
 Calls: DISPLAY-SUBORDINATE-TASKING-TABLE  
       >saf>ui>subordinate-tasking.lisp  
 Called by: None  
 Description: None

### 2.2.2.3.14 (METHOD DISPLAY-TITLE SUBORDINATE-UNIT-TASKING)

Definition 14

>saf>ui>subordinate-tasking.lisp  
 Type: Method  
 Arguments: (STREAM)  
 Outputs:  
 Calls: \*DEFAULT-BATTALION-NUMBER\*  
       >saf>bmi>bmi-frame.lisp  
 Called by: None  
 Description: None

### 2.2.2.3.15 UNIT-TASK-OVERLAY

Definition 15

>saf>ui>subordinate-tasking.lisp  
 Type: DEFINE-PRESENTATION-TYPE  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (PRESENTATION-MOUSE-HANDLER SELECT-OVERLAY)  
           No Source File Record  
           (METHOD DISPLAY-OVERLAY-TASKING UNIT-TASK)  
           >saf>ui>subordinate-tasking.lisp  
           (METHOD COM-CHOOSE-OVERLAY-PARSER SUBORDINATE-UNIT-TASKING)  
           No Source File Record  
           (PROPERTY UNIT-TASK-OVERLAY DEFTYPE)  
           No Source File Record  
 Description: None

**2.2.2.3.16 UNIT-TASK-UNIT**

Definition 16

>saf>ui>subordinate-tasking.lisp  
 Type: DEFINE-PRESENTATION-TYPE  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (PRESENTATION-MOUSE-HANDLER SELECT-SUB-TASK)  
           No Source File Record  
           (METHOD DISPLAY-OVERLAY-TASKING UNIT-TASK)  
           >saf>ui>subordinate-tasking.lisp  
           (METHOD COM-CHANGE-SUB-TASK-PARSER SUBORDINATE-UNIT-TASKING)  
           No Source File Record  
           (PROPERTY UNIT-TASK-UNIT DEFTYPE)  
           No Source File Record  
           (METHOD SET-HIGHLIGHTED-PRESENTATION SUB-TASK-PANE AFTER)  
           >saf>ui>subordinate-tasking.lisp  
 Description: None

**2.2.2.3.17 COMBAT-INSTRUCTION-SET**

Definition 17

>saf>ui>subordinate-tasking.lisp  
 Type: DEFINE-PRESENTATION-TYPE  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (LOCATE-INSTANCE-VARIABLE (LOCF COMBAT-INSTRUCTION-SET)  
           CONTROL-MEASURE-BEHAVIOR  
           COMBAT-INSTRUCTION-SET)  
           No Source File Record  
           (WRITE-INSTANCE-VARIABLE (SETF COMBAT-INSTRUCTION-SET)  
           CONTROL-MEASURE-BEHAVIOR  
           COMBAT-INSTRUCTION-SET)  
           No Source File Record  
           (READ-INSTANCE-VARIABLE COMBAT-INSTRUCTION-SET CONTROL-MEASURE-BEHAVIOR COMBAT-INSTRUCTION-SET)  
           No Source File Record  
           (METHOD REVIEW-DATA ZONE)  
           >saf>cm>zone.lisp  
           (METHOD REVIEW-DATA AREA)  
           >saf>cm>area.lisp  
           (METHOD REVIEW-DATA LINE)  
           >saf>cm>line.lisp  
           (METHOD REVIEW-DATA CM-POINT)  
           >saf>cm>point.lisp  
           (METHOD COPY-BEHAVIOR LINE-BEHAVIOR)  
           >saf>cm>line.lisp  
           (METHOD SEND-BEH-INFO LINE-BEHAVIOR)  
           >saf>cm>line.lisp



```
(METHOD COPY-BEHAVIOR CM-POINT-BEHAVIOR)
>saf>cm>point.lisp
(METHOD SEND-BEH-INFO CM-POINT-BEHAVIOR)
>saf>cm>point.lisp
(METHOD SEND-CM-INFO GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK)
>saf>ui>subordinate-tasking.lisp
(PRESENTATION-FUNCTION COMBAT-INSTRUCTION-SET PARSER)
No Source File Record
(PROPERTY COMBAT-INSTRUCTION-SET DEFTYPE)
No Source File Record
CONTROL-MEASURE-BEHAVIOR
>saf>cm>control-measure.lisp
```

Description: None

### 2.2.2.3.18 UNIT-TASK

Definition 18

```
>saf>ui>subordinate-tasking.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: STORABLE-MIXIN
>saf>objects>storable-mixin.lisp
OVERLAY
>saf>cm>overlay.lisp
OVERLAY
>saf>cm>overlay.lisp
Called by: FILTERED-SAVE-INSTANCE
>saf>sys>new-storage.lisp
SAVE-FOR-TASKING-P
>saf>sys>new-storage.lisp
(METHOD COM-CHOOSE-OVERLAY-PARSER SUBORDINATE-UNIT-
TASKING)
No Source File Record
BUILD-UNIT-TASKING-STRUCTURE
>saf>ui>subordinate-tasking.lisp
Description: None
```

### 2.2.2.3.19 (METHOD MAKE-INSTANCE UNIT-TASK AFTER)

Definition 19

```
>saf>ui>subordinate-tasking.lisp
Type: Method
Arguments: (&REST IGNORE)
Outputs:
Calls: None
Called by: None
Description: None
```

**2.2.2.3.20 SUB-TASK**

## Definition 20

```

>saf>ui>subordinate-tasking.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: ROUTE
       >saf>cm>route.lisp
       STORABLE-MIXIN
       >saf>objects>storable-mixin.lisp
       UNIT
       >saf>cm>control-measure.lisp
       ROUTE
       >saf>cm>route.lisp
       ROUTE
       >saf>cm>route.lisp
Called by: FILTERED-SAVE-INSTANCE
          >saf>sys>new-storage.lisp
          (METHOD COM-CHANGE-SUB-TASK-PARSER SUBORDINATE-UNIT-
TASKING)
          No Source File Record
          MAKE-UNIT-LIST
          >saf>ui>subordinate-tasking.lisp
Description: None

```

**2.2.2.3.21 (METHOD MAKE-INSTANCE SUB-TASK AFTER)**

## Definition 21

```

>saf>ui>subordinate-tasking.lisp
Type: Method
Arguments: (&REST IGNORE)
Outputs:
Calls: ROUTE
       >saf>cm>route.lisp
       ROUTE
       >saf>cm>route.lisp
       ROUTE
       >saf>cm>route.lisp
Called by: None
Description: None

```

**2.2.2.3.22 (METHOD CIS-NAME SUB-TASK)**

## Definition 22

```

>saf>ui>subordinate-tasking.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

**2.2.2.3.23 (METHOD EXECUTE-SUB-TASK SUB-TASK)**

## Definition 23

>saf>ui>subordinate-tasking.lisp

Type: Method

Arguments: (OVERLAY)

Outputs:

Calls: ROUTE

>saf>cm>route.lisp

EXECUTE-OVERLAY

>saf>network>vars.lisp

NET-MSG

>saf>rudp>outgoing.lisp

UNIT

>saf>cm>control-measure.lisp

ROUTE

>saf>cm>route.lisp

ROUTE

>saf>cm>route.lisp

Called by: None

Description: None

**2.2.2.3.24 (METHOD REEXECUTE-SUB-TASK SUB-TASK)**

## Definition 24

>saf>ui>subordinate-tasking.lisp

Type: Method

Arguments: (OVERLAY)

Outputs:

Calls: ROUTE

>saf>cm>route.lisp

EXECUTE-OVERLAY

>saf>network>vars.lisp

NET-MSG

>saf>rudp>outgoing.lisp

UNIT

>saf>cm>control-measure.lisp

ROUTE

>saf>cm>route.lisp

ROUTE

>saf>cm>route.lisp

Called by: None

Description: None

**2.2.2.3.25 MAKE-UNIT-LIST**

## Definition 25

>saf>ui>subordinate-tasking.lisp

Type: Function

Arguments: (SUBS)

Outputs:

Calls: SUB-TASK  
>saf>ui>subordinate-tasking.lisp  
Called by: MERGE-UNIT-TASKING  
>saf>ui>subordinate-tasking.lisp  
BUILD-UNIT-TASKING-STRUCTURE  
>saf>ui>subordinate-tasking.lisp  
Description: None

#### 2.2.2.3.26 BUILD-UNIT-TASKING-STRUCTURE

Definition 26

>saf>ui>subordinate-tasking.lisp  
Type: Function  
Arguments: (SUBORDINATES OVERLAYS)  
Outputs:  
Calls: UNIT-TASK  
>saf>ui>subordinate-tasking.lisp  
MAKE-UNIT-LIST  
>saf>ui>subordinate-tasking.lisp  
Called by: MERGE-UNIT-TASKING  
>saf>ui>subordinate-tasking.lisp  
Description: None

#### 2.2.2.3.27 MERGE-UNIT-TASKING

Definition 27

>saf>ui>subordinate-tasking.lisp  
Type: Function  
Arguments: (TASKS SUBORDS)  
Outputs:  
Calls: \*ALL-OVERLAYS\*  
>saf>sys>vars.lisp  
MAKE-UNIT-LIST  
>saf>ui>subordinate-tasking.lisp  
BUILD-UNIT-TASKING-STRUCTURE  
>saf>ui>subordinate-tasking.lisp  
UNIT  
>saf>cm>control-measure.lisp  
OVERLAY  
>saf>cm>overlay.lisp  
OVERLAY  
>saf>cm>overlay.lisp  
Called by: SUBORDINATE-TASK  
>saf>ui>subordinate-tasking.lisp  
Description: None

### 2.2.2.3.28 (METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK) Definition 28

>saf>ui>subordinate-tasking.lisp  
 Type: Method  
 Arguments: ()  
 Outputs:  
 Calls: TYPE-OR-NO-CHANGE  
       >saf>sys>dw-presentation-types.lisp  
       ROUTE  
       >saf>cm>route.lisp  
       ROUTE  
       >saf>cm>route.lisp  
       COMBAT-INSTRUCTION-SET  
       >saf>ui>subordinate-tasking.lisp  
       UNIT  
       >saf>cm>control-measure.lisp  
       ROUTE  
       >saf>cm>route.lisp  
       ROUTE  
       >saf>cm>route.lisp  
       ROUTE  
       >saf>cm>route.lisp  
       ROUTE  
       >saf>cm>route.lisp  
       OVERLAY  
       >saf>cm>overlay.lisp  
       OVERLAY  
       >saf>cm>overlay.lisp  
 Called by: None  
 Description: None

### 2.2.2.3.29 (COM-CANCEL MENU-ACCELERATOR T) Definition 29

>saf>ui>subordinate-tasking.lisp  
 Type: DEFINE-SUBORDINATE-UNIT-TASKING-COMMAND  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

### 2.2.2.3.30 (COM-DONE MENU-ACCELERATOR T) Definition 30

>saf>ui>subordinate-tasking.lisp  
 Type: DEFINE-SUBORDINATE-UNIT-TASKING-COMMAND  
 Arguments: ()  
 Outputs:

Calls: None  
Called by: None  
Description: None

**2.2.2.3.31 (COM-WARN-OVERLAY MENU-ACCELERATOR T)**  
Definition 31

>saf>ui>subordinate-tasking.lisp  
Type: DEFINE-SUBORDINATE-UNIT-TASKING-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.3.32 (COM-EXECUTE-OVERLAY MENU-ACCELERATOR T)**  
Definition 32

>saf>ui>subordinate-tasking.lisp  
Type: DEFINE-SUBORDINATE-UNIT-TASKING-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.3.33 (COM-ISSUE-FRAG-ORDER MENU-ACCELERATOR T)**  
Definition 33

>saf>ui>subordinate-tasking.lisp  
Type: DEFINE-SUBORDINATE-UNIT-TASKING-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.3.34 (COM-CHOOSE-OVERLAY)**  
Definition 34

>saf>ui>subordinate-tasking.lisp  
Type: DEFINE-SUBORDINATE-UNIT-TASKING-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.3.35 SELECT-OVERLAY**

Definition 35

>saf>ui>subordinate-tasking.lisp  
Type: DEFINE-PRESENTATION-TO-COMMAND-TRANSLATOR  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.3.36 (COM-CHANGE-SUB-TASK)**

Definition 36

>saf>ui>subordinate-tasking.lisp  
Type: DEFINE-SUBORDINATE-UNIT-TASKING-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.3.37 SELECT-SUB-TASK**

Definition 37

>saf>ui>subordinate-tasking.lisp  
Type: DEFINE-PRESENTATION-TO-COMMAND-TRANSLATOR  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.2.3.38 (METHOD DISPLAY-SUB-TASKING SUB-TASK)**

Definition 38

>saf>ui>subordinate-tasking.lisp  
Type: Method  
Arguments: (STREAM)  
Outputs:  
Calls: ROUTE  
>saf>cm>route.lisp  
UNIT  
>saf>cm>control-measure.lisp  
ROUTE  
>saf>cm>route.lisp  
ROUTE  
>saf>cm>route.lisp  
Called by: None  
Description: None

**2.2.2.3.39 (METHOD DISPLAY-OVERLAY-TASKING UNIT-TASK)**

Definition 39

>saf>ui>subordinate-tasking.lisp  
 Type: Method  
 Arguments: (STREAM)  
 Outputs:  
 Calls: OVERLAY-IS-MODIFIED  
 >saf>ui>subordinate-tasking.lisp  
 UNIT-TASK-OVERLAY  
 >saf>ui>subordinate-tasking.lisp  
 UNIT-TASK-UNIT  
 >saf>ui>subordinate-tasking.lisp  
 OVERLAY  
 >saf>cm>overlay.lisp  
 OVERLAY  
 >saf>cm>overlay.lisp  
 Called by: None  
 Description: None

**2.2.2.3.40 DISPLAY-SUBORDINATE-TASKING-TABLE**

Definition 40

>saf>ui>subordinate-tasking.lisp  
 Type: Function  
 Arguments: (STREAM OVERLAY-TASKS)  
 Outputs:  
 Calls: None  
 Called by: (METHOD DISPLAY-TASKING-TABLE SUBORDINATE-UNIT-TASKING)  
 >saf>ui>subordinate-tasking.lisp  
 Description: None

**2.2.2.3.41 SUBORDINATE-TASK**

Definition 41

>saf>ui>subordinate-tasking.lisp  
 Type: Function  
 Arguments: (UNIT)  
 Outputs:  
 Calls: \*ALL-OVERLAYS\*  
 >saf>sys>vars.lisp  
 LOCAL  
 >saf>network>vars.lisp  
 GET-SUBORDINATES-INSTANCES  
 >saf>objects>simnet-agent.lisp  
 TOP-LEVEL-UNITS  
 >saf>simnet-objects>vehicle-tracking.lisp  
 \*TOP-LEVEL-TASKING\*  
 >saf>ui>subordinate-tasking.lisp



**SUBORDINATE-UNIT-TASKING**

&gt;saf&gt;ui&gt;subordinate-tasking.lisp

**MERGE-UNIT-TASKING**

&gt;saf&gt;ui&gt;subordinate-tasking.lisp

**SUBORDINATE-UNIT-TASKING**

&gt;saf&gt;ui&gt;subordinate-tasking.lisp

Called by: OPORD

&gt;saf&gt;ui&gt;opord.lisp

Description: This function is the entry point into the subordinate unit tasking sequence

**2.2.2.3.42 CLEAR-TOP-LEVEL-TASKING**

Definition 42

&gt;saf&gt;ui&gt;subordinate-tasking.lisp

Type: Function

Arguments: ()

Outputs:

Calls: \*TOP-LEVEL-TASKING\*

&gt;saf&gt;ui&gt;subordinate-tasking.lisp

Called by: COMPLETE-C2-RESET

&gt;saf&gt;network&gt;top-level.lisp

Description: None

**2.2.2.3.43 RESET-ALL-OVERLAYS-AND-TASKS**

Definition 43

&gt;saf&gt;ui&gt;subordinate-tasking.lisp

Type: Function

Arguments: ()

Outputs:

Calls: \*ALL-OVERLAYS\*

&gt;saf&gt;sys&gt;vars.lisp

ALL-LOCAL-VEHICLES

&gt;saf&gt;simnet-objects&gt;vehicle-tracking.lisp

\*TOP-LEVEL-TASKING\*

&gt;saf&gt;ui&gt;subordinate-tasking.lisp

Called by: COMPLETE-C2-RESET

&gt;saf&gt;network&gt;top-level.lisp

EXIT-CONN

&gt;saf&gt;network&gt;connection.lisp

Description: None

**2.2.2.3.44 SUBORDINATE-UNIT-TASKING**

Definition 44

&gt;saf&gt;ui&gt;subordinate-tasking.lisp

Type: COMPILE-FLAVOR-METHODS

Arguments: ()

Outputs:

Calls: None

Called by: LEFT-ON-EXECUTE OVERLAY-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
 >saf>patch>saf-6>saf-6-3.lisp  
 SUBORDINATE-TASK  
 >saf>ui>subordinate-tasking.lisp  
 LEFT-ON-ISSUE FRAG ORDER-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
 >saf>ui>subordinate-tasking.lisp  
 LEFT-ON-WARN OVERLAY-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
 >saf>ui>subordinate-tasking.lisp  
 LEFT-ON-DONE-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
 >saf>ui>subordinate-tasking.lisp  
 LEFT-ON-CANCEL-AT-TOP-LEVEL-SUBORDINATE-UNIT-TASKING-MENU-COMMAND  
 >saf>ui>subordinate-tasking.lisp  
 DEFINE-SUBORDINATE-UNIT-TASKING-COMMAND  
 >saf>ui>subordinate-tasking.lisp  
 Description: None

#### 2.2.2.4 CSU cm>overlay.lisp

This unit contains the definition of the overlay structure, as well as the routines to add, delete and display control measures. It also includes an overlay method called *send-overlay-to-simhost*, that sends control measure information to the Simhost.

##### 2.2.2.4.1 OVERLAY

Definition 1

```
>saf>cm>overlay.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: None
Called by: (WRITE-INSTANCE-VARIABLE (SETF OVERLAY) UNIT-TASK OVERLAY)
           No Source File Record
           (READ-INSTANCE-VARIABLE OVERLAY UNIT-TASK OVERLAY)
           No Source File Record
           (WRITE-INSTANCE-VARIABLE (SETF OVERLAY) SIMNET-AGENT OVERLAY)
           No Source File Record
           (READ-INSTANCE-VARIABLE OVERLAY SIMNET-AGENT OVERLAY)
           No Source File Record
           COPY-RELEVANT-IVS
           >saf>sys>new-storage.lisp
           CLEAR-OVERLAYS
           >saf>ui>mouse-interface.lisp
           (METHOD DISPLAY-OVERLAY-TASKING UNIT-TASK)
           >saf>ui>subordinate-tasking.lisp
```

(METHOD COM-EXECUTE-OVERLAY-INTERNAL SUBORDINATE-UNIT-TASKING)

No Source File Record

(METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK)

>saf>ui>subordinate-tasking.lisp

COPY-RELEVANT-IVS

>saf>sys>new-storage.lisp

(METHOD COM-ISSUE-FRAG-ORDER-INTERNAL SUBORDINATE-UNIT-TASKING)

No Source File Record

(METHOD COM-WARN-OVERLAY-INTERNAL SUBORDINATE-UNIT-TASKING)

No Source File Record

MERGE-UNIT-TASKING

>saf>ui>subordinate-tasking.lisp

COPY-RELEVANT-IVS

>saf>sys>new-storage.lisp

LOAD-SCENARIO

>saf>sys>new-storage.lisp

LOAD-OVERLAY

>saf>sys>new-storage.lisp

MAKE-OVERLAY

>saf>cm>overlay.lisp

OVERLAY?

>saf>cm>overlay.lisp

UNIT-TASK

>saf>ui>subordinate-tasking.lisp

SIMNET-AGENT

>saf>objects>simnet-agent.lisp

Description: None

#### 2.2.2.4.2 OVERLAY?

Definition 2

>saf>cm>overlay.lisp

Type: Function

Arguments: (OVERLAY)

Outputs:

Calls: OVERLAY

>saf>cm>overlay.lisp

OVERLAY

>saf>cm>overlay.lisp

Called by: None

Description: None

#### 2.2.2.4.3 (METHOD MAKE-INSTANCE OVERLAY AFTER)

Definition 3

>saf>cm>overlay.lisp

Type: Method

Arguments: (&REST INIT-ARGS)

Outputs:

Calls: \*CONTROL-MEASURE-ID\*  
>saf>cm>control-measure.lisp  
UNIQUE-CM-ID  
>saf>cm>control-measure.lisp  
Called by: None  
Description: None

#### 2.2.2.4.4 (METHOD KILL OVERLAY)

Definition 4

>saf>cm>overlay.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: \*PVD-DISPLAY\*  
>saf>sys>vars.lisp  
\*ALL-OVERLAYS\*  
>saf>sys>vars.lisp  
Called by: None  
Description: None

#### 2.2.2.4.5 (METHOD PRINT-SELF OVERLAY)

Definition 5

>saf>cm>overlay.lisp  
Type: Method  
Arguments: (STREAM PRINT-DEPTH SLASHIFY-P)  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.2.2.4.6 (METHOD REVIEW-DATA OVERLAY)

Definition 6

>saf>cm>overlay.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.2.2.4.7 (METHOD REFRESH OVERLAY)

Definition 7

>saf>cm>overlay.lisp  
Type: Method  
Arguments: (STREAM)  
Outputs:

Calls: None  
Called by: None  
Description: None

#### 2.2.2.4.8 (METHOD DRAW OVERLAY)

Definition 8

>saf>cm>overlay.lisp

Type: Method  
Arguments: (STREAM)  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.2.2.4.9 (METHOD ERASE OVERLAY)

Definition 9

>saf>cm>overlay.lisp

Type: Method  
Arguments: (STREAM)  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.2.2.4.10 (METHOD ADD-NEW-CONTROL-MEASURE OVERLAY)

Definition 10

>saf>cm>overlay.lisp

Type: Method  
Arguments: ()  
Outputs:  
Calls: \*PVD-DISPLAY\*  
>saf>sys>vars.lisp  
MENU-CHOOSE  
>saf>sys>utilities.lisp  
MAKE-ROUTE  
>saf>cm>route.lisp  
MAKE-POINT  
>saf>cm>point.lisp  
MAKE-LINE  
>saf>cm>line.lisp  
MAKE-AREA  
>saf>cm>area.lisp  
MAKE-ZONE  
>saf>cm>zone.lisp  
Called by: None  
Description: None

**2.2.2.4.11 (METHOD ADD-CONTROL-MEASURE OVERLAY)**

Definition 11

&gt;saf&gt;cm&gt;overlay.lisp

Type: Method

Arguments: (CM BEH)

Outputs:

Calls: \*PVD-DISPLAY\*

&gt;saf&gt;sys&gt;vars.lisp

Called by: None

Description: None

**2.2.2.4.12 (METHOD DELETE-CONTROL-MEASURE OVERLAY)**

Definition 12

&gt;saf&gt;cm&gt;overlay.lisp

Type: Method

Arguments: (CM)

Outputs:

Calls: \*PVD-DISPLAY\*

&gt;saf&gt;sys&gt;vars.lisp

Called by: None

Description: None

**2.2.2.4.13 (METHOD DELETE-ALL-CONTROL-MEASURES OVERLAY)**

Definition 13

&gt;saf&gt;cm&gt;overlay.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.2.4.14 \*CM-DELETE-MENU\***

Definition 14

&gt;saf&gt;cm&gt;overlay.lisp

Type: Variable

Arguments: ()

Outputs:

Calls: None

Called by: GET-DELETE-CM-MENU

&gt;saf&gt;cm&gt;overlay.lisp

Description: None

**2.2.2.4.15 \*CM-DELETE-MENU-COLOR\***

Definition 15

&gt;saf&gt;cm&gt;overlay.lisp

Type: Variable

Arguments: ()

Outputs:

Calls: None

Called by: GET-DELETE-CM-MENU

&gt;saf&gt;cm&gt;overlay.lisp

Description: None

**2.2.2.4.16 GET-DELETE-CM-MENU**

Definition 16

&gt;saf&gt;cm&gt;overlay.lisp

Type: Function

Arguments: ()

Outputs:

Calls: \*CM-DELETE-MENU\*

&gt;saf&gt;cm&gt;overlay.lisp

\*CM-DELETE-MENU-COLOR\*

&gt;saf&gt;cm&gt;overlay.lisp

Called by: MULTIPLE-MENU-DELETE-CMS

&gt;saf&gt;cm&gt;overlay.lisp

Description: None

**2.2.2.4.17 MULTIPLE-MENU-DELETE-CMS**

Definition 17

&gt;saf&gt;cm&gt;overlay.lisp

Type: Function

Arguments: (CHOICE-LIST)

Outputs:

Calls: GET-DELETE-CM-MENU

&gt;saf&gt;cm&gt;overlay.lisp

Called by: (METHOD DELETE-SOME-CONTROL-MEASURES OVERLAY)

&gt;saf&gt;cm&gt;overlay.lisp

Description: None

**2.2.2.4.18 (METHOD DELETE-SOME-CONTROL-MEASURES OVERLAY)**

Definition 18

&gt;saf&gt;cm&gt;overlay.lisp

Type: Method

Arguments: ()

Outputs:  
Calls: MULTIPLE-MENU-DELETE-CMS  
      >saf>cm>overlay.lisp  
Called by: None  
Description: None

#### 2.2.2.4.19 (METHOD SEND-OVERLAY-TO-SIMHOST OVERLAY)

Definition 19

      >saf>cm>overlay.lisp  
Type: Method  
Arguments: (UNIT-ID &KEY FRAGO FORCE)  
Outputs:  
Calls: POINT  
      >saf>interface>model-menu.lisp  
      DELETE-CM  
      >saf>network>vars.lisp  
      NET-MSG  
      >saf>rdp>outgoing.lisp  
      CM-POINT  
      >saf>cm>point.lisp  
      CM-POINT  
      >saf>cm>point.lisp  
      CM-POINT  
      >saf>cm>point.lisp  
      POINT  
      >saf>interface>model-menu.lisp  
Called by: None  
Description: None

#### 2.2.2.4.20 (METHOD CM-NEEDS-UPDATING OVERLAY)

Definition 20

      >saf>cm>overlay.lisp  
Type: Method  
Arguments: (CM)  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.2.2.4.21 (METHOD ALL-ROUTES OVERLAY)

Definition 21

      >saf>cm>overlay.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None



**2.2.2.4.22 (METHOD OVERLAY-OPS OVERLAY)**

## Definition 22

>saf>cm>overlay.lisp  
 Type: Method  
 Arguments: ()  
 Outputs:  
 Calls: \*PVD-DISPLAY\*  
       >saf>sys>vars.lisp  
       MENU-CHOOSE  
       >saf>sys>utilities.lisp  
 Called by: None  
 Description: None

**2.2.2.4.23 OVERLAY**

## Definition 23

>saf>cm>overlay.lisp  
 Type: COMPILE-FLAVOR-METHODS  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (WRITE-INSTANCE-VARIABLE (SETF OVERLAY) UNIT-TASK  
 OVERLAY)  
       No Source File Record  
       (READ-INSTANCE-VARIABLE OVERLAY UNIT-TASK OVERLAY)  
       No Source File Record  
       (WRITE-INSTANCE-VARIABLE (SETF OVERLAY) SIMNET-AGENT  
 OVERLAY)  
       No Source File Record  
       (READ-INSTANCE-VARIABLE OVERLAY SIMNET-AGENT OVERLAY)  
       No Source File Record  
       COPY-RELEVANT-IVS  
       >saf>sys>new-storage.lisp  
       CLEAR-OVERLAYS  
       >saf>ui>mouse-interface.lisp  
       (METHOD DISPLAY-OVERLAY-TASKING UNIT-TASK)  
       >saf>ui>subordinate-tasking.lisp  
       (METHOD COM-EXECUTE-OVERLAY-INTERNAL SUBORDINATE-UNIT-  
 TASKING)  
       No Source File Record  
       (METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK)  
       >saf>ui>subordinate-tasking.lisp  
       COPY-RELEVANT-IVS  
       >saf>sys>new-storage.lisp  
       (METHOD COM-ISSUE-FRAG-ORDER-INTERNAL SUBORDINATE-UNIT-  
 TASKING)  
       No Source File Record  
       (METHOD COM-WARN-OVERLAY-INTERNAL SUBORDINATE-UNIT-  
 TASKING)  
       No Source File Record  
       MERGE-UNIT-TASKING  
       >saf>ui>subordinate-tasking.lisp

```

COPY-RELEVANT-IVS
>saf>sys>new-storage.lisp
LOAD-SCENARIO
>saf>sys>new-storage.lisp
LOAD-OVERLAY
>saf>sys>new-storage.lisp
MAKE-OVERLAY
>saf>cm>overlay.lisp
OVERLAY?
>saf>cm>overlay.lisp
UNIT-TASK
>saf>ui>subordinate-tasking.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
Description:  None

```

#### 2.2.2.4.24 MAKE-OVERLAY

Definition 24

```

>saf>cm>overlay.lisp
Type: Function
Arguments: (&OPTIONAL NAME)
Outputs:
Calls: NAME
>saf>sysdcl.lisp
*PVD-DISPLAY*
>saf>sys>vars.lisp
*ALL-OVERLAYS*
>saf>sys>vars.lisp
OVERLAY
>saf>cm>overlay.lisp
OVERLAY
>saf>cm>overlay.lisp
Called by:  CHOOSE-AN-OVERLAY
>saf>cm>overlay.lisp
CHOOSE-AN-OVERLAY
>saf>cm>overlay.lisp
Description:  None

```

#### 2.2.2.4.25 REDRAW-OVERLAYS

Definition 25

```

>saf>cm>overlay.lisp
Type: Function
Arguments: ()
Outputs:
Calls: *PVD-DISPLAY*
>saf>sys>vars.lisp
*ALL-OVERLAYS*
>saf>sys>vars.lisp

```

Called by: DRAW-MAP  
>saf>sys>update-process.lisp  
Description: None

#### 2.2.2.4.26 CHOOSE-AN-OVERLAY

Definition 26

>saf>cm>overlay.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*ALL-OVERLAYS\*  
>saf>sys>vars.lisp  
MENU-CHOOSE  
>saf>sys>utilities.lisp  
MAKE-OVERLAY  
>saf>cm>overlay.lisp  
MAKE-OVERLAY  
>saf>cm>overlay.lisp  
Called by: (METHOD COM-SELECT-SUBPARAGRAPH-INTERNAL SAF)  
No Source File Record  
Description: None

#### 2.2.2.4.27 SORT-CMS

Definition 27

>saf>cm>overlay.lisp  
Type: Function  
Arguments: (POINT CM-LIST)  
Outputs:  
Calls: DISTANCE  
>map>utilities.lisp  
POINT  
>saf>interface>model-menu.lisp  
POINT  
>saf>interface>model-menu.lisp  
Called by: None  
Description: None

### 2.2.3 Control Measures CSC

This CSC contains the code to create and manipulate graphical control measures on the color map display. These control measures include points, lines, areas, and zones.

The CSUs in this CSC are:

```
cm>control-measure.lisp csu
cm>control-measure-point.lisp csu
cm>point.lisp csu
cm>line.lisp csu
cm>generic-area.lisp csu
cm>area.lisp csu
cm>zone.lisp csu
```

### 2.2.3.1 CSU `cm>control-measure.lisp`

This unit contains the definition of the basic control measure structure, as well as the routines to manipulate them. Presentation types that make control measures mouse-sensitive are also included.

After control measures are created as geometrical objects on the Symbolics, they are sent to the Simhost. It is the Simhost which detects when a unit has "triggered" a control measure, and carries out the associated control-measure-behavior. This explains why there is no code on the Symbolics for detecting when a unit has triggered a control measure.

Each of the control measure types, described below -- route, cm-point, line, zone, and area -- have a similar set of related structures. These include an associated behavior object (e.g., the behavior object for *line* is called *line-behavior*, etc.), which will be a subclass of the *control-measure-behavior class*. (see the object hierarchy table in CSU objects>defobject.lisp, section 2.5.2) Each control measure also has a *make-behavior* method that creates a behavior instance for it, a *review-data* method that allows the user to modify the control-measure from a menu, and a *cm-intersection* method that can be used to determine if the control measure intersects a given line. All control measures except the point control measure have associated paths composed of multiple line segments. These can be edited using methods like *move-point*, *delete-point*, *insert-point-after*, and *insert-point-before*. The *orthogonalize* method takes a list of points and puts all the x coordinates in one list and all the y coordinates in another, for use in a message to the Simhost.

#### 2.2.3.1.1 \*CONTROL-MEASURE-ID\*

Definition 1

```
>saf>cm>control-measure.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: (METHOD MAKE-INSTANCE OVERLAY AFTER)
>saf>cm>overlay.lisp
(METHOD MAKE-INSTANCE CONTROL-MEASURE-POINT AFTER)
>saf>cm>control-measure-point.lisp
(METHOD MAKE-INSTANCE CONTROL-MEASURE AFTER)
>saf>cm>control-measure.lisp
UNIQUE-CM-ID
>saf>cm>control-measure.lisp
Description: None
```

**2.2.3.1.2 UNIQUE-CM-ID**

## Definition 2

>saf>cm>control-measure.lisp  
 Type: Subst  
 Arguments: ()  
 Outputs:  
 Calls: \*CONTROL-MEASURE-ID\*  
 >saf>cm>control-measure.lisp  
 Called by: (METHOD MAKE-INSTANCE OVERLAY AFTER)  
 >saf>cm>overlay.lisp  
 (METHOD MAKE-INSTANCE CONTROL-MEASURE-POINT AFTER)  
 >saf>cm>control-measure-point.lisp  
 (METHOD MAKE-INSTANCE CONTROL-MEASURE AFTER)  
 >saf>cm>control-measure.lisp  
 Description: None

**2.2.3.1.3 'CONTROL-MEASURE**

## Definition 3

>saf>cm>control-measure.lisp  
 Type: SHADOW  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

**2.2.3.1.4 CONTROL-MEASURE**

## Definition 4

>saf>cm>control-measure.lisp  
 Type: DEFOBJECT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (WRITE-INSTANCE-VARIABLE (SETF CONTROL-MEASURE)  
 CONTROL-MEASURE-POINT CONTROL-MEASURE)  
 No Source File Record  
 (READ-INSTANCE-VARIABLE CONTROL-MEASURE CONTROL-  
 MEASURE-POINT CONTROL-MEASURE)  
 No Source File Record  
 (LOCATE-INSTANCE-VARIABLE (LOCF CONTROL-MEASURE) CONTROL-  
 MEASURE-BEHAVIOR CONTROL-MEASURE)  
 No Source File Record  
 (WRITE-INSTANCE-VARIABLE (SETF CONTROL-MEASURE) CONTROL-  
 MEASURE-BEHAVIOR CONTROL-MEASURE)  
 No Source File Record  
 (READ-INSTANCE-VARIABLE CONTROL-MEASURE CONTROL-  
 MEASURE-BEHAVIOR CONTROL-MEASURE)  
 No Source File Record  
 GENERIC-AREA

```

>saf>cm>generic-area.lisp
ZONE
>saf>cm>zone.lisp
AREA
>saf>cm>area.lisp
LINE
>saf>cm>line.lisp
CM-POINT
>saf>cm>point.lisp
ROUTE
>saf>cm>route.lisp
(METHOD COPY ZONE)
>saf>cm>zone.lisp
(METHOD COPY AREA)
>saf>cm>area.lisp
(METHOD INITIALIZE-POINTS GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD COPY LINE)
>saf>cm>line.lisp
(METHOD INITIALIZE-POINTS LINE)
>saf>cm>line.lisp
(METHOD COPY ROUTE)
>saf>cm>route.lisp
(METHOD INSERT-POINT-BEFORE ROUTE)
>saf>cm>route.lisp
(METHOD INSERT-POINT-AFTER ROUTE)
>saf>cm>route.lisp
(METHOD INITIALIZE-POINTS ROUTE)
>saf>cm>route.lisp
(METHOD PRINT-SELF CONTROL-MEASURE-BEHAVIOR)
>saf>cm>control-measure.lisp
(PRESENTATION-MOUSE-HANDLER CONTROL-MEASURE-GESTURE)
No Source File Record
CONTROL-MEASURE-POINT
>saf>cm>control-measure-point.lisp
CONTROL-MEASURE-BEHAVIOR
>saf>cm>control-measure.lisp

```

Description: None

### 2.2.3.1.5 (METHOD MAKE-INSTANCE CONTROL-MEASURE AFTER)

Definition 5

```

>saf>cm>control-measure.lisp
Type: Method
Arguments: (&REST INIT-ARGS)
Outputs:
Calls: *CONTROL-MEASURE-ID*
>saf>cm>control-measure.lisp
UNIQUE-CM-ID
>saf>cm>control-measure.lisp
Called by: None
Description: None

```

**2.2.3.1.6 (METHOD PRINT-SELF CONTROL-MEASURE)**

Definition 6

&gt;saf&gt;cm&gt;control-measure.lisp

Type: Method

Arguments: (STREAM PRINT-DEPTH SLASHIFY-P)

Outputs:

Calls: None

Called by: None

Description: None

**2.2.3.1.7 (METHOD REFRESH CONTROL-MEASURE)**

Definition 7

&gt;saf&gt;cm&gt;control-measure.lisp

Type: Method

Arguments: (STREAM)

Outputs:

Calls: None

Called by: None

Description: None

**2.2.3.1.8 (METHOD ROUTEP CONTROL-MEASURE)**

Definition 8

&gt;saf&gt;cm&gt;control-measure.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.3.1.9 (DRAW CONTROL-MEASURE)**

Definition 9

&gt;saf&gt;cm&gt;control-measure.lisp

Type: DEFWHOPPER

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.3.1.10 (ERASE CONTROL-MEASURE)**

Definition 10

&gt;saf&gt;cm&gt;control-measure.lisp

Type: DEFWHOPPER

Arguments: ()

Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.2.3.1.11 (METHOD DRAW-NAME CONTROL-MEASURE)

Definition 11

>saf>cm>control-measure.lisp  
Type: Method  
Arguments: (STREAM ALU)  
Outputs:  
Calls: WITH-INTEGGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
CONTROL-MEASURE-LABEL  
>saf>cm>control-measure.lisp  
Called by: None  
Description: None

#### 2.2.3.1.12 (METHOD ERASE-NAME CONTROL-MEASURE)

Definition 12

>saf>cm>control-measure.lisp  
Type: Method  
Arguments: (STREAM ALU)  
Outputs:  
Calls: WITH-INTEGGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
DELETE-DISPLAYED-PRESENTATION  
>saf>sys>utilities.lisp  
Called by: None  
Description: None

#### 2.2.3.1.13 (REVIEW-DATA CONTROL-MEASURE)

Definition 13

>saf>cm>control-measure.lisp  
Type: DEFWHOPPER  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None



**2.2.3.1.14 (MOVE-POINT CONTROL-MEASURE)**

Definition 14

```
>saf>cm>control-measure.lisp
Type: DEFWHOPPER
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.2.3.1.15 (DELETE-POINT CONTROL-MEASURE)**

Definition 15

```
>saf>cm>control-measure.lisp
Type: DEFWHOPPER
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.2.3.1.16 (INSERT-POINT-AFTER CONTROL-MEASURE)**

Definition 16

```
>saf>cm>control-measure.lisp
Type: DEFWHOPPER
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.2.3.1.17 (INSERT-POINT-BEFORE CONTROL-MEASURE)**

Definition 17

```
>saf>cm>control-measure.lisp
Type: DEFWHOPPER
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.2.3.1.18 (METHOD ADD-CM-TO-OVERLAY CONTROL-MEASURE)**

Definition 18

```
>saf>cm>control-measure.lisp
Type: Method
Arguments: (OVERLAY &OPTIONAL OLD-BEHAVIOR)
Outputs:
```

Calls: None  
Called by: None  
Description: None

### 2.2.3.1.19 CONTROL-MEASURE-BEHAVIOR

Definition 19

>saf>cm>control-measure.lisp  
Type: DEFOBJECT  
Arguments: ()  
Outputs:  
Calls: CHANGE-SPEED  
>saf>network>vars.lisp  
STORABLE-MIXIN  
>saf>objects>storable-mixin.lisp  
COMBAT-INSTRUCTION-SET  
>saf>ui>subordinate-tasking.lisp  
CONTROL-MEASURE  
>saf>cm>control-measure.lisp  
CONTROL-MEASURE  
>saf>cm>control-measure.lisp  
Called by: ZONE-BEHAVIOR  
>saf>cm>zone.lisp  
AREA-BEHAVIOR  
>saf>cm>area.lisp  
LINE-BEHAVIOR  
>saf>cm>line.lisp  
CM-POINT-BEHAVIOR  
>saf>cm>point.lisp  
ROUTE-BEHAVIOR  
>saf>cm>route.lisp  
FILTERED-SAVE-INSTANCE  
>saf>sys>new-storage.lisp  
Description: None

### 2.2.3.1.20 (METHOD PRINT-SELF CONTROL-MEASURE-BEHAVIOR)

Definition 20

>saf>cm>control-measure.lisp  
Type: Method  
Arguments: (STREAM PRINT-DEPTH SLASHIFY-P)  
Outputs:  
Calls: CONTROL-MEASURE  
>saf>cm>control-measure.lisp  
CONTROL-MEASURE  
>saf>cm>control-measure.lisp  
Called by: None  
Description: None

**2.2.3.1.21 CONTROL-MEASURE**

## Definition 21

```

>saf>cm>control-measure.lisp
Type: COMPILE-FLAVOR-METHODS
Arguments: ()
Outputs:
Calls: None
Called by: (WRITE-INSTANCE-VARIABLE (SETF CONTROL-MEASURE)
CONTROL-MEASURE-POINT CONTROL-MEASURE)
          No Source File Record
          (READ-INSTANCE-VARIABLE CONTROL-MEASURE CONTROL-
MEASURE-POINT CONTROL-MEASURE)
          No Source File Record
          (LOCATE-INSTANCE-VARIABLE (LOCF CONTROL-MEASURE) CONTROL-
MEASURE-BEHAVIOR CONTROL-MEASURE)
          No Source File Record
          (WRITE-INSTANCE-VARIABLE (SETF CONTROL-MEASURE) CONTROL-
MEASURE-BEHAVIOR CONTROL-MEASURE)
          No Source File Record
          (READ-INSTANCE-VARIABLE CONTROL-MEASURE CONTROL-
MEASURE-BEHAVIOR CONTROL-MEASURE)
          No Source File Record
          GENERIC-AREA
          >saf>cm>generic-area.lisp
          ZONE
          >saf>cm>zone.lisp
          AREA
          >saf>cm>area.lisp
          LINE
          >saf>cm>line.lisp
          CM-POINT
          >saf>cm>point.lisp
          ROUTE
          >saf>cm>route.lisp
          (METHOD COPY ZONE)
          >saf>cm>zone.lisp
          (METHOD COPY AREA)
          >saf>cm>area.lisp
          (METHOD INITIALIZE-POINTS GENERIC-AREA)
          >saf>cm>generic-area.lisp
          (METHOD COPY LINE)
          >saf>cm>line.lisp
          (METHOD INITIALIZE-POINTS LINE)
          >saf>cm>line.lisp
          (METHOD COPY ROUTE)
          >saf>cm>route.lisp
          (METHOD INSERT-POINT-BEFORE ROUTE)
          >saf>cm>route.lisp
          (METHOD INSERT-POINT-AFTER ROUTE)
          >saf>cm>route.lisp

```

```
(METHOD INITIALIZE-POINTS ROUTE)
>saf>cm>route.lisp
(METHOD PRINT-SELF CONTROL-MEASURE-BEHAVIOR)
>saf>cm>control-measure.lisp
(PRESENTATION-MOUSE-HANDLER CONTROL-MEASURE-GESTURE)
No Source File Record
CONTROL-MEASURE-POINT
>saf>cm>control-measure-point.lisp
CONTROL-MEASURE-BEHAVIOR
>saf>cm>control-measure.lisp
```

Description: None

### 2.2.3.1.22 REVERSE-XY

Definition 22

```
>saf>cm>control-measure.lisp
Type: Function
Arguments: (LIST)
Outputs:
Calls: None
Called by: MAKE-ZONE
>saf>cm>zone.lisp
MAKE-AREA
>saf>cm>area.lisp
MAKE-LINE
>saf>cm>line.lisp
MAKE-ROUTE
>saf>cm>route.lisp
FIND-RIVER-BEND-POINTS
>saf>cm>water-avoidance.lisp
CROSSING-LOCATION
>saf>cm>water-avoidance.lisp
ALIGN-POINTS
>saf>cm>water-avoidance.lisp
FIND-RIVER-POINTS
>saf>cm>water-avoidance.lisp
FIND-SEGMENT-CROSS-POINTS
>saf>cm>water-avoidance.lisp
INTERSECTION-DIRECTION
>saf>cm>water-avoidance.lisp
EXTEND-BRIDGE
>saf>cm>water-avoidance.lisp
GET-BRIDGE-POINTS
>saf>cm>road-routes.lisp
DRAW-EXPANDED-ROUTE-CORE
>saf>cm>road-routes.lisp
EXPAND-ROUTE-INTO-POINTS
>saf>cm>route-finder.lisp
```

Description: None

**2.2.3.1.23 UNIT**

## Definition 23

```

>saf>cm>control-measure.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: (WRITE-INSTANCE-VARIABLE (SETF UNIT) SUB-TASK UNIT)
           No Source File Record
           (READ-INSTANCE-VARIABLE UNIT SUB-TASK UNIT)
           No Source File Record
           (WRITE-INSTANCE-VARIABLE (SETF UNIT) OVERLAY UNIT)
           No Source File Record
           (READ-INSTANCE-VARIABLE UNIT OVERLAY UNIT)
           No Source File Record
           COPY-RELEVANT-IVS
>saf>sys>new-storage.lisp
(METHOD REEXECUTE-SUB-TASK SUB-TASK)
>saf>ui>subordinate-tasking.lisp
(METHOD DISPLAY-SUB-TASKING SUB-TASK)
>saf>ui>subordinate-tasking.lisp
(METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK)
>saf>ui>subordinate-tasking.lisp
(METHOD EXECUTE-SUB-TASK SUB-TASK)
>saf>ui>subordinate-tasking.lisp
SAVE-FOR-TASKING-P
>saf>sys>new-storage.lisp
MERGE-UNIT-TASKING
>saf>ui>subordinate-tasking.lisp
(METHOD SET-HIGHLIGHTED-PRESENTATION SUB-TASK-PANE AFTER)
>saf>ui>subordinate-tasking.lisp
CLEAR-OVERLAYS
>saf>ui>mouse-interface.lisp
(COMMAND-PARSER-FUNCTION COM-UNIT-OPS)
No Source File Record
(PRESENTATION-FUNCTION CM-FORMATION DATA-TYPE-
EQUIVALENT-STACK)
No Source File Record
(PRESENTATION-FUNCTION CM-FORMATION DATA-TYPE-
EQUIVALENT)
No Source File Record
(PRESENTATION-FUNCTION FORMATION DATA-TYPE-EQUIVALENT-
STACK)
No Source File Record
(PRESENTATION-FUNCTION FORMATION DATA-TYPE-EQUIVALENT)
No Source File Record
(PRESENTATION-FUNCTION LOCAL-UNIT DATA-TYPE-EQUIVALENT-
STACK)
No Source File Record
(PRESENTATION-FUNCTION LOCAL-UNIT DATA-TYPE-EQUIVALENT)
No Source File Record

```

(PRESENTATION-FUNCTION UNIT PRINTER)  
No Source File Record  
(PRESENTATION-FUNCTION UNIT PARSER)  
No Source File Record  
(PROPERTY UNIT DEFTYPE)  
No Source File Record  
SUB-TASK  
>saf>ui>subordinate-tasking.lisp  
OVERLAY  
>saf>cm>overlay.lisp

Description: None

#### 2.2.3.1.24 LOCAL-UNIT

Definition 24

>saf>cm>control-measure.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (PROPERTY LOCAL-UNIT DEFTYPE)  
No Source File Record  
Description: None

#### 2.2.3.1.25 \*PREV-UNITS\*

Definition 25

>saf>cm>control-measure.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (METHOD REVIEW-DATA ZONE)  
>saf>cm>zone.lisp  
(METHOD REVIEW-DATA AREA)  
>saf>cm>area.lisp  
(METHOD REVIEW-DATA LINE)  
>saf>cm>line.lisp  
(METHOD REVIEW-DATA CM-POINT)  
>saf>cm>point.lisp  
Description: None

#### 2.2.3.1.26 \*APPLIES-TO-UNIT-MENU\*

Definition 26

>saf>cm>control-measure.lisp  
Type: Parameter  
Arguments: ()  
Outputs:  
Calls: None

Called by: **MULTIPLE-MENU-CHOOSE-UNITS**

```
>saf>cm>control-measure.lisp
MAKE-APPLIES-TO-UNIT-MENU
>saf>cm>control-measure.lisp
MAKE-APPLIES-TO-UNIT-MENU
>saf>cm>control-measure.lisp
```

Description: None

### 2.2.3.1.27 **MAKE-APPLIES-TO-UNIT-MENU**

Definition 27

```
>saf>cm>control-measure.lisp
```

Type: Function

Arguments: ()

Outputs:

Calls: **\*APPLIES-TO-UNIT-MENU\***

```
>saf>cm>control-measure.lisp
```

**\*APPLIES-TO-UNIT-MENU\***

```
>saf>cm>control-measure.lisp
```

Called by: **MULTIPLE-MENU-CHOOSE-UNITS**

```
>saf>cm>control-measure.lisp
```

Description: None

### 2.2.3.1.28 **MULTIPLE-MENU-CHOOSE-UNITS**

Definition 28

```
>saf>cm>control-measure.lisp
```

Type: Function

Arguments: (CHOICE-LIST CHOSEN-LIST &OPTIONAL  
(LABEL '(STRING Assign Control Measure to Units STYLE (SAF MENU  
NORMAL))))

Outputs:

Calls: **\*APPLIES-TO-UNIT-MENU\***

```
>saf>cm>control-measure.lisp
```

**MAKE-APPLIES-TO-UNIT-MENU**

```
>saf>cm>control-measure.lisp
```

Called by: **CHOOSE-UNITS-FOR-CM**

```
>saf>cm>control-measure.lisp
```

Description: pops up menu and collects unit choices from it

### 2.2.3.1.29 **CHOOSE-UNITS-FOR-CM**

Definition 29

```
>saf>cm>control-measure.lisp
```

Type: Function

Arguments: (PREV-UNITS-LOCATIVE)

Outputs:

Calls: \*OPFOR-IO\*  
       >saf>sys>vars.lisp  
       SAY  
       >saf>sys>macros.lisp  
       COMPOSITE-OBJECT  
       >saf>objects>composite-object.lisp  
       COMPOSITE-OBJECT  
       >saf>objects>composite-object.lisp  
       ALL-LOCAL-VEHICLES  
       >saf>simnet-objects>vehicle-tracking.lisp  
       MULTIPLE-MENU-CHOOSE-UNITS  
       >saf>cm>control-measure.lisp  
 Called by: (PRESENTATION-FUNCTION CM-UNIT PARSER)  
           No Source File Record  
 Description: None

### 2.2.3.1.30 CM-UNIT

Definition 30

      >saf>cm>control-measure.lisp  
 Type: DEFINE-PRESENTATION-TYPE  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (METHOD REVIEW-DATA ZONE)  
           >saf>cm>zone.lisp  
           (METHOD REVIEW-DATA AREA)  
           >saf>cm>area.lisp  
           (METHOD REVIEW-DATA LINE)  
           >saf>cm>line.lisp  
           (METHOD REVIEW-DATA CM-POINT)  
           >saf>cm>point.lisp  
           (PRESENTATION-FUNCTION CM-UNIT PRINTER)  
           No Source File Record  
           (PRESENTATION-FUNCTION CM-UNIT PARSER)  
           No Source File Record  
           (PROPERTY CM-UNIT DEFTYPE)  
           No Source File Record  
 Description: None

### 2.2.3.1.31 REMOVE-UNIT-POINTERS-IN-BEHAVIORS

Definition 31

      >saf>cm>control-measure.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: \*ALL-OVERLAYS\*  
       >saf>sys>vars.lisp  
 Called by: COMPLETE-C2-RESET  
           >saf>network>top-level.lisp  
 Description: None



**2.2.3.1.32 FORMATION**

Definition 32

```

>saf>cm>control-measure.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: (WRITE-INSTANCE-VARIABLE SET-FORMATION FORMATION-
OBJECT FORMATION)
           No Source File Record
           (WRITE-INSTANCE-VARIABLE (SETF .FORMATION) FORMATION-
OBJECT FORMATION)
           No Source File Record
           (READ-INSTANCE-VARIABLE FORMATION FORMATION-OBJECT
FORMATION)
           No Source File Record
           (READ-INSTANCE-VARIABLE .FORMATION FORMATION-OBJECT
FORMATION)
           No Source File Record
           (WRITE-INSTANCE-VARIABLE (SETF FORMATION) SIMNET-AGENT
FORMATION)
           No Source File Record
           (READ-INSTANCE-VARIABLE FORMATION SIMNET-AGENT
FORMATION)
           No Source File Record
           MAKE-AGENT
>saf>simnet-objects>vehicle-tracking.lisp
(METHOD INTERVENE SIMNET-AGENT FORMATION)
>saf>objects>intervention.lisp
FIND-FORMATION-INFO
>saf>sandbox>sandbox.lisp
(PRESENTATION-FUNCTION CM-FORMATION DATA-TYPE-
EQUIVALENT-STACK)
           No Source File Record
           (PRESENTATION-FUNCTION CM-FORMATION DATA-TYPE-
EQUIVALENT)
           No Source File Record
           (PRESENTATION-FUNCTION FORMATION DATA-TYPE-EQUIVALENT-
STACK)
           No Source File Record
           (PRESENTATION-FUNCTION FORMATION DATA-TYPE-EQUIVALENT)
           No Source File Record
           (PROPERTY FORMATION DEFTYPE)
           No Source File Record
           (METHOD INTERVENE SIMNET-AGENT FORMATION)
           >saf>objects>intervention.lisp
           FORMATION-OBJECT
           >saf>interface>formations.lisp
           SIMNET-AGENT
           >saf>objects>simnet-agent.lisp
Description: None

```

**2.2.3.1.33 CM-FORMATION**

Definition 33

>saf>cm>control-measure.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (PRESENTATION-FUNCTION CM-FORMATION DATA-TYPE-EQUIVALENT-STACK)  
No Source File Record  
(PRESENTATION-FUNCTION CM-FORMATION DATA-TYPE-EQUIVALENT)  
No Source File Record  
(PROPERTY CM-FORMATION DEFTYPE)  
No Source File Record  
Description: None

**2.2.3.1.34 CIS-FOR-CM**

Definition 34

>saf>cm>control-measure.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (PRESENTATION-FUNCTION CM-CIS DATA-TYPE-EQUIVALENT-STACK)  
No Source File Record  
(PRESENTATION-FUNCTION CM-CIS DATA-TYPE-EQUIVALENT)  
No Source File Record  
(PRESENTATION-FUNCTION CIS-FOR-CM PARSER)  
No Source File Record  
(PROPERTY CIS-FOR-CM DEFTYPE)  
No Source File Record  
Description: None

**2.2.3.1.35 CM-CIS**

Definition 35

>saf>cm>control-measure.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (METHOD REVIEW-DATA ZONE)  
>saf>cm>zone.lisp  
(METHOD REVIEW-DATA AREA)  
>saf>cm>area.lisp  
(METHOD REVIEW-DATA LINE)  
>saf>cm>line.lisp  
(METHOD REVIEW-DATA CM-POINT)

```
>saf>cm>point.lisp
(PRESENTATION-FUNCTION CM-CIS DATA-TYPE-EQUIVALENT-STACK)
No Source File Record
(PRESENTATION-FUNCTION CM-CIS DATA-TYPE-EQUIVALENT)
No Source File Record
(PROPERTY CM-CIS DEFTYPE)
No Source File Record
```

Description: None

### 2.2.3.1.36 CM-SPEED

Definition 36

```
>saf>cm>control-measure.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: (PROPERTY CM-SPEED DEFTYPE)
No Source File Record
(METHOD REVIEW-DATA LINE)
>saf>cm>line.lisp
(METHOD REVIEW-DATA CM-POINT)
>saf>cm>point.lisp
```

Description: None

### 2.2.3.1.37 WORLD-COORDS

Definition 37

```
>saf>cm>control-measure.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: (PROPERTY WORLD-COORDS DEFTYPE)
No Source File Record
(METHOD INTERVENE SIMNET-AGENT ATTACK)
>saf>objects>intervention.lisp
(METHOD INTERVENE SIMNET-AGENT LAND)
>saf>objects>intervention.lisp
(METHOD INTERVENE SIMNET-AGENT GO-TO-LOCATION)
>saf>objects>intervention.lisp
(METHOD SPECIFY-RULES-OF-ENGAGEMENT GUNNER)
>saf>objects>gunner.lisp
```

Description: None

### 2.2.3.1.38 CONTROL-MEASURE-LABEL

Definition 38

```
>saf>cm>control-measure.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
```

Outputs:

Calls: None

Called by: (PRESENTATION-MOUSE-HANDLER CONTROL-MEASURE-LABEL-GESTURE)

No Source File Record  
(METHOD DRAW CM-POINT)

>saf>cm>point.lisp

(PROPERTY CONTROL-MEASURE-LABEL DEFTYPE)

No Source File Record

(METHOD DRAW-NAME CONTROL-MEASURE)

>saf>cm>control-measure.lisp

Description: None

### 2.2.3.1.39 CONTROL-MEASURE-LABEL-GESTURE

Definition 39

>saf>cm>control-measure.lisp

Type: DEFINE-PRESENTATION-ACTION

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

### 2.2.3.2 CSU cm>control-measure-point.lisp

This unit contains the definition of the control measure vertex points, which are the mouse-sensitive points that make up the control measures. It also contains the routines to manipulate and display these points. Their object class is called *control-measure-point*. Methods for this class allow creation, display, display as the first point, and erasure. The presentation for control-measure-points defines a menu of operations for moving, deleting, and inserting points. A utility is also included that takes a list of numbers, breaks them up into (x, y) coordinate pairs, and makes control-measure-points at those locations.

#### 2.2.3.2.1 CONTROL-MEASURE-POINT

Definition 1

>saf>cm>control-measure-point.lisp

Type: DEFOBJECT

Arguments: ()

Outputs:

Calls: None

Called by: ROUTE-POINT

>saf>cm>route-point.lisp

(PRESENTATION-MOUSE-HANDLER CONTROL-MEASURE-GESTURE)

No Source File Record

XY-LIST-TO-POINTS

>saf>cm>control-measure-point.lisp

(METHOD COPY CONTROL-MEASURE-POINT)  
 >saf>cm>control-measure-point.lisp  
 (METHOD (SETF BOX-SIZE) CONTROL-MEASURE-POINT)  
 No Source File Record  
 (METHOD BOX-SIZE CONTROL-MEASURE-POINT)  
 No Source File Record

Description: None

#### 2.2.3.2.2 (METHOD MAKE-INSTANCE CONTROL-MEASURE-POINT AFTER)

Definition 2

>saf>cm>control-measure-point.lisp

Type: Method

Arguments: (&REST INIT-ARGS)

Outputs:

Calls: \*CONTROL-MEASURE-ID\*

>saf>cm>control-measure.lisp

UNIQUE-CM-ID

>saf>cm>control-measure.lisp

Called by: None

Description: None

#### 2.2.3.2.3 (METHOD PAINT CONTROL-MEASURE-POINT)

Definition 3

>saf>cm>control-measure-point.lisp

Type: Method

Arguments: (STREAM ALU)

Outputs:

Calls: WITH-INTEGER-CONVERSION-MODE

>map>utilities.lisp

WITH-MAP-GRAPHICS

>map>utilities.lisp

Called by: None

Description: None

#### 2.2.3.2.4 (METHOD DRAW-AS-FIRST-POINT CONTROL-MEASURE-POINT)

Definition 4

>saf>cm>control-measure-point.lisp

Type: Method

Arguments: (STREAM)

Outputs:

Calls: WITH-INTEGER-CONVERSION-MODE

>map>utilities.lisp

WITH-MAP-GRAPHICS

>map>utilities.lisp

**WITH-FAST-MAP-GRAPHICS**

&gt;map&gt;utilities.lisp

**\*ERASE-OVERLAY-ALU\***

&gt;map&gt;color-map.lisp

Called by: None

Description: None

**2.2.3.2.5 (METHOD DRAW CONTROL-MEASURE-POINT)**

Definition 5

&gt;saf&gt;cm&gt;control-measure-point.lisp

Type: Method

Arguments: (STREAM)

Outputs:

Calls: **\*OVERLAY-ALU\***

&gt;map&gt;color-map.lisp

Called by: None

Description: None

**2.2.3.2.6 (METHOD ERASE CONTROL-MEASURE-POINT)**

Definition 6

&gt;saf&gt;cm&gt;control-measure-point.lisp

Type: Method

Arguments: (STREAM)

Outputs:

Calls: **\*ERASE-OVERLAY-ALU\***

&gt;map&gt;color-map.lisp

**DELETE-DISPLAYED-PRESENTATION**

&gt;saf&gt;sys&gt;utilities.lisp

Called by: None

Description: None

**2.2.3.2.7 (METHOD COPY CONTROL-MEASURE-POINT)**

Definition 7

&gt;saf&gt;cm&gt;control-measure-point.lisp

Type: Method

Arguments: ()

Outputs:

Calls: **CONTROL-MEASURE-POINT**

&gt;saf&gt;cm&gt;control-measure-point.lisp

**CONTROL-MEASURE-POINT**

&gt;saf&gt;cm&gt;control-measure-point.lisp

**CONTROL-MEASURE-POINT**

&gt;saf&gt;cm&gt;control-measure-point.lisp

Called by: None

Description: None

**2.2.3.2.8 CONTROL-MEASURE-POINT**

Definition 8

```

>saf>cm>control-measure-point.lisp
Type: COMPILE-FLAVOR-METHODS
Arguments: ()
Outputs:
Calls: None
Called by: ROUTE-POINT
>saf>cm>route-point.lisp
(PRESENTATION-MOUSE-HANDLER CONTROL-MEASURE-GESTURE)
No Source File Record
XY-LIST-TO-POINTS
>saf>cm>control-measure-point.lisp
(METHOD COPY CONTROL-MEASURE-POINT)
>saf>cm>control-measure-point.lisp
(METHOD (SETF BOX-SIZE) CONTROL-MEASURE-POINT)
No Source File Record
(METHOD BOX-SIZE CONTROL-MEASURE-POINT)
No Source File Record
Description: None

```

**2.2.3.2.9 CONTROL-MEASURE-POINT**

Definition 9

```

>saf>cm>control-measure-point.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: ROUTE-POINT
>saf>cm>route-point.lisp
(PRESENTATION-MOUSE-HANDLER CONTROL-MEASURE-GESTURE)
No Source File Record
XY-LIST-TO-POINTS
>saf>cm>control-measure-point.lisp
(METHOD COPY CONTROL-MEASURE-POINT)
>saf>cm>control-measure-point.lisp
(METHOD (SETF BOX-SIZE) CONTROL-MEASURE-POINT)
No Source File Record
(METHOD BOX-SIZE CONTROL-MEASURE-POINT)
No Source File Record
Description: None

```

**2.2.3.2.10 CONTROL-MEASURE-GESTURE**

Definition 10

```

>saf>cm>control-measure-point.lisp
Type: DEFINE-PRESENTATION-ACTION
Arguments: ()
Outputs:

```

Calls: None  
 Called by: None  
 Description: None

### 2.2.3.2.11 XY-LIST-TO-POINTS

Definition 11

>saf>cm>control-measure-point.lisp  
 Type: Function  
 Arguments: (XY-LIST)  
 Outputs:  
 Calls: CONTROL-MEASURE-POINT  
       >saf>cm>control-measure-point.lisp  
       CONTROL-MEASURE-POINT  
       >saf>cm>control-measure-point.lisp  
       CONTROL-MEASURE-POINT  
       >saf>cm>control-measure-point.lisp  
 Called by: MAKE-ZONE  
           >saf>cm>zone.lisp  
           MAKE-AREA  
           >saf>cm>area.lisp  
           MAKE-LINE  
           >saf>cm>line.lisp  
 Description: None

### 2.2.3.3 CSU cm>point.lisp

This unit contains the definition of the point control measure structure, as well as the routines to manipulate and display them. The object class for point control measures is called *cm-point*. Notice that the width and height slots for *cm-point* are *:class* slots, that is, there is only one slot value for all instances. In addition to make-behavior, review-data, and cm-intersection methods, cm-points can be erased, moved, deleted and copied.

The point control measure, called *cm-point*, differs from the control measure vertex point, called *control-measure-point*, in that *cm-point* is a kind of control measure, whereas *control-measure-point* is a component of every control measure.

#### 2.2.3.3.1 CM-POINT

Definition 1

>saf>cm>point.lisp  
 Type: DEFOBJECT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (PRESENTATION-MOUSE-HANDLER CM-POINT-GESTURE)  
           No Source File Record  
           (METHOD SEND-OVERLAY-TO-SIMHOST OVERLAY)  
           >saf>cm>overlay.lisp  
           MAKE-POINT  
           >saf>cm>point.lisp



**(PRESENTATION-FUNCTION CM-POINT HIGHLIGHTING-BOX-FUNCTION)**

No Source File Record  
 (METHOD COPY CM-POINT)  
 >saf>cm>point.lisp  
 (METHOD (SETF HEIGHT) CM-POINT)  
 No Source File Record  
 (METHOD HEIGHT CM-POINT)  
 No Source File Record  
 (METHOD (SETF WIDTH) CM-POINT)  
 No Source File Record  
 (METHOD WIDTH CM-POINT)  
 No Source File Record  
 CISS-FOR-CONTROL-MEASURE  
 >saf>sys>interim-model.lisp

Description: None

**2.2.3.3.2 CM-POINT-BEHAVIOR**

Definition 2

>saf>cm>point.lisp  
 Type: DEFOBJECT  
 Arguments: ()  
 Outputs:  
 Calls: ROUTE  
 >saf>cm>route.lisp  
 STORABLE-MIXIN  
 >saf>objects>storable-mixin.lisp  
 CONTROL-MEASURE-BEHAVIOR  
 >saf>cm>control-measure.lisp  
 ROUTE  
 >saf>cm>route.lisp  
 ROUTE  
 >saf>cm>route.lisp

Called by: None

Description: None

**2.2.3.3.3 (METHOD SEND-BEH-INFO CM-POINT-BEHAVIOR)**

Definition 3

>saf>cm>point.lisp  
 Type: Method  
 Arguments: ()  
 Outputs:  
 Calls: ROUTE  
 >saf>cm>route.lisp  
 CHANGE-SPEED  
 >saf>network>vars.lisp  
 COMBAT-INSTRUCTION-SET

```
>saf>ui>subordinate-tasking.lisp
ROUTE
>saf>cm>route.lisp
ROUTE
>saf>cm>route.lisp
Called by:  None
Description: None
```

#### 2.2.3.3.4 (METHOD COPY-BEHAVIOR CM-POINT-BEHAVIOR)

Definition 4

```
>saf>cm>point.lisp
Type: Method
Arguments: (CM)
Outputs:
Calls: ROUTE
>saf>cm>route.lisp
CHANGE-SPEED
>saf>network>vars.lisp
COMBAT-INSTRUCTION-SET
>saf>ui>subordinate-tasking.lisp
ROUTE
>saf>cm>route.lisp
ROUTE
>saf>cm>route.lisp
Called by:  None
Description: None
```

#### 2.2.3.3.5 (METHOD MAKE-BEHAVIOR CM-POINT)

Definition 5

```
>saf>cm>point.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by:  None
Description: None
```

#### 2.2.3.3.6 (METHOD REVIEW-DATA CM-POINT)

Definition 6

```
>saf>cm>point.lisp
Type: Method
Arguments: ()
Outputs:
Calls: *LAST-UNITS-SPEED*
>saf>sys>vars.lisp
M/SEC-TO-SPEED
>saf>sys>utilities.lisp
SPEED-TO-M/SEC
```

```

>saf>sys>utilities.lisp
CHANGE-SPEED
>saf>network>vars.lisp
COMBAT-INSTRUCTION-SET
>saf>ui>subordinate-tasking.lisp
*PREV-UNITS*
>saf>cm>control-measure.lisp
CM-UNIT
>saf>cm>control-measure.lisp
CM-CIS
>saf>cm>control-measure.lisp
CM-SPEED
>saf>cm>control-measure.lisp

```

Called by: None  
 Description: None

#### 2.2.3.3.7 (METHOD DRAW CM-POINT) Definition 7

```

>saf>cm>point.lisp
Type: Method
Arguments: (STREAM)
Outputs:
Calls: WITH-INTEGER-CONVERSION-MODE
>map>utilities.lisp
WITH-MAP-GRAPHICS
>map>utilities.lisp
*OVERLAY-ALU*
>map>color-map.lisp
CONTROL-MEASURE-LABEL
>saf>cm>control-measure.lisp

```

Called by: None  
 Description: None

#### 2.2.3.3.8 (METHOD ERASE CM-POINT) Definition 8

```

>saf>cm>point.lisp
Type: Method
Arguments: (STREAM)
Outputs:
Calls: WITH-INTEGER-CONVERSION-MODE
>map>utilities.lisp
WITH-MAP-GRAPHICS
>map>utilities.lisp
WITH-FAST-MAP-GRAPHICS
>map>utilities.lisp
*ERASE-OVERLAY-ALU*
>map>color-map.lisp
DELETE-DISPLAYED-PRESENTATION
>saf>sys>utilities.lisp

```

Called by: None  
 Description: None

**2.2.3.3.9 (METHOD MOVE-POINT CM-POINT)**

## Definition 9

```
>saf>cm>point.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: SINGLE-POINT
       >map>control.lisp
       *PVD-DISPLAY*
       >saf>sys>vars.lisp
       POINT
       >saf>interface>model-menu.lisp
       POINT
       >saf>interface>model-menu.lisp
Called by: None
Description: None
```

**2.2.3.3.10 (METHOD DELETE-POINT CM-POINT)**

## Definition 10

```
>saf>cm>point.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: *PVD-DISPLAY*
       >saf>sys>vars.lisp
       POINT
       >saf>interface>model-menu.lisp
       POINT
       >saf>interface>model-menu.lisp
Called by: None
Description: None
```

**2.2.3.3.11 (METHOD SEND-CM-INFO CM-POINT)**

## Definition 11

```
>saf>cm>point.lisp
Type: Method
Arguments: (BEH)
Outputs:
Calls: None
Called by: None
Description: None
```

**2.2.3.3.12 (METHOD COPY CM-POINT)**

## Definition 12

```
>saf>cm>point.lisp
Type: Method
Arguments: ()
```

## Outputs:

Calls: NAME  
       >saf>sysdcl.lisp  
       CM-POINT  
       >saf>cm>point.lisp  
       CM-POINT  
       >saf>cm>point.lisp  
       CM-POINT  
       >saf>cm>point.lisp

Called by: None

Description: None

### 2.2.3.3.13 (METHOD CM-INTERSECTION CM-POINT)

Definition 13

      >saf>cm>point.lisp  
 Type: Method  
 Arguments: (P1 P2)  
 Outputs:  
 Calls: DISTANCE  
       >map>utilities.lisp  
       VEC-SUB  
       >map>vectors.lisp

Called by: None

Description: None

### 2.2.3.3.14 CM-POINT

Definition 14

      >saf>cm>point.lisp  
 Type: COMPILE-FLAVOR-METHODS  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (PRESENTATION-MOUSE-HANDLER CM-POINT-GESTURE)  
           No Source File Record  
           (METHOD SEND-OVERLAY-TO-SIMHOST OVERLAY)  
           >saf>cm>overlay.lisp  
           MAKE-POINT  
           >saf>cm>point.lisp  
           (PRESENTATION-FUNCTION CM-POINT HIGHLIGHTING-BOX-  
 FUNCTION)  
           No Source File Record  
           (METHOD COPY CM-POINT)  
           >saf>cm>point.lisp  
           (METHOD (SETF HEIGHT) CM-POINT)  
           No Source File Record  
           (METHOD HEIGHT CM-POINT)  
           No Source File Record

(METHOD (SETF WIDTH) CM-POINT)

No Source File Record

(METHOD WIDTH CM-POINT)

No Source File Record

CISS-FOR-CONTROL-MEASURE

>saf>sys>interim-model.lisp

Description: None

### 2.2.3.3.15 CM-POINT

Definition 15

>saf>cm>point.lisp

Type: DEFINE-PRESENTATION-TYPE

Arguments: ()

Outputs:

Calls: None

Called by: (PRESENTATION-MOUSE-HANDLER CM-POINT-GESTURE)

No Source File Record

(METHOD SEND-OVERLAY-TO-SIMHOST OVERLAY)

>saf>cm>overlay.lisp

MAKE-POINT

>saf>cm>point.lisp

(PRESENTATION-FUNCTION CM-POINT HIGHLIGHTING-BOX-FUNCTION)

No Source File Record

(METHOD COPY CM-POINT)

>saf>cm>point.lisp

(METHOD (SETF HEIGHT) CM-POINT)

No Source File Record

(METHOD HEIGHT CM-POINT)

No Source File Record

(METHOD (SETF WIDTH) CM-POINT)

No Source File Record

(METHOD WIDTH CM-POINT)

No Source File Record

CISS-FOR-CONTROL-MEASURE

>saf>sys>interim-model.lisp

Description: None

### 2.2.3.3.16 CM-POINT-GESTURE

Definition 16

>saf>cm>point.lisp

Type: DEFINE-PRESENTATION-ACTION

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.3.3.17 MAKE-POINT**

Definition 17

```

>saf>cm>point.lisp
Type: Function
Arguments: (OVERLAY STREAM)
Outputs:
Calls: SINGLE-POINT
       >map>control.lisp
       POINT
       >saf>interface>model-menu.lisp
       CM-POINT
       >saf>cm>point.lisp
       CM-POINT
       >saf>cm>point.lisp
       CM-POINT
       >saf>cm>point.lisp
       POINT
       >saf>interface>model-menu.lisp
Called by: (METHOD ADD-NEW-CONTROL-MEASURE OVERLAY)
          >saf>cm>overlay.lisp
Description: None

```

**2.2.3.4 CSU cm>line.lisp**

This unit contains the definition of the line control measure structure, as well as the routines to manipulate and display them. In addition to *make-behavior*, *review-data* and *cm-intersection*, methods are included for moving, deleting and inserting the points that make up the line.

**2.2.3.4.1 LINE**

Definition 1

```

>saf>cm>line.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: None
Called by: MAKE-GRAPH-GIVEN-POINTS
          >saf>interface>model-menu.lisp
          MAKE-LINE
          >saf>cm>line.lisp
          FIND-SEGMENT-CROSS-POINTS
          >saf>cm>water-avoidance.lisp
          PRINT-MESSAGE
          >saf>rudp>handle-incoming.lisp
          SEND-LINE
          >saf>network>commands.lisp
          (METHOD TOP-LEVEL CONFIGURATION-MENU)
          >saf>interface>formations.lisp

```

MAKE-LINE  
>saf>cm>line.lisp  
(METHOD COPY LINE)  
>saf>cm>line.lisp  
(METHOD INTERVENE SIMNET-AGENT FORMATION)  
>saf>objects>intervention.lisp

Description: None

#### 2.2.3.4.2 LINE-BEHAVIOR

Definition 2

>saf>cm>line.lisp  
Type: DEFOBJECT  
Arguments: ()  
Outputs:  
Calls: STORABLE-MIXIN  
>saf>objects>storable-mixin.lisp  
CONTROL-MEASURE-BEHAVIOR  
>saf>cm>control-measure.lisp

Called by: None

Description: None

#### 2.2.3.4.3 (METHOD SEND-BEH-INFO LINE-BEHAVIOR)

Definition 3

>saf>cm>line.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: CHANGE-SPEED  
>saf>network>vars.lisp  
COMBAT-INSTRUCTION-SET  
>saf>ui>subordinate-tasking.lisp

Called by: None

Description: None

#### 2.2.3.4.4 (METHOD COPY-BEHAVIOR LINE-BEHAVIOR)

Definition 4

>saf>cm>line.lisp  
Type: Method  
Arguments: (CM)  
Outputs:  
Calls: CHANGE-SPEED  
>saf>network>vars.lisp  
COMBAT-INSTRUCTION-SET  
>saf>ui>subordinate-tasking.lisp

Called by: None

Description: None



**2.2.3.4.5 (METHOD MAKE-BEHAVIOR LINE)**

## Definition 5

`>saf>cm>line.lisp`

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.3.4.6 (METHOD MAKE-INSTANCE LINE AFTER)**

## Definition 6

`>saf>cm>line.lisp`

Type: Method

Arguments: (&amp;REST INIT-ARGS)

Outputs:

Calls: None

Called by: None

Description: None

**2.2.3.4.7 (METHOD INITIALIZE-POINTS LINE)**

## Definition 7

`>saf>cm>line.lisp`

Type: Method

Arguments: (POINT-LIST)

Outputs:

Calls: POINT

`>saf>interface>model-menu.lisp`

SAF

`>saf>ui>frame.lisp`

CONTROL-MEASURE

`>saf>cm>control-measure.lisp`

CONTROL-MEASURE

`>saf>cm>control-measure.lisp`

POINT

`>saf>interface>model-menu.lisp`

Called by: None

Description: None

**2.2.3.4.8 (METHOD REVIEW-DATA LINE)**

## Definition 8

`>saf>cm>line.lisp`

Type: Method

Arguments: ()

Outputs:

Calls: \*LAST-UNITS-SPEED\*  
>saf>sys>vars.lisp  
M/SEC-TO-SPEED  
>saf>sys>utilities.lisp  
SPEED-TO-M/SEC  
>saf>sys>utilities.lisp  
CHANGE-SPEED  
>saf>network>vars.lisp  
COMBAT-INSTRUCTION-SET  
>saf>ui>subordinate-tasking.lisp  
\*PREV-UNITS\*  
>saf>cm>control-measure.lisp  
CM-UNIT  
>saf>cm>control-measure.lisp  
CM-CIS  
>saf>cm>control-measure.lisp  
CM-SPEED  
>saf>cm>control-measure.lisp

Called by: None

Description: None

#### 2.2.3.4.9 (METHOD PAINT-NAME LINE)

Definition 9

>saf>cm>line.lisp

Type: Method

Arguments: (STREAM ALU)

Outputs:

Calls: None

Called by: None

Description: None

#### 2.2.3.4.10 (DRAW-SEGMENT LINE)

Definition 10

>saf>cm>line.lisp

Type: DEFSUBST-IN-FLAVOR

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.2.3.4.11 (METHOD PAINT LINE)

Definition 11

>saf>cm>line.lisp

Type: Method

Arguments: (STREAM ALU)

Outputs:

**Calls:** WITH-INTEGER-CONVERSION-MODE

>map>utilities.lisp

WITH-MAP-GRAPHICS

>map>utilities.lisp

WITH-FAST-MAP-GRAPHICS

>map>utilities.lisp

**Called by:** None

**Description:** None

#### 2.2.3.4.12 (METHOD DRAW LINE)

Definition 12

>saf>cm>line.lisp

**Type:** Method

**Arguments:** (STREAM)

**Outputs:**

**Calls:** \*OVERLAY-ALU\*

>map>color-map.lisp

**Called by:** None

**Description:** None

#### 2.2.3.4.13 (METHOD ERASE LINE)

Definition 13

>saf>cm>line.lisp

**Type:** Method

**Arguments:** (STREAM)

**Outputs:**

**Calls:** \*ERASE-OVERLAY-ALU\*

>map>color-map.lisp

**Called by:** None

**Description:** None

#### 2.2.3.4.14 (METHOD ORTHOGONALIZE LINE)

Definition 14

>saf>cm>line.lisp

**Type:** Method

**Arguments:** ()

**Outputs:**

**Calls:** POINT

>saf>interface>model-menu.lisp

POINT

>saf>interface>model-menu.lisp

**Called by:** None

**Description:** None

**2.2.3.4.15 (METHOD MOVE-POINT LINE)**

## Definition 15

>saf>cm>line.lisp  
Type: Method  
Arguments: (POINT)  
Outputs:  
Calls: WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
\*OVERLAY-ALU\*  
>map>color-map.lisp  
\*ERASE-OVERLAY-ALU\*  
>map>color-map.lisp  
SINGLE-POINT  
>map>control.lisp  
\*PVD-DISPLAY\*  
>saf>sys>vars.lisp  
POINT  
>saf>interface>model-menu.lisp  
POINT  
>saf>interface>model-menu.lisp  
Called by: None  
Description: None

**2.2.3.4.16 (METHOD DELETE-POINT LINE)**

## Definition 16

>saf>cm>line.lisp  
Type: Method  
Arguments: (POINT)  
Outputs:  
Calls: WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
\*OVERLAY-ALU\*  
>map>color-map.lisp  
\*ERASE-OVERLAY-ALU\*  
>map>color-map.lisp  
\*PVD-DISPLAY\*  
>saf>sys>vars.lisp  
POINT  
>saf>interface>model-menu.lisp  
POINT  
>saf>interface>model-menu.lisp  
Called by: None  
Description: None

**2.2.3.4.17 (METHOD INSERT-POINT-AFTER LINE)**

Definition 17

```

>saf>cm>line.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: WITH-INTEGER-CONVERSION-MODE
       >map>utilities.lisp
       WITH-MAP-GRAPHICS
       >map>utilities.lisp
       WITH-FAST-MAP-GRAPHICS
       >map>utilities.lisp
       *OVERLAY-ALU*
       >map>color-map.lisp
       *ERASE-OVERLAY-ALU*
       >map>color-map.lisp
       SINGLE-POINT
       >map>control.lisp
       NAME
       >saf>sysdcl.lisp
       *PVD-DISPLAY*
       >saf>sys>vars.lisp
       POINT
       >saf>interface>model-menu.lisp
       SAF
       >saf>ui>frame.lisp
       POINT
       >saf>interface>model-menu.lisp
Called by: None
Description: None

```

**2.2.3.4.18 (METHOD INSERT-POINT-BEFORE LINE)**

Definition 18

```

>saf>cm>line.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: WITH-INTEGER-CONVERSION-MODE
       >map>utilities.lisp
       WITH-MAP-GRAPHICS
       >map>utilities.lisp
       WITH-FAST-MAP-GRAPHICS
       >map>utilities.lisp
       *OVERLAY-ALU*
       >map>color-map.lisp
       *ERASE-OVERLAY-ALU*
       >map>color-map.lisp
       SINGLE-POINT
       >map>control.lisp
       NAME
       >saf>sysdcl.lisp

```

```
*PVD-DISPLAY*
>saf>sys>vars.lisp
POINT
>saf>interface>model-menu.lisp
SAF
>saf>ui>frame.lisp
POINT
>saf>interface>model-menu.lisp
```

Called by: None

Description: None

#### 2.2.3.4.19 (METHOD SEND-CM-INFO LINE)

Definition 19

```
>saf>cm>line.lisp
Type: Method
Arguments: (BEH)
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.2.3.4.20 (METHOD MOVE-CONTROL-MEASURE LINE)

Definition 20

```
>saf>cm>line.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: SINGLE-POINT
>map>control.lisp
*PVD-DISPLAY*
>saf>sys>vars.lisp
POINT
>saf>interface>model-menu.lisp
POINT
>saf>interface>model-menu.lisp
Called by: None
Description: None
```

#### 2.2.3.4.21 (METHOD COPY LINE)

Definition 21

```
>saf>cm>line.lisp
Type: Method
Arguments: ()
Outputs:
Calls: NAME
>saf>sysdcl.lisp
POINT
>saf>interface>model-menu.lisp
LINE
```

```

>saf>cm>line.lisp
CONTROL-MEASURE
>saf>cm>control-measure.lisp
CONTROL-MEASURE
>saf>cm>control-measure.lisp
LINE
>saf>cm>line.lisp
LINE
>saf>cm>line.lisp
POINT
>saf>interface>model-menu.lisp

```

Called by: None

Description: None

#### 2.2.3.4.22 (METHOD CM-INTERSECTION LINE)

Definition 22

```

>saf>cm>line.lisp
Type: Method
Arguments: (P1 P2)
Outputs:
Calls: None
Called by: None
Description: None

```

#### 2.2.3.4.23 LINE

Definition 23

```

>saf>cm>line.lisp
Type: COMPILE-FLAVOR-METHODS
Arguments: ()
Outputs:
Calls: None
Called by: MAKE-GRAPH-GIVEN-POINTS
>saf>interface>model-menu.lisp
MAKE-LINE
>saf>cm>line.lisp
FIND-SEGMENT-CROSS-POINTS
>saf>cm>water-avoidance.lisp
PRINT-MESSAGE
>saf>rudp>handle-incoming.lisp
SEND-LINE
>saf>network>commands.lisp
(METHOD TOP-LEVEL CONFIGURATION-MENU)
>saf>interface>formations.lisp
MAKE-LINE
>saf>cm>line.lisp
(METHOD COPY LINE)
>saf>cm>line.lisp
(METHOD INTERVENE SIMNET-AGENT FORMATION)
>saf>objects>intervention.lisp
Description: None

```

**2.2.3.4.24 MAKE-LINE**

Definition 24

```
>saf>cm>line.lisp
Type: Function
Arguments: (OVERLAY STREAM)
Outputs:
Calls: RUBBER-LINE
       >map>control.lisp
       LINE
       >saf>cm>line.lisp
       LINE
       >saf>cm>line.lisp
       REVERSE-XY
       >saf>cm>control-measure.lisp
       XY-LIST-TO-POINTS
       >saf>cm>control-measure-point.lisp
       LINE
       >saf>cm>line.lisp
       LINE
       >saf>cm>line.lisp
       LINE
       >saf>cm>line.lisp
       LINE
       >saf>cm>line.lisp
Called by: (METHOD ADD-NEW-CONTROL-MEASURE OVERLAY)
          >saf>cm>overlay.lisp
Description: None
```

**2.2.3.5 CSU cm>generic-area.lisp**

This unit contains the definition of the generic area control measure structure, which area and zone control measures are specializations of. It has methods for making instances, and for display, moving, erasing, deleting, and inserting control-measure points.

**2.2.3.5.1 GENERIC-AREA**

Definition 1

```
>saf>cm>generic-area.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: None
Called by: ZONE
          >saf>cm>zone.lisp
          AREA
          >saf>cm>area.lisp
          GENERIC-AREA?
          >saf>cm>generic-area.lisp
Description: None
```



**2.2.3.5.2 GENERIC-AREA?**

## Definition 2

```
>saf>cm>generic-area.lisp
Type: Function
Arguments: (CM)
Outputs:
Calls: GENERIC-AREA
       >saf>cm>generic-area.lisp
       GENERIC-AREA
       >saf>cm>generic-area.lisp
Called by: None
Description: None
```

**2.2.3.5.3 (METHOD MAKE-INSTANCE GENERIC-AREA AFTER)**

## Definition 3

```
>saf>cm>generic-area.lisp
Type: Method
Arguments: (&REST INIT-ARGS)
Outputs:
Calls: None
Called by: None
Description: None
```

**2.2.3.5.4 (METHOD INITIALIZE-POINTS GENERIC-AREA)**

## Definition 4

```
>saf>cm>generic-area.lisp
Type: Method
Arguments: (POINT-LIST)
Outputs:
Calls: POINT
       >saf>interface>model-menu.lisp
       SAF
       >saf>ui>frame.lisp
       CONTROL-MEASURE
       >saf>cm>control-measure.lisp
       CONTROL-MEASURE
       >saf>cm>control-measure.lisp
       POINT
       >saf>interface>model-menu.lisp
Called by: None
Description: None
```

**2.2.3.5.5 (METHOD PAINT-NAME GENERIC-AREA)**

## Definition 5

```
>saf>cm>generic-area.lisp
Type: Method
Arguments: (STREAM ALU)
```

## Outputs:

Calls: POINT

&gt;saf&gt;interface&gt;model-menu.lisp

POINT

&gt;saf&gt;interface&gt;model-menu.lisp

Called by: None

Description: None

**2.2.3.5.6 (METHOD PAINT GENERIC-AREA)**

## Definition 6

&gt;saf&gt;cm&gt;generic-area.lisp

Type: Method

Arguments: (STREAM ALU)

## Outputs:

Calls: WITH-INTEGER-CONVERSION-MODE

&gt;map&gt;utilities.lisp

WITH-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

WITH-FAST-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

Called by: None

Description: None

**2.2.3.5.7 (METHOD DRAW GENERIC-AREA)**

## Definition 7

&gt;saf&gt;cm&gt;generic-area.lisp

Type: Method

Arguments: (STREAM)

## Outputs:

Calls: \*OVERLAY-ALU\*

&gt;map&gt;color-map.lisp

Called by: None

Description: None

**2.2.3.5.8 (METHOD ERASE GENERIC-AREA)**

## Definition 8

&gt;saf&gt;cm&gt;generic-area.lisp

Type: Method

Arguments: (STREAM)

## Outputs:

Calls: \*ERASE-OVERLAY-ALU\*

&gt;map&gt;color-map.lisp

Called by: None

Description: None

**2.2.3.5.9 (METHOD MOVE-POINT GENERIC-AREA)**

## Definition 9

```
>saf>cm>generic-area.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: WITH-INTEGER-CONVERSION-MODE
       >map>utilities.lisp
       WITH-MAP-GRAPHICS
       >map>utilities.lisp
       WITH-FAST-MAP-GRAPHICS
       >map>utilities.lisp
       *OVERLAY-ALU*
       >map>color-map.lisp
       *ERASE-OVERLAY-ALU*
       >map>color-map.lisp
       SINGLE-POINT
       >map>control.lisp
       *PVD-DISPLAY*
       >saf>sys>vars.lisp
       POINT
       >saf>interface>model-menu.lisp
       POINT
       >saf>interface>model-menu.lisp
Called by: None
Description: None
```

**2.2.3.5.10 (METHOD DELETE-POINT GENERIC-AREA)**

## Definition 10

```
>saf>cm>generic-area.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: WITH-INTEGER-CONVERSION-MODE
       >map>utilities.lisp
       WITH-MAP-GRAPHICS
       >map>utilities.lisp
       WITH-FAST-MAP-GRAPHICS
       >map>utilities.lisp
       *OVERLAY-ALU*
       >map>color-map.lisp
       *ERASE-OVERLAY-ALU*
       >map>color-map.lisp
       *PVD-DISPLAY*
       >saf>sys>vars.lisp
       POINT
       >saf>interface>model-menu.lisp
       POINT
       >saf>interface>model-menu.lisp
Called by: None
Description: None
```

**2.2.3.5.11 (METHOD INSERT-POINT-AFTER GENERIC-AREA)**

Definition 11

```
>saf>cm>generic-area.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: WITH-INTEGER-CONVERSION-MODE
      >map>utilities.lisp
      WITH-MAP-GRAPHICS
      >map>utilities.lisp
      WITH-FAST-MAP-GRAPHICS
      >map>utilities.lisp
      *OVERLAY-ALU*
      >map>color-map.lisp
      *ERASE-OVERLAY-ALU*
      >map>color-map.lisp
      SINGLE-POINT
      >map>control.lisp
      NAME
      >saf>sysdcl.lisp
      *PVD-DISPLAY*
      >saf>sys>vars.lisp
      POINT
      >saf>interface>model-menu.lisp .
      SAF
      >saf>ui>frame.lisp
      POINT
      >saf>interface>model-menu.lisp
Called by: None
Description: None
```

**2.2.3.5.12 (METHOD INSERT-POINT-BEFORE GENERIC-AREA)**

Definition 12

```
>saf>cm>generic-area.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: POINT
      >saf>interface>model-menu.lisp
      POINT
      >saf>interface>model-menu.lisp
Called by: None
Description: None
```

**2.2.3.5.13 (METHOD ORTHOGONALIZE GENERIC-AREA)**

Definition 13

```

>saf>cm>generic-area.lisp
Type: Method
Arguments: ()
Outputs:
Calls: POINT
       >saf>interface>model-menu.lisp
       POINT
       >saf>interface>model-menu.lisp
Called by: None
Description: None

```

**2.2.3.5.14 (METHOD SEND-CM-INFO GENERIC-AREA)**

Definition 14

```

>saf>cm>generic-area.lisp
Type: Method
Arguments: (BEH)
Outputs:
Calls: CHANGE-SPEED
       >saf>network>vars.lisp
       COMBAT-INSTRUCTION-SET
       >saf>ui>subordinate-tasking.lisp
Called by: None
Description: None

```

**2.2.3.5.15 GENERIC-AREA**

Definition 15

```

>saf>cm>generic-area.lisp
Type: COMPILE-FLAVOR-METHODS
Arguments: ()
Outputs:
Calls: None
Called by: ZONE
       >saf>cm>zone.lisp
       AREA
       >saf>cm>area.lisp
       GENERIC-AREA?
       >saf>cm>generic-area.lisp
Description: None

```

**2.2.3.6 CSU cm>area.lisp**

This unit contains the definition of the area control measure structure, as well as the routines to manipulate and display them. In addition to *make-behavior*, *review-data* and *cm-intersection*, methods are included for moving, deleting and inserting the points that make up the area boundary.

The code for area and zone control measures is nearly identical. They differ in the "Types" shown in the review-data menu. The types for the area control-measure are *Assembly Area* and *Battle Position*.

### 2.2.3.6.1 AREA

#### Definition 1

```

>saf>cm>area.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: None
Called by: MAKE-ZONE-BEHAVIOR
>saf>cm>zone.lisp
MAKE-AREA
>saf>cm>area.lisp
MAKE-AREA-BEHAVIOR
>saf>cm>area.lisp
MAKE-LINE-BEHAVIOR
>saf>cm>line.lisp
MAKE-POINT-BEHAVIOR
>saf>cm>point.lisp
MAKE-ROUTE-BEHAVIOR
>saf>cm>route.lisp
MAKE-ROUTE-POINT
>saf>cm>route-point.lisp
MAKE-CONTROL-MEASURE-POINT
>saf>cm>control-measure-point.lisp
SEND-AREA
>saf>network>commands.lisp
MAKE-AREA
>saf>cm>area.lisp
(METHOD COPY AREA)
>saf>cm>area.lisp
Description: None

```

### 2.2.3.6.2 AREA-BEHAVIOR

#### Definition 2

```

>saf>cm>area.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: STORABLE-MIXIN
>saf>objects>storable-mixin.lisp
CONTROL-MEASURE-BEHAVIOR
>saf>cm>control-measure.lisp
Called by: None
Description: None

```

**2.2.3.6.3 (METHOD MAKE-BEHAVIOR AREA)**

Definition 3

>saf>cm>area.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.3.6.4 (METHOD COPY-BEHAVIOR AREA-BEHAVIOR)**

Definition 4

>saf>cm>area.lisp  
Type: Method  
Arguments: (CM)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.3.6.5 (METHOD REVIEW-DATA AREA)**

Definition 5

>saf>cm>area.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: COMBAT-INSTRUCTION-SET  
>saf>ui>subordinate-tasking.lisp  
\*PREV-UNITS\*  
>saf>cm>control-measure.lisp  
CM-UNIT  
>saf>cm>control-measure.lisp  
CM-CIS  
>saf>cm>control-measure.lisp  
Called by: None  
Description: None

**2.2.3.6.6 (METHOD COPY AREA)**

Definition 6

>saf>cm>area.lisp  
Type: Method  
Arguments: ()  
Outputs:

Calls: NAME  
       >saf>sysdcl.lisp  
       POINT  
       >saf>interface>model-menu.lisp  
       AREA  
       >saf>cm>area.lisp  
       CONTROL-MEASURE  
       >saf>cm>control-measure.lisp  
       CONTROL-MEASURE  
       >saf>cm>control-measure.lisp  
       AREA  
       >saf>cm>area.lisp  
       AREA  
       >saf>cm>area.lisp  
       POINT  
       >saf>interface>model-menu.lisp

Called by: None

Description: None

#### 2.2.3.6.7 (METHOD MOVE-CONTROL-MEASURE AREA)

Definition 7

      >saf>cm>area.lisp  
 Type: Method  
 Arguments: (POINT)  
 Outputs:  
 Calls: SINGLE-POINT  
       >map>control.lisp  
       \*PVD-DISPLAY\*  
       >saf>sys>vars.lisp  
       POINT  
       >saf>interface>model-menu.lisp  
       POINT  
       >saf>interface>model-menu.lisp

Called by: None

Description: None

#### 2.2.3.6.8 (METHOD CM-INTERSECTION AREA)

Definition 8

      >saf>cm>area.lisp  
 Type: Method  
 Arguments: (P1 P2)  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None



**2.2.3.6.9 AREA**

## Definition 9

```

>saf>cm>area.lisp
Type: COMPILE-FLAVOR-METHODS
Arguments: ()
Outputs:
Calls: None
Called by: MAKE-ZONE-BEHAVIOR
>saf>cm>zone.lisp
MAKE-AREA
>saf>cm>area.lisp
MAKE-AREA-BEHAVIOR
>saf>cm>area.lisp
MAKE-LINE-BEHAVIOR
>saf>cm>line.lisp
MAKE-POINT-BEHAVIOR
>saf>cm>point.lisp
MAKE-ROUTE-BEHAVIOR
>saf>cm>route.lisp
MAKE-ROUTE-POINT
>saf>cm>route-point.lisp
MAKE-CONTROL-MEASURE-POINT
>saf>cm>control-measure-point.lisp
SEND-AREA
>saf>network>commands.lisp
MAKE-AREA
>saf>cm>area.lisp
(METHOD COPY AREA)
>saf>cm>area.lisp
Description: None

```

**2.2.3.6.10 MAKE-AREA**

## Definition 10

```

>saf>cm>area.lisp
Type: Function
Arguments: (OVERLAY STREAM)
Outputs:
Calls: RUBBER-LINE
>map>control.lisp
AREA
>saf>cm>area.lisp
AREA
>saf>cm>area.lisp
REVERSE-XY
>saf>cm>control-measure.lisp
XY-LIST-TO-POINTS
>saf>cm>control-measure-point.lisp

```

```

AREA
>saf>cm>area.lisp
AREA
>saf>cm>area.lisp
AREA
>saf>cm>area.lisp
AREA
>saf>cm>area.lisp
Called by: (METHOD ADD-NEW-CONTROL-MEASURE OVERLAY)
>saf>cm>overlay.lisp
Description: None

```

### 2.2.3.7 CSU cm>zone.lisp

This unit contains the definition of the zone control measure structure, as well as the routines to manipulate and display them. In addition to *make-behavior*, *review-data* and *cm-intersection*, methods are included for moving, deleting and inserting the points that make up the zone boundary. The types for the zone control-measure are *Recon/Surveillance Zone*, *No Fire Zone*, and *Transit Corridor*.

#### 2.2.3.7.1 ZONE

Definition 1

```

>saf>cm>zone.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: None
Called by: MAKE-ZONE
>saf>cm>zone.lisp
SEND-ZONE
>saf>network>commands.lisp
MAKE-ZONE
>saf>cm>zone.lisp
(METHOD COPY ZONE)
>saf>cm>zone.lisp
Description: None

```

#### 2.2.3.7.2 ZONE-BEHAVIOR

Definition 2

```

>saf>cm>zone.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: STORABLE-MIXIN
>saf>objects>storable-mixin.lisp
CONTROL-MEASURE-BEHAVIOR
>saf>cm>control-measure.lisp
Called by: None
Description: None

```

**2.2.3.7.3 (METHOD COPY-BEHAVIOR ZONE-BEHAVIOR)**

## Definition 3

>saf>cm>zone.lisp  
Type: Method  
Arguments: (CM)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.3.7.4 (METHOD MAKE-BEHAVIOR ZONE)**

## Definition 4

>saf>cm>zone.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.3.7.5 (METHOD REVIEW-DATA ZONE)**

## Definition 5

>saf>cm>zone.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: COMBAT-INSTRUCTION-SET  
>saf>ui>subordinate-tasking.lisp  
\*PREV-UNITS\*  
>saf>cm>control-measure.lisp  
CM-UNIT  
>saf>cm>control-measure.lisp  
CM-CIS  
>saf>cm>control-measure.lisp  
Called by: None  
Description: None

**2.2.3.7.6 (METHOD COPY ZONE)**

## Definition 6

>saf>cm>zone.lisp  
Type: Method  
Arguments: ()  
Outputs:

Calls: NAME  
       >saf>sysdcl.lisp  
       POINT  
       >saf>interface>model-menu.lisp  
       ZONE  
       >saf>cm>zone.lisp  
       CONTROL-MEASURE  
       >saf>cm>control-measure.lisp  
       CONTROL-MEASURE  
       >saf>cm>control-measure.lisp  
       ZONE  
       >saf>cm>zone.lisp  
       ZONE  
       >saf>cm>zone.lisp  
       POINT  
       >saf>interface>model-menu.lisp

Called by: None

Description: None

### 2.2.3.7.7 (METHOD MOVE-CONTROL-MEASURE ZONE)

Definition 7

      >saf>cm>zone.lisp  
 Type: Method  
 Arguments: (POINT)  
 Outputs:  
 Calls: SINGLE-POINT  
       >map>control.lisp  
       \*PVD-DISPLAY\*  
       >saf>sys>vars.lisp  
       POINT  
       >saf>interface>model-menu.lisp  
       POINT  
       >saf>interface>model-menu.lisp

Called by: None

Description: None

### 2.2.3.7.8 (METHOD CM-INTERSECTION ZONE)

Definition 8

      >saf>cm>zone.lisp  
 Type: Method  
 Arguments: (P1 P2)  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

**2.2.3.7.9 ZONE**

## Definition 9

&gt;saf&gt;cm&gt;zone.lisp

Type: COMPILE-FLAVOR-METHODS

Arguments: ()

Outputs:

Calls: None

Called by: MAKE-ZONE

&gt;saf&gt;cm&gt;zone.lisp

SEND-ZONE

&gt;saf&gt;network&gt;commands.lisp

MAKE-ZONE

&gt;saf&gt;cm&gt;zone.lisp

(METHOD COPY ZONE)

&gt;saf&gt;cm&gt;zone.lisp

Description: None

**2.2.3.7.10 MAKE-ZONE**

## Definition 10

&gt;saf&gt;cm&gt;zone.lisp

Type: Function

Arguments: (OVERLAY STREAM)

Outputs:

Calls: RUBBER-LINE

&gt;map&gt;control.lisp

ZONE

&gt;saf&gt;cm&gt;zone.lisp

ZONE

&gt;saf&gt;cm&gt;zone.lisp

REVERSE-XY

&gt;saf&gt;cm&gt;control-measure.lisp

XY-LIST-TO-POINTS

&gt;saf&gt;cm&gt;control-measure-point.lisp

ZONE

&gt;saf&gt;cm&gt;zone.lisp

ZONE

&gt;saf&gt;cm&gt;zone.lisp

ZONE

&gt;saf&gt;cm&gt;zone.lisp

ZONE

&gt;saf&gt;cm&gt;zone.lisp

Called by: (METHOD ADD-NEW-CONTROL-MEASURE OVERLAY)

&gt;saf&gt;cm&gt;overlay.lisp

Description: None

### 2.2.4 Routes CSC

This CSC contains the code to create, manipulate and check routes. It supports the automatic generation of road routes via an A\* algorithm and the automatic checking for routes that would cause a unit to go through unfordable water. The CSUs in this CSC are:

```
cm>water-avoidance.lisp csu
cm>water-check.lisp csu
cm>route-point.lisp csu
cm>road-routes.lisp csu
cm>route-finder.lisp csu
cm>route.lisp csu
```

#### 2.2.4.1 CSU cm>water-avoidance.lisp

This unit contains routines that generate ground routes around water, using bridges, fording points and ends of water segments. A vector approach is used to find crossings for each water segment encountered in turn. Routines to skirt lakes and river bends are also included in this unit.

The algorithm for generating these water-avoiding routes is somewhat complex; a conceptual overview is provided in Appendix A1, Water Avoidance Algorithm.

##### 2.2.4.1.1 \*INTERSECTIONS-SEARCHED\*

Definition 1

```
>saf>cm>water-avoidance.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: FOLLOW-WATER-SEGMENTS
           >saf>cm>water-avoidance.lisp
           FIND-ROUTE-CORE
           >saf>cm>water-avoidance.lisp
Description: None
```

##### 2.2.4.1.2 \*QUADS-INDEX-LIST\*

Definition 2

```
>saf>cm>water-avoidance.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: FINAL-RELAX-POINTS
           >saf>cm>water-avoidance.lisp
           RELAX-POINTS-AUX
           >saf>cm>water-avoidance.lisp
           RELAX-POINTS
           >saf>cm>water-avoidance.lisp
```

**FIND-SUITABLE-CROSSING-ROUTE**

&gt;saf&gt;cm&gt;water-avoidance.lisp

**FIND-ROUTE-CORE**

&gt;saf&gt;cm&gt;water-avoidance.lisp

**FIND-ROUTE-AROUND-WATER**

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.3 FIND-ROUTE-AROUND-WATER**

## Definition 3

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (X1 Y1 X2 Y2 &amp;OPTIONAL (OFFSET 50))

Outputs:

Calls: \*OPFOR-IO\*

&gt;saf&gt;sys&gt;vars.lisp

SAY

&gt;saf&gt;sys&gt;macros.lisp

ROUTE

&gt;saf&gt;cm&gt;route.lisp

ALL-WIDE-SEGMENTS-THRU-WATER

&gt;saf&gt;cm&gt;water-check.lisp

\*QUADS-INDEX-LIST\*

&gt;saf&gt;cm&gt;water-avoidance.lisp

FIND-ROUTE-CORE

&gt;saf&gt;cm&gt;water-avoidance.lisp

FIRST-ITEMS

&gt;saf&gt;cm&gt;water-avoidance.lisp

FINAL-RELAX-POINTS

&gt;saf&gt;cm&gt;water-avoidance.lisp

GET-QUADS-IN-REGION

&gt;saf&gt;cm&gt;water-avoidance.lisp

ROUTE

&gt;saf&gt;cm&gt;route.lisp

ROUTE

&gt;saf&gt;cm&gt;route.lisp

Called by: (METHOD CHECK-ROUTE-SEGMENT ROUTE)

&gt;saf&gt;cm&gt;route.lisp

Description: None

**2.2.4.1.4 THRU-RIVER-BEND**

## Definition 4

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (WATER-LIST)

Outputs:

Calls: None

Called by: FIND-SUITABLE-CROSSING-ROUTE

>saf>cm>water-avoidance.lisp

FIND-ROUTE-CORE

>saf>cm>water-avoidance.lisp

Description: None

#### 2.2.4.1.5 FIND-ROUTE-CORE

Definition 5

>saf>cm>water-avoidance.lisp

Type: Function

Arguments: (POINT-LIST WATER-LIST &OPTIONAL (OFFSET 100.0))

Outputs:

Calls: \*INTERSECTIONS-SEARCHED\*

>saf>cm>water-avoidance.lisp

\*QUADS-INDEX-LIST\*

>saf>cm>water-avoidance.lisp

THRU-RIVER-BEND

>saf>cm>water-avoidance.lisp

FOLLOW-WATER-SEGMENTS

>saf>cm>water-avoidance.lisp

FIND-WATER-INTERSECTIONS

>saf>cm>water-avoidance.lisp

FIND-SUITABLE-CROSSING-ROUTE

>saf>cm>water-avoidance.lisp

INTERSECTION-DIRECTION

>saf>cm>water-avoidance.lisp

FIND-CLOSER-CROSSING

>saf>cm>water-avoidance.lisp

CROSSING-LOCATION

>saf>cm>water-avoidance.lisp

SKIRT-LAKE

>saf>cm>water-avoidance.lisp

SKIRT-RIVER-BEND

>saf>cm>water-avoidance.lisp

GET-QUADS-IN-REGION

>saf>cm>water-avoidance.lisp

Called by: FIND-ROUTE-AROUND-WATER

>saf>cm>water-avoidance.lisp

Description: None

#### 2.2.4.1.6 FOLLOW-WATER-SEGMENTS

Definition 6

>saf>cm>water-avoidance.lisp

Type: Function

Arguments: (PAIR DIRECTION LEVEL SEGMENT-LIST OFFSET)



## Outputs:

Calls: \*WATER-SEGMENT-ARRAY\*  
       >map>terrain-vars.lisp  
       \*WATER-INTERSECTION-ARRAY\*  
       >map>terrain-vars.lisp  
       \*BRIDGE-ARRAY\*  
       >map>terrain-vars.lisp  
       \*QUAD-TREE\*  
       >map>terrain-vars.lisp  
       \*INTERSECTIONS-SEARCHED\*  
       >saf>cm>water-avoidance.lisp  
 FOLLOW-WATER-SEGMENTS  
       >saf>cm>water-avoidance.lisp  
 GET-PAIRS-BY-DIRECTION  
       >saf>cm>water-avoidance.lisp  
 Called by: FOLLOW-WATER-SEGMENTS  
           >saf>cm>water-avoidance.lisp  
           FIND-ROUTE-CORE  
           >saf>cm>water-avoidance.lisp  
 Description: None

**2.2.4.1.7 FIND-WATER-INTERSECTIONS**

## Definition 7

      >saf>cm>water-avoidance.lisp  
 Type: Function  
 Arguments: (WATER-INDEX)  
 Outputs:  
 Calls: \*WATER-INTERSECTION-ARRAY\*  
       >map>terrain-vars.lisp  
 Called by: FIND-ROUTE-CORE  
           >saf>cm>water-avoidance.lisp  
 Description: None

**2.2.4.1.8 GET-PAIRS-BY-DIRECTION**

## Definition 8

      >saf>cm>water-avoidance.lisp  
 Type: Function  
 Arguments: (INTERSECTION WATER-INDEX DIRECTION)  
 Outputs:  
 Calls: None  
 Called by: FOLLOW-WATER-SEGMENTS  
           >saf>cm>water-avoidance.lisp  
 Description: None

**2.2.4.1.9 FIND-SUITABLE-CROSSING-ROUTE**

Definition 9

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (START NEXT-CROSSING REST-CROSSINGS PREV-SEGMENT-LIST LAST-CROSSING DESTINATION DIRECTION OFFSET)

Outputs:

Calls: ALL-WIDE-SEGMENTS-THRU-WATER

&gt;saf&gt;cm&gt;water-check.lisp

\*QUADS-INDEX-LIST\*

&gt;saf&gt;cm&gt;water-avoidance.lisp

THRU-RIVER-BEND

&gt;saf&gt;cm&gt;water-avoidance.lisp

FIND-SUITABLE-CROSSING-ROUTE

&gt;saf&gt;cm&gt;water-avoidance.lisp

SET-XOR

&gt;saf&gt;cm&gt;water-avoidance.lisp

EXTEND-CROSSING

&gt;saf&gt;cm&gt;water-avoidance.lisp

SKIRT-RIVER

&gt;saf&gt;cm&gt;water-avoidance.lisp

SKIRT-LAKE

&gt;saf&gt;cm&gt;water-avoidance.lisp

SKIRT-RIVER-BEND

&gt;saf&gt;cm&gt;water-avoidance.lisp

Called by: FIND-SUITABLE-CROSSING-ROUTE

&gt;saf&gt;cm&gt;water-avoidance.lisp

FIND-ROUTE-CORE

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.10 SET-XOR**

Definition 10

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (LIST1 LIST2)

Outputs:

Calls: None

Called by: FIND-SUITABLE-CROSSING-ROUTE

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.11 EXTEND-CROSSING**

Definition 11

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (CROSSING DIRECTION &amp;OPTIONAL (OFFSET 100.0))

**Outputs:**

**Calls:** EXTEND-INTERSECTION  
>saf>cm>water-avoidance.lisp  
EXTEND-BRIDGE  
>saf>cm>water-avoidance.lisp  
EXTEND-SEGMENT  
>saf>cm>water-avoidance.lisp  
**Called by:** FIND-SUITABLE-CROSSING-ROUTE  
>saf>cm>water-avoidance.lisp  
**Description:** None

**2.2.4.1.12 EXTEND-INTERSECTION****Definition 12**

>saf>cm>water-avoidance.lisp  
**Type:** Function  
**Arguments:** (INTERSECTION DIRECTION &OPTIONAL (OFFSET 100.0))  
**Outputs:**  
**Calls:** \*WATER-SEGMENT-ARRAY\*  
>map>terrain-vars.lisp  
VEC-NORMALIZE  
>map>vectors.lisp  
VEC-ROTATE  
>map>vectors.lisp  
VEC-ADD  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
VEC-SCALE  
>map>vectors.lisp  
FIND-NEXT-POINT  
>saf>cm>water-avoidance.lisp  
**Called by:** EXTEND-CROSSING  
>saf>cm>water-avoidance.lisp  
**Description:** None

**2.2.4.1.13 FIRST-ITEMS****Definition 13**

>saf>cm>water-avoidance.lisp  
**Type:** Function  
**Arguments:** (LIST INDEX)  
**Outputs:**  
**Calls:** None  
**Called by:** EXTEND-BRIDGE  
>saf>cm>water-avoidance.lisp  
FIND-ROUTE-AROUND-WATER  
>saf>cm>water-avoidance.lisp  
**Description:** None

**2.2.4.1.14 EXTEND-BRIDGE**

Definition 14

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (BRIDGE SEGMENT-LIST DIRECTION &amp;OPTIONAL (OFFSET 200.0))

Outputs:

Calls: \*WATER-SEGMENT-ARRAY\*

&gt;map&gt;terrain-vars.lisp

VEC-NORMALIZE

&gt;map&gt;vectors.lisp

VEC-ADD

&gt;map&gt;vectors.lisp

VEC-SUB

&gt;map&gt;vectors.lisp

VEC-SCALE

&gt;map&gt;vectors.lisp

REVERSE-XY

&gt;saf&gt;cm&gt;control-measure.lisp

FIRST-ITEMS

&gt;saf&gt;cm&gt;water-avoidance.lisp

FIND-FIRST-VECTOR

&gt;saf&gt;cm&gt;water-avoidance.lisp

FIND-NEXT-POINT

&gt;saf&gt;cm&gt;water-avoidance.lisp

Called by: EXTEND-CROSSING

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.15 EXTEND-SEGMENT**

Definition 15

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (SEGMENT SEGMENT-LIST DIRECTION &amp;OPTIONAL (OFFSET 100.0))

Outputs:

Calls: PIE

&gt;map&gt;utilities.lisp

VEC-NORMALIZE

&gt;map&gt;vectors.lisp

VEC-ROTATE

&gt;map&gt;vectors.lisp

VEC-ADD

&gt;map&gt;vectors.lisp

VEC-SUB

&gt;map&gt;vectors.lisp

VEC-SCALE

&gt;map&gt;vectors.lisp

**FIND-FIRST-VECTOR**

&gt;saf&gt;cm&gt;water-avoidance.lisp

**FIND-SEGMENT-CROSS-POINTS**

&gt;saf&gt;cm&gt;water-avoidance.lisp

Called by: **EXTEND-CROSSING**

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.16 INTERSECTION-DIRECTION**

## Definition 16

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (ORIGIN WATER-LIST INT-INDEX)

Outputs:

Calls: **\*WATER-SEGMENT-ARRAY\***

&gt;map&gt;terrain-vars.lisp

**\*WATER-INTERSECTION-ARRAY\***

&gt;map&gt;terrain-vars.lisp

**REVERSE-XY**

&gt;saf&gt;cm&gt;control-measure.lisp

**FIND-DIRECTION-AT-CROSSING**

&gt;saf&gt;cm&gt;water-avoidance.lisp

Called by: **FIND-ROUTE-CORE**

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.17 NORMALIZE-AND-ROTATE**

## Definition 17

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (ORIGIN-VECTOR VECTOR-A VECTOR-B)

Outputs:

Calls: **VEC-NORMALIZE**

&gt;map&gt;vectors.lisp

**VEC-ROTATE**

&gt;map&gt;vectors.lisp

**VEC-SUB**

&gt;map&gt;vectors.lisp

Called by: **FIND-DIRECTION-AT-CROSSING**

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.18 FIND-FIRST-VECTOR**

## Definition 18

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (ORIGIN-VECTOR VECTOR-A VECTOR-B DIRECTION)

## Outputs:

Calls: VEC-ROTATE

&gt;map&gt;vectors.lisp

VECTOR-IS-FIRST-P

&gt;saf&gt;cm&gt;water-avoidance.lisp

Called by: OFFSET-POINT

&gt;saf&gt;cm&gt;water-avoidance.lisp

EXTEND-SEGMENT

&gt;saf&gt;cm&gt;water-avoidance.lisp

EXTEND-BRIDGE

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.19 VECTOR-IS-FIRST-P**

Definition 19

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (ANGLE1 ANGLE2 DIRECTION)

Outputs:

Calls: None

Called by: FIND-FIRST-VECTOR

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.20 FIND-NEXT-POINT**

Definition 20

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (XY POINTS)

Outputs:

Calls: None

Called by: EXTEND-BRIDGE

&gt;saf&gt;cm&gt;water-avoidance.lisp

EXTEND-INTERSECTION

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.21 FIND-SEGMENT-CROSS-POINTS**

Definition 21

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (SEGMENT PREV-SEGMENT-INDEX OFFSET)

Outputs:

Calls: \*WATER-SEGMENT-ARRAY\*

&gt;map&gt;terrain-vars.lisp

DISTANCE

&gt;map&gt;utilities.lisp

VEC-SUB

```

>map>vectors.lisp
VEC-SCALE
>map>vectors.lisp
LINE
>saf>cm>line.lisp
REVERSE-XY
>saf>cm>control-measure.lisp
LINE
>saf>cm>line.lisp
LINE
>saf>cm>line.lisp
Called by:  EXTEND-SEGMENT
>saf>cm>water-avoidance.lisp
Description:  None

```

#### 2.2.4.1.22 FIND-CLOSER-CROSSING

Definition 22

```

>saf>cm>water-avoidance.lisp
Type:  Function
Arguments:  (CROSSINGS-1 CROSSINGS-2 XY)
Outputs:
Calls:  DISTANCE
>map>utilities.lisp
CROSSING-LOCATION
>saf>cm>water-avoidance.lisp
Called by:  FIND-ROUTE-CORE
>saf>cm>water-avoidance.lisp
Description:  None

```

#### 2.2.4.1.23 SKIRT-RIVER

Definition 23

```

>saf>cm>water-avoidance.lisp
Type:  Function
Arguments:  (START DESTINATION CROSSING LAST-CROSSING SEGMENT-
LIST DIRECTION OFFSET)
Outputs:
Calls:  *WATER-SEGMENT-ARRAY*
>map>terrain-vars.lisp
FIND-RIVER-POINTS
>saf>cm>water-avoidance.lisp
OFFSET-POINTS
>saf>cm>water-avoidance.lisp
PRUNE-TO-POINT
>saf>cm>water-avoidance.lisp
RELAX-POINTS
>saf>cm>water-avoidance.lisp
FLAT-LIST-TO-POINTS
>saf>cm>water-avoidance.lisp

```

Called by: FIND-SUITABLE-CROSSING-ROUTE  
>saf>cm>water-avoidance.lisp  
Description: None

#### 2.2.4.1.24 FIND-RIVER-POINTS

Definition 24

>saf>cm>water-avoidance.lisp  
Type: Function  
Arguments: (SEGMENT-LIST CROSSING LAST-CROSSING)  
Outputs:  
Calls: \*WATER-SEGMENT-ARRAY\*  
>map>terrain-vars.lisp  
REVERSE-XY  
>saf>cm>control-measure.lisp  
ALIGN-POINTS  
>saf>cm>water-avoidance.lisp  
CROSSING-LOCATION  
>saf>cm>water-avoidance.lisp  
Called by: SKIRT-RIVER  
>saf>cm>water-avoidance.lisp  
Description: None

#### 2.2.4.1.25 ALIGN-POINTS

Definition 25

>saf>cm>water-avoidance.lisp  
Type: Function  
Arguments: (POINTS1 POINTS2 DIRECTION)  
Outputs:  
Calls: REVERSE-XY  
>saf>cm>control-measure.lisp  
Called by: CROSSING-LOCATION  
>saf>cm>water-avoidance.lisp  
FIND-RIVER-POINTS  
>saf>cm>water-avoidance.lisp  
Description: None

#### 2.2.4.1.26 OFFSET-POINTS

Definition 26

>saf>cm>water-avoidance.lisp  
Type: Function  
Arguments: (POINTS ORIGIN OFFSET DIRECTION)  
Outputs:  
Calls: VEC-ADD  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
OFFSET-POINT  
>saf>cm>water-avoidance.lisp



Called by: SKIRT-RIVER-BEND  
 >saf>cm>water-avoidance.lisp  
 FOLLOW-LAKE-AROUND  
 >saf>cm>water-avoidance.lisp  
 SKIRT-RIVER  
 >saf>cm>water-avoidance.lisp

Description: None

#### 2.2.4.1.27 OFFSET-POINT

Definition 27

>saf>cm>water-avoidance.lisp

Type: Function

Arguments: (P1 P2 P3 DIRECTION OFFSET)

Outputs:

Calls: PIE

>map>utilities.lisp  
 VEC-NORMALIZE  
 >map>vectors.lisp  
 VEC-ROTATE  
 >map>vectors.lisp  
 VEC-ADD  
 >map>vectors.lisp  
 VEC-SUB  
 >map>vectors.lisp  
 VEC-SCALE  
 >map>vectors.lisp  
 VEC-ANGLE  
 >map>vectors.lisp  
 FIND-FIRST-VECTOR  
 >saf>cm>water-avoidance.lisp

Called by: SKIRT-LAKE  
 >saf>cm>water-avoidance.lisp  
 OFFSET-POINTS  
 >saf>cm>water-avoidance.lisp

Description: None

#### 2.2.4.1.28 PRUNE-TO-POINT

Definition 28

>saf>cm>water-avoidance.lisp

Type: Function

Arguments: (POINTS ORIGIN DIRECTION)

Outputs:

Calls: DISTANCE

>map>utilities.lisp  
 POINT  
 >saf>interface>model-menu.lisp  
 POINT  
 >saf>interface>model-menu.lisp

Called by: SKIRT-RIVER-BEND  
 >saf>cm>water-avoidance.lisp  
 SKIRT-RIVER  
 >saf>cm>water-avoidance.lisp  
 Description: None

#### 2.2.4.1.29 CROSSING-LOCATION

Definition 29

>saf>cm>water-avoidance.lisp  
 Type: Function  
 Arguments: (CROSSING)  
 Outputs:  
 Calls: \*WATER-SEGMENT-ARRAY\*  
 >map>terrain-vars.lisp  
 REVERSE-XY  
 >saf>cm>control-measure.lisp  
 ALIGN-POINTS  
 >saf>cm>water-avoidance.lisp  
 Called by: FIND-RIVER-POINTS  
 >saf>cm>water-avoidance.lisp  
 FIND-CLOSER-CROSSING  
 >saf>cm>water-avoidance.lisp  
 FIND-ROUTE-CORE  
 >saf>cm>water-avoidance.lisp  
 Description: None

#### 2.2.4.1.30 RELAX-POINTS

Definition 30

>saf>cm>water-avoidance.lisp  
 Type: Function  
 Arguments: (START DESTINATION POINTS OFFSET)  
 Outputs:  
 Calls: \*QUADS-INDEX-LIST\*  
 >saf>cm>water-avoidance.lisp  
 RELAX-POINTS-AUX  
 >saf>cm>water-avoidance.lisp  
 GET-QUADS-IN-REGION  
 >saf>cm>water-avoidance.lisp  
 Called by: SKIRT-RIVER-BEND  
 >saf>cm>water-avoidance.lisp  
 FOLLOW-LAKE-AROUND  
 >saf>cm>water-avoidance.lisp  
 SKIRT-RIVER  
 >saf>cm>water-avoidance.lisp  
 Description: None

**2.2.4.1.31 RELAX-POINTS-AUX**

## Definition 31

>saf>cm>water-avoidance.lisp

Type: Function

Arguments: (START POINTS OFFSET)

Outputs:

Calls: POINT

>saf>interface>model-menu.lisp

ANY-WIDE-SEGMENT-THRU-WATER

>saf>cm>water-check.lisp

\*QUADS-INDEX-LIST\*

>saf>cm>water-avoidance.lisp

RELAX-POINTS-AUX

>saf>cm>water-avoidance.lisp

POINT

>saf>interface>model-menu.lisp

Called by: RELAX-POINTS-AUX

>saf>cm>water-avoidance.lisp

RELAX-POINTS

>saf>cm>water-avoidance.lisp

Description: None

**2.2.4.1.32 FINAL-RELAX-POINTS**

## Definition 32

>saf>cm>water-avoidance.lisp

Type: Function

Arguments: (START POINTS OFFSET)

Outputs:

Calls: POINT

>saf>interface>model-menu.lisp

ANY-WIDE-SEGMENT-THRU-WATER

>saf>cm>water-check.lisp

\*QUADS-INDEX-LIST\*

>saf>cm>water-avoidance.lisp

FINAL-RELAX-POINTS

>saf>cm>water-avoidance.lisp

POINT

>saf>interface>model-menu.lisp

Called by: FINAL-RELAX-POINTS

>saf>cm>water-avoidance.lisp

FIND-ROUTE-AROUND-WATER

>saf>cm>water-avoidance.lisp

Description: None

**2.2.4.1.33 FLAT-LIST-TO-POINTS**

Definition 33

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (POINTS)

Outputs:

Calls: None

Called by: SKIRT-RIVER-BEND

&gt;saf&gt;cm&gt;water-avoidance.lisp

SKIRT-LAKE

&gt;saf&gt;cm&gt;water-avoidance.lisp

SKIRT-RIVER

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.34 SKIRT-LAKE**

Definition 34

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (START LAKE-LIST NEXT-LIST OFFSET)

Outputs:

Calls: \*WATER-AREA-ARRAY\*

&gt;map&gt;terrain-vars.lisp

POINT

&gt;saf&gt;interface&gt;model-menu.lisp

OFFSET-POINT

&gt;saf&gt;cm&gt;water-avoidance.lisp

FLAT-LIST-TO-POINTS

&gt;saf&gt;cm&gt;water-avoidance.lisp

DISTANCE-AROUND-PATH

&gt;saf&gt;cm&gt;water-avoidance.lisp

FOLLOW-LAKE-AROUND

&gt;saf&gt;cm&gt;water-avoidance.lisp

POINT

&gt;saf&gt;interface&gt;model-menu.lisp

Called by: FIND-SUITABLE-CROSSING-ROUTE

&gt;saf&gt;cm&gt;water-avoidance.lisp

FIND-ROUTE-CORE

&gt;saf&gt;cm&gt;water-avoidance.lisp

Description: None

**2.2.4.1.35 DISTANCE-AROUND-PATH**

Definition 35

&gt;saf&gt;cm&gt;water-avoidance.lisp

Type: Function

Arguments: (PATH)

Outputs:

Calls: DISTANCE  
       >map>utilities.lisp  
 Called by: SKIRT-LAKE  
           >saf>cm>water-avoidance.lisp  
 Description: None

#### 2.2.4.1.36 FOLLOW-LAKE-AROUND

Definition 36

      >saf>cm>water-avoidance.lisp  
 Type: Function  
 Arguments: (START DESTINATION LAKE-POINTS MIN-START MIN-  
 DESTINATION INCREMENT OFFSET)  
 Outputs:  
 Calls: OFFSET-POINTS  
       >saf>cm>water-avoidance.lisp  
       RELAX-POINTS  
       >saf>cm>water-avoidance.lisp  
 Called by: SKIRT-LAKE  
           >saf>cm>water-avoidance.lisp  
 Description: None

#### 2.2.4.1.37 SKIRT-RIVER-BEND

Definition 37

      >saf>cm>water-avoidance.lisp  
 Type: Function  
 Arguments: (START DESTINATION WATER-LIST OFFSET)  
 Outputs:  
 Calls: \*WATER-SEGMENT-ARRAY\*  
       >map>terrain-vars.lisp  
       OFFSET-POINTS  
       >saf>cm>water-avoidance.lisp  
       PRUNE-TO-POINT  
       >saf>cm>water-avoidance.lisp  
       RELAX-POINTS  
       >saf>cm>water-avoidance.lisp  
       FLAT-LIST-TO-POINTS  
       >saf>cm>water-avoidance.lisp  
       FIND-RIVER-BEND-POINTS  
       >saf>cm>water-avoidance.lisp  
       FIND-DIRECTION-AT-CROSSING  
       >saf>cm>water-avoidance.lisp  
 Called by: FIND-SUITABLE-CROSSING-ROUTE  
           >saf>cm>water-avoidance.lisp  
           FIND-ROUTE-CORE  
           >saf>cm>water-avoidance.lisp  
 Description: None

**2.2.4.1.38 FIND-RIVER-BEND-POINTS**

Definition 38

>saf>cm>water-avoidance.lisp  
Type: Function  
Arguments: (FIRST-CROSSING LAST-CROSSING)  
Outputs:  
Calls: \*WATER-SEGMENT-ARRAY\*  
>map>terrain-vars.lisp  
REVERSE-XY  
>saf>cm>control-measure.lisp  
Called by: SKIRT-RIVER-BEND  
>saf>cm>water-avoidance.lisp  
Description: None

**2.2.4.1.39 FIND-DIRECTION-AT-CROSSING**

Definition 39

>saf>cm>water-avoidance.lisp  
Type: Function  
Arguments: (WATER-POINTS ORIGIN WATER-LIST POINT-DIRECTION)  
Outputs:  
Calls: VEC-SUB  
>map>vectors.lisp  
NORMALIZE-AND-ROTATE  
>saf>cm>water-avoidance.lisp  
Called by: SKIRT-RIVER-BEND  
>saf>cm>water-avoidance.lisp  
INTERSECTION-DIRECTION  
>saf>cm>water-avoidance.lisp  
Description: None

**2.2.4.1.40 GET-QUADS-IN-REGION**

Definition 40

>saf>cm>water-avoidance.lisp  
Type: Function  
Arguments: (POINT-LIST)  
Outputs:  
Calls: \*QUAD-TREE\*  
>map>terrain-vars.lisp  
GET-QUADS-PASSED-THRU  
>saf>cm>water-check.lisp  
Called by: RELAX-POINTS  
>saf>cm>water-avoidance.lisp  
FIND-ROUTE-CORE  
>saf>cm>water-avoidance.lisp  
FIND-ROUTE-AROUND-WATER  
>saf>cm>water-avoidance.lisp  
Description: None

### 2.2.4.2 CSU cm>water-check.lisp

This unit contains routines to check if ground routes cross water, which can be either rivers or lakes. These functions are called by the water-avoidance code in CSU cm>water-avoidance.lisp. Routines are provided that check if routes cross any water or return an ordered list of the water features that are crossed.

The function any-wide-segment-thru-water checks to see if a segment passes within a given clearance of water. This is done by constructing parallel segments on either side and checking to see if they cross water. This check is accomplished by a call to segment-thru-water.

The function segment-thru-water calls get-quads-passed-thru to obtain a lists of the rivers and lakes in the quads the segment passes through. This list is then used in calls to segment-thru-river and segment-thru-lake.

The function segment-thru-river loops through all the water segments in the rivers list returned by get-quads-passed-thru, checking to see if the water-segment intersects the segment from (x1, y1) to (x2, y2). It returns as soon as it finds the first intersection; fordable water segments are ignored.

The function segment-thru-lake checks a segment for lake intersections. This task is made complex because a lake can contain islands, which may contain lakes, which themselves may have islands, etc. To deal with this, lake/island representations are stored in a recursive data-structure where the first element is the outer boundary of the lake or island, and the second element is a list of the lake/island data-structures for the next level of boundaries that lie inside the outer one. If the level zero boundary is a lake, even level boundaries are the outer boundaries of lakes and odd level boundaries are the outer boundaries of islands.

Using this data-structure, checking for lake intersections is reduced to checking for intersections with the polygonal boundaries at various levels. In segment-thru-lake, a call is made to check-lake-intersections. This function calls itself recursively, incrementing the depth level in the lake/island data-structure. It also calls polygon-intersection to check if the segment crosses the polygonal lake/island boundaries.

The function all-wide-segments-thru-water returns a list of all water crossings, unlike any-wide-segment-thru-water, which only returns the first intersection it finds. This is done via a call to water-thru, which, like segment-thru-water, calls get-quads-passed-thru, then makes separate calls to check rivers and lakes. The river check is accomplished by the function water-segments-thru, the lake check by the function lake-thru. The final list of water-crossings is sorted by the distance from a designated endpoint of the route segment; the sort is useful to the water-avoidance algorithm.

#### 2.2.4.2.1 ANY-WIDE-SEGMENT-THRU-WATER

Definition 1

>saf>cm>water-check.lisp

Type: Function

Arguments: (XY1 XY2 OFFSET &OPTIONAL INDEX-LIST (QUAD-TREE \*QUAD-TREE\*))

## Outputs:

Calls: \*QUAD-TREE\*  
       >map>terrain-vars.lisp  
       PIE  
       >map>utilities.lisp  
       VEC-NORMALIZE  
       >map>vectors.lisp  
       VEC-ROTATE  
       >map>vectors.lisp  
       VEC-ADD  
       >map>vectors.lisp  
       VEC-SUB  
       >map>vectors.lisp  
       VEC-SCALE  
       >map>vectors.lisp  
       SEGMENT-THRU-WATER  
       >saf>cm>water-check.lisp  
 Called by: FINAL-RELAX-POINTS  
           >saf>cm>water-avoidance.lisp  
           RELAX-POINTS-AUX  
           >saf>cm>water-avoidance.lisp  
 Description: None

**2.2.4.2.2 SEGMENT-THRU-WATER**

## Definition 2

      >saf>cm>water-check.lisp  
 Type: Function  
 Arguments: (XY1 XY2 &OPTIONAL INDEX-LIST (QUAD-TREE \*QUAD-TREE\*))  
 Outputs:  
 Calls: \*QUAD-TREE\*  
       >map>terrain-vars.lisp  
       SEGMENT-THRU-RIVER  
       >saf>cm>water-check.lisp  
       SEGMENT-THRU-LAKE  
       >saf>cm>water-check.lisp  
       GET-QUADS-PASSED-THRU  
       >saf>cm>water-check.lisp  
 Called by: (METHOD CHECK-ROUTE-SEGMENT ROUTE)  
           >saf>cm>route.lisp  
           ANY-WIDE-SEGMENT-THRU-WATER  
           >saf>cm>water-check.lisp  
           GET-BRIDGE-POINTS  
           >saf>cm>road-routes.lisp  
 Description: None

**2.2.4.2.3 SEGMENT-THRU-RIVER**

## Definition 3

      >saf>cm>water-check.lisp  
 Type: Function  
 Arguments: (X1 Y1 X2 Y2 INDEX-LIST)



## Outputs:

Calls: \*WATER-SEGMENT-ARRAY\*

&gt;map&gt;terrain-vars.lisp

POSSIBLE-INTERSECTION

&gt;map&gt;intersection.lisp

Called by: SEGMENT-THRU-WATER

&gt;saf&gt;cm&gt;water-check.lisp

Description: None

**2.2.4.2.4 \*INSIDE-LEVEL\***

Definition 4

&gt;saf&gt;cm&gt;water-check.lisp

Type: Variable

Arguments: ()

Outputs:

Calls: None

Called by: CHECK-LAKE-INTERSECTIONS

&gt;saf&gt;cm&gt;water-check.lisp

SEGMENT-THRU-LAKE

&gt;saf&gt;cm&gt;water-check.lisp

Description: None

**2.2.4.2.5 SEGMENT-THRU-LAKE**

Definition 5

&gt;saf&gt;cm&gt;water-check.lisp

Type: Function

Arguments: (X1 Y1 X2 Y2 INDEX-LIST)

Outputs:

Calls: \*WATER-AREA-ARRAY\*

&gt;map&gt;terrain-vars.lisp

\*WATER-AREA-TRIANGLES\*

&gt;map&gt;terrain-vars.lisp

\*INSIDE-LEVEL\*

&gt;saf&gt;cm&gt;water-check.lisp

CHECK-LAKE-INTERSECTIONS

&gt;saf&gt;cm&gt;water-check.lisp

Called by: SEGMENT-THRU-WATER

&gt;saf&gt;cm&gt;water-check.lisp

Description: None

**2.2.4.2.6 POLYGON-INTERSECTION**

Definition 6

&gt;saf&gt;cm&gt;water-check.lisp

Type: Function

Arguments: (POLYGON X3 Y3 X4 Y4)

Outputs:

Calls: None

Called by: CHECK-LAKE-INTERSECTIONS  
 >saf>cm>water-check.lisp  
 Description: None

#### 2.2.4.2.7 CHECK-LAKE-INTERSECTIONS

Definition 7

>saf>cm>water-check.lisp  
 Type: Function  
 Arguments: (LAKE X1 Y1 X2 Y2 LEVEL)  
 Outputs:  
 Calls: SEGMENT-INSIDE-POLYGON-P  
 >map>intersection.lisp  
 \*INSIDE-LEVEL\*  
 >saf>cm>water-check.lisp  
 POLYGON-INTERSECTION  
 >saf>cm>water-check.lisp  
 CHECK-LAKE-INTERSECTIONS  
 >saf>cm>water-check.lisp  
 Called by: CHECK-LAKE-INTERSECTIONS  
 >saf>cm>water-check.lisp  
 SEGMENT-THRU-LAKE  
 >saf>cm>water-check.lisp  
 Description: None

#### 2.2.4.2.8 ALL-WIDE-SEGMENTS-THRU-WATER

Definition 8

>saf>cm>water-check.lisp  
 Type: Function  
 Arguments: (XY1 XY2 OFFSET &OPTIONAL INDEX-LIST (QUAD-TREE \*QUAD-TREE\*))  
 Outputs:  
 Calls: \*QUAD-TREE\*  
 >map>terrain-vars.lisp  
 PIE  
 >map>utilities.lisp  
 VEC-NORMALIZE  
 >map>vectors.lisp  
 VEC-ROTATE  
 >map>vectors.lisp  
 VEC-ADD  
 >map>vectors.lisp  
 VEC-SUB  
 >map>vectors.lisp  
 VEC-SCALE  
 >map>vectors.lisp  
 WATER-THRU  
 >saf>cm>water-check.lisp

Called by: FIND-SUITABLE-CROSSING-ROUTE

>saf>cm>water-avoidance.lisp  
FIND-ROUTE-AROUND-WATER  
>saf>cm>water-avoidance.lisp

Description: None

#### 2.2.4.2.9 WATER-THRU

Definition 9

>saf>cm>water-check.lisp

Type: Function

Arguments: (XY1 XY2 INDEX-LIST &OPTIONAL (QUAD-TREE \*QUAD-TREE\*))

Outputs:

Calls: \*QUAD-TREE\*

>map>terrain-vars.lisp

DISTANCE

>map>utilities.lisp

WATER-SEGMENTS-THRU

>saf>cm>water-check.lisp

LAKES-THRU

>saf>cm>water-check.lisp

GET-QUADS-PASSED-THRU

>saf>cm>water-check.lisp

Called by: ALL-WIDE-SEGMENTS-THRU-WATER

>saf>cm>water-check.lisp

Description: None

#### 2.2.4.2.10 WATER-SEGMENTS-THRU

Definition 10

>saf>cm>water-check.lisp

Type: Function

Arguments: (X1 Y1 X2 Y2 INDEX-LIST)

Outputs:

Calls: \*WATER-SEGMENT-ARRAY\*

>map>terrain-vars.lisp

POSSIBLE-INTERSECTION

>map>intersection.lisp

Called by: WATER-THRU

>saf>cm>water-check.lisp

Description: None

#### 2.2.4.2.11 LAKES-THRU

Definition 11

>saf>cm>water-check.lisp

Type: Function

Arguments: (X3 Y3 X4 Y4 INDEX-LIST)

## Outputs:

Calls: \*WATER-AREA-ARRAY\*  
       >map>terrain-vars.lisp  
       \*WATER-AREA-TRIANGLES\*  
       >map>terrain-vars.lisp  
       SEGMENT-INSIDE-POLYGON-P  
       >map>intersection.lisp  
       POSSIBLE-INTERSECTION  
       >map>intersection.lisp  
 Called by: WATER-THRU  
           >saf>cm>water-check.lisp  
 Description: None

**2.2.4.2.12 GET-QUADS-PASSED-THRU**

## Definition 12

      >saf>cm>water-check.lisp  
 Type: Function  
 Arguments: (X1 Y1 X2 Y2 QUAD-TREE)  
 Outputs:  
 Calls: GET-QUAD-NODES  
       >map>quadtree-search.lisp  
 Called by: GET-QUADS-IN-REGION  
           >saf>cm>water-avoidance.lisp  
           WATER-THRU  
           >saf>cm>water-check.lisp  
           SEGMENT-THRU-WATER  
           >saf>cm>water-check.lisp  
 Description: None

**2.2.4.3 CSU cm>route-point.lisp**

This unit contains the definition of the route points, which are specialized control measure points. The first slot in the route-point object is a boolean variable called road-point. It is true if the route point is intended to be part of a road route, and nil otherwise. Cross country routes will be made of points with this slot set to nil; road routes will be made of points with this slot set to true. A route point with the road-point slot true can be called a road-point, even though there is no separate object class for these kind of route-points. The next three slots are only relevant if the route-point is a road point. The intersection-index slot is non-nil if the point is intended to be at a road-intersection. In this case, its value is simply the index, in the road intersections array, of that road intersection. Other road-points are intended to be points in the interior of a road segment (a portion of road connecting two road-intersections). These will have the third slot, road-index, set to the non-nil value of the index, in the road-segments array, of that road segment. It is worth noting that a dead-end of a road segment is considered a road intersection, so every road segment has two ends, which are road intersections. (Roads that run off the edge of the map end in an intersection; circular roads start and end at the same intersection.) The fourth slot, expanded-route, is used to store a list of road-indices and traversal directions that describes a road path from a previous road-point to the given road-point. This slot is convenient because many of the road route algorithms construct routes as a series of paths from one road-point to another; the expanded-route slot provides a convenient place to put the description of the path.

**2.2.4.3.1 ROUTE-POINT**

Definition 1

&gt;saf&gt;cm&gt;route-point.lisp

Type: DEFOBJECT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.4.3.2 (METHOD COPY ROUTE-POINT)**

Definition 2

&gt;saf&gt;cm&gt;route-point.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.4.3.3 ROUTE-POINT**

Definition 3

&gt;saf&gt;cm&gt;route-point.lisp

Type: COMPILE-FLAVOR-METHODS

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.4.3.4 XY-LIST-TO-ROUTE-POINTS**

Definition 4

&gt;saf&gt;cm&gt;route-point.lisp

Type: Function

Arguments: (XY-LIST)

Outputs:

Calls: None

Called by: MAKE-ROUTE

&gt;saf&gt;cm&gt;route.lisp

Description: None

#### 2.2.4.4 CSU `cm>road-routes.lisp`

This unit contains the routines for entering road routes, as well as routines for expanding the road routes so they can be drawn on the color display.

The top-level function in this unit, `get-road-route`, is called by the function `make-route`, in CSU `cm>route.lisp`, when the user selects "Road Route" from the route menu. It allows the user to create a road route by clicking on a series of points. As each new point is selected, a road route from the last point to the new point is constructed and drawn on the PVD. Mouse documentation presented by `get-road-route` allows the user to select a new road-point, remove the last road-point, or exit from the road-point selection mode. When mouse-left is selected, a new road-point is created, by the function `get-road-point`, at or near the current mouse location. Then, the function `expand-route` is called to create a road route from the previous road-point to the newly selected road-point. This expanded route is then drawn on the PVD by a call to the function `draw-expanded-route`, with the overlay-`alu`. When mouse-right is selected, the last road-point is deleted, and the previously constructed road-route leading to the last point is erased by a call to `draw-expanded-route`, this time using the `erase-overlay-alu`.

The function `get-road-point` takes a point (specified by a coordinate pair from a mouse selection) and returns a road-point located at the nearest road intersection or road segment to the selected point. First, it calls the function `find-nearest-intersection`, to find the closest intersection to the selected point. If this intersection is "within the cursor" (within 10 pixels of the selected point, as determined by the function `within-cursor`), the intersection is chosen as the road-point. Otherwise, the function `find-nearest-road-segment` is called to find the closest segment to the selected point; the closest point on this segment is chosen as the road-point.

The function `find-nearest-road-segment` calls `get-neighbor-quad-roads` to explore nearby quads for road segments. It first tries a fast check for road segments that pass very close to the selected point, within 8 pixels. If it is unable to find a road segment of this kind, it does a slower, more systematic check of road segments, calling `parallel-distance` to compute the distance from the selected point to the closest point on each road segment found in the nearby quads.

The function `expand-route` calls `expand-road-route` to find an optimal route from one road-point to another. The function `expand-road-route` constructs, for each road-point, a list of intersections. If the road-point represents a road intersection (that is, if its `intersection-index` slot is non-`nil`), the intersection list constructed will be of length one, consisting of the intersection index of the road-point. If the road-point represents a road-segment (that is, if its `road-index` slot is non-`nil`), the intersection list constructed will be of length two, consisting of the two intersection indices of the two ends of the road segment of the given road-point, as computed by the utility function `find-road-intersections`. Once the two intersection lists (`intersections-1` and `intersections-2`) are constructed, `expand-road-route` makes four calls to the function `find-route`, in CSU `cm>route-finder.lisp`, to find the shortest routes between various pairs of road intersections. Since each intersection list may have up to two members, there are four possible combinations to test. Of these four, the shortest is selected, by the function `find-shortest-route`, and returned. If `find-route` was unable to find any route, the user is advised to choose an intermediate point, to make the connection process easier.

The function `draw-expanded-route`, called by `get-road-route`, draws the road route to a newly selected road-point. It can also be used to erase, by passing it an erasing `alu`. It begins by drawing a distinguishing box around the road-point, with corners at offsets 4

pixels from the road-point. It then calls draw-expanded-route-core to draw the actual road route.

The function draw-expanded-route-core draws the road-route from one road point (the argument prev-point) to the next (the argument road-point). Road segments, which end at intersections, are made up of shorter linear parts called "legs". A road segment which is fully traversed by the route will have all its legs drawn, while a road segment which contains the start and/or finish road-points may not be fully traversed by the route. For example, if a route starts somewhere near the middle of an east-west road segment, and runs east from there, the only legs in that road-segment that are part of the route are those east of the starting point. For this reason, draw-expanded-route-core is divided into two cases according to whether the start and end points are on the same road segment. (They may even be on the same leg as well.) In each case, road segments that are not fully traversed are handled separately from those that are fully traversed.

Finally, this unit includes functions that handle bridge crossings. In the road-segment array, a bridge is represented as a short road-segment. A boolean slot identifies those road-segments that are bridges. There is also a separate array, called \*bridge-array\*, containing only the bridges; it is used when a fast scan of bridges is required.

The function get-bridge-route allows the user to include bridges in a cross-country ground route. After the user clicks near the bridge they want to cross, the function calls find-nearest-bridge, which scans the road-indices returned by get-neighbor-quad-roads, locating the intended bridge. Then, get-bridge-points is called. Its purpose is to find points near the ends of the bridge, but far enough away to give groups of vehicles time to change to column formation before crossing the bridge. It allows 200 meters for this. The direction of traversal of the bridge is inferred using water checks, on the assumption that crossing it the right way will create a water-avoiding route from the previous point. If there is no previous point, get-bridge-points calls mouse-on-bridge-approach-point to get explicit guidance from the user about which side of the river to approach the bridge from.

#### 2.2.4.4.1 GET-ROAD-ROUTE

Definition 1

```

>saf>cm>road-routes.lisp
Type: Function
Arguments: (STREAM)
Outputs:
Calls: WITH-INTEGGER-CONVERSION-MODE
       >map>utilities.lisp
       WITH-MAP-GRAPHICS
       >map>utilities.lisp
       SCREEN-TO-WORLD
       >map>utilities.lisp
       *OVERLAY-ALU*
       >map>color-map.lisp
       *ERASE-OVERLAY-ALU*
       >map>color-map.lisp
       *PVD-DISPLAY*
       >saf>sys>vars.lisp

```

GET-ROAD-POINT  
 >saf>cm>road-routes.lisp  
 EXPAND-ROUTE  
 >saf>cm>road-routes.lisp  
 DRAW-EXPANDED-ROUTE  
 >saf>cm>road-routes.lisp  
 Called by: MAKE-ROUTE  
 >saf>cm>route.lisp  
 Description: None

#### 2.2.4.4.2 GET-ROAD-POINT

Definition 2

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (X Y &OPTIONAL POINT)  
 Outputs:  
 Calls: \*ROAD-INTERSECTION-ARRAY\*  
 >map>terrain-vars.lisp  
 POINT  
 >saf>interface>model-menu.lisp  
 FIND-NEAREST-INTERSECTION  
 >saf>cm>road-routes.lisp  
 FIND-NEAREST-ROAD-SEGMENT  
 >saf>cm>road-routes.lisp  
 WITHIN-CURSOR  
 >saf>cm>road-routes.lisp  
 POINT  
 >saf>interface>model-menu.lisp  
 Called by: (METHOD MOVE-POINT ROUTE)  
 >saf>cm>route.lisp  
 GET-ROAD-ROUTE  
 >saf>cm>road-routes.lisp  
 Description: None

#### 2.2.4.4.3 GET-ROAD-SEGMENT-POINT

Definition 3

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (X Y &OPTIONAL POINT)  
 Outputs:  
 Calls: POINT  
 >saf>interface>model-menu.lisp  
 FIND-NEAREST-ROAD-SEGMENT  
 >saf>cm>road-routes.lisp  
 POINT  
 >saf>interface>model-menu.lisp



Called by: (METHOD INSERT-POINT-BEFORE ROUTE)  
 >saf>cm>route.lisp  
 (METHOD INSERT-POINT-AFTER ROUTE)  
 >saf>cm>route.lisp

Description: None

#### 2.2.4.4.4 FIND-NEAREST-INTERSECTION

Definition 4

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (X-WC Y-WC &OPTIONAL (QUAD-TREE \*QUAD-TREE\*))  
 Outputs:  
 Calls: \*ROAD-INTERSECTION-ARRAY\*  
 >map>terrain-vars.lisp  
 \*QUAD-TREE\*  
 >map>terrain-vars.lisp  
 DISTANCE  
 >map>utilities.lisp  
 GET-NEIGHBOR-QUAD-ROADS  
 >saf>cm>road-routes.lisp  
 Called by: GET-ROAD-POINT  
 >saf>cm>road-routes.lisp  
 Description: None

#### 2.2.4.4.5 FIND-NEAREST-ROAD-SEGMENT

Definition 5

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (X-WC Y-WC &OPTIONAL (QUAD-TREE \*QUAD-TREE\*))  
 Outputs:  
 Calls: \*ROAD-SEGMENT-ARRAY\*  
 >map>terrain-vars.lisp  
 \*QUAD-TREE\*  
 >map>terrain-vars.lisp  
 DISTANCE  
 >map>utilities.lisp  
 CLIP  
 >map>clip.lisp  
 \*PVD-DISPLAY\*  
 >saf>sys>vars.lisp  
 GET-NEIGHBOR-QUAD-ROADS  
 >saf>cm>road-routes.lisp  
 PARALLEL-DISTANCE  
 >saf>cm>road-routes.lisp  
 ROUTE-INTERSECTION  
 >saf>cm>road-routes.lisp

Called by: GET-ROAD-SEGMENT-POINT

>saf>cm>road-routes.lisp

GET-ROAD-POINT

>saf>cm>road-routes.lisp

Description: None

#### 2.2.4.4.6 GET-NEIGHBOR-QUAD-ROADS

Definition 6

>saf>cm>road-routes.lisp

Type: Function

Arguments: (X-WC Y-WC &OPTIONAL (INTERSECTIONS NIL) (QUAD-TREE \*QUAD-TREE\*))

Outputs:

Calls: \*QUAD-TREE\*

>map>terrain-vars.lisp

GET-QUAD-NODES

>map>quadtree-search.lisp

Called by: FIND-NEAREST-BRIDGE

>saf>cm>road-routes.lisp

FIND-NEAREST-ROAD-SEGMENT

>saf>cm>road-routes.lisp

FIND-NEAREST-INTERSECTION

>saf>cm>road-routes.lisp

Description: None

#### 2.2.4.4.7 CALCULATE-POINT-LINE-INTERSECTION

Definition 7

>saf>cm>road-routes.lisp

Type: Function

Arguments: (X1 Y1 X2 Y2 X Y)

Outputs:

Calls: NEAR

>map>utilities.lisp

POINT-LINE-INTERSECTION

>map>intersection.lisp

Called by: ROUTE-INTERSECTION

>saf>cm>road-routes.lisp

PARALLEL-DISTANCE

>saf>cm>road-routes.lisp

Description: None

#### 2.2.4.4.8 PARALLEL-DISTANCE

Definition 8

>saf>cm>road-routes.lisp

Type: Function

Arguments: (X1 Y1 X2 Y2 X Y)

Outputs:

**Calls:** DISTANCE  
>map>utilities.lisp  
CALCULATE-POINT-LINE-INTERSECTION  
>saf>cm>road-routes.lisp  
**Called by:** FIND-NEAREST-ROAD-SEGMENT  
>saf>cm>road-routes.lisp  
**Description:** None

#### 2.2.4.4.9 ROUTE-INTERSECTION

Definition 9

>saf>cm>road-routes.lisp  
**Type:** Function  
**Arguments:** (SEGMENT INDEX X Y)  
**Outputs:**  
**Calls:** \*ROAD-SEGMENT-ARRAY\*  
>map>terrain-vars.lisp  
DISTANCE  
>map>utilities.lisp  
CALCULATE-POINT-LINE-INTERSECTION  
>saf>cm>road-routes.lisp  
**Called by:** FIND-NEAREST-ROAD-SEGMENT  
>saf>cm>road-routes.lisp  
**Description:** None

#### 2.2.4.4.10 WITHIN-CURSOR

Definition 10

>saf>cm>road-routes.lisp  
**Type:** Function  
**Arguments:** (DISTANCE)  
**Outputs:**  
**Calls:** \*PVD-DISPLAY\*  
>saf>sys>vars.lisp  
**Called by:** GET-ROAD-POINT  
>saf>cm>road-routes.lisp  
**Description:** None

#### 2.2.4.4.11 EXPAND-ROUTE

Definition 11

>saf>cm>road-routes.lisp  
**Type:** Function  
**Arguments:** (PREV-ITEM ITEM &OPTIONAL FORCE)  
**Outputs:**  
**Calls:** EXPAND-ROAD-ROUTE  
>saf>cm>road-routes.lisp  
**Called by:** (METHOD INSERT-POINT-BEFORE ROUTE)  
>saf>cm>route.lisp  
(METHOD INSERT-POINT-AFTER ROUTE)  
>saf>cm>route.lisp  
(METHOD DELETE-POINT ROUTE)

>saf>cm>route.lisp  
 (METHOD MOVE-POINT ROUTE)

>saf>cm>route.lisp  
 GET-BRIDGE-POINTS  
 >saf>cm>road-routes.lisp  
 GET-ROAD-ROUTE

>saf>cm>road-routes.lisp

Description: None

#### 2.2.4.4.12 EXPAND-ROAD-ROUTE

Definition 12

>saf>cm>road-routes.lisp

Type: Function

Arguments: (PREV-ITEM ITEM)

Outputs:

Calls: \*OPFOR-IO\*

>saf>sys>vars.lisp

SAY

>saf>sys>macros.lisp

ROUTE

>saf>cm>route.lisp

FIND-ROUTE

>saf>cm>route-finder.lisp

FIND-ROAD-INTERSECTIONS

>saf>cm>road-routes.lisp

FIND-SHORTEST-ROUTE

>saf>cm>road-routes.lisp

FIND-ROAD-DIRECTION

>saf>cm>road-routes.lisp

ROAD-SEGMENTS-FROM-INTERSECTIONS

>saf>cm>road-routes.lisp

ROUTE

>saf>cm>route.lisp

ROUTE

>saf>cm>route.lisp

Called by: DRAW-EXPANDED-ROUTE-CORE

>saf>cm>road-routes.lisp

EXPAND-ROUTE

>saf>cm>road-routes.lisp

Description: None

#### 2.2.4.4.13 FIND-ROAD-INTERSECTIONS

Definition 13

>saf>cm>road-routes.lisp

Type: Function

Arguments: (ROAD-INDEX)

Outputs:

Calls: \*ROAD-INTERSECTION-ARRAY\*

>map>terrain-vars.lisp

Called by: EXPAND-ROAD-ROUTE

>saf>cm>road-routes.lisp

Description: None

**2.2.4.4.14 FIND-SHORTEST-ROUTE**

## Definition 14

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (&REST ROUTES)  
 Outputs:  
 Calls: ROUTE  
       >saf>cm>route.lisp  
       CALCULATE-ROUTE-DISTANCE  
       >saf>cm>road-routes.lisp  
       ROUTE  
       >saf>cm>route.lisp  
       ROUTE  
       >saf>cm>route.lisp  
 Called by: EXPAND-ROAD-ROUTE  
           >saf>cm>road-routes.lisp  
 Description: None

**2.2.4.4.15 CALCULATE-ROUTE-DISTANCE**

## Definition 15

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (INTERSECTION-LIST)  
 Outputs:  
 Calls: \*ROAD-SEGMENT-ARRAY\*  
       >map>terrain-vars.lisp  
       \*ROAD-INTERSECTION-ARRAY\*  
       >map>terrain-vars.lisp  
       CALCULATE-ROUTE-DISTANCE  
       >saf>cm>road-routes.lisp  
 Called by: CALCULATE-ROUTE-DISTANCE  
           >saf>cm>road-routes.lisp  
           FIND-SHORTEST-ROUTE  
           >saf>cm>road-routes.lisp  
 Description: None

**2.2.4.4.16 FIND-ROAD-DIRECTION**

## Definition 16

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (PREV-ITEM ITEM)  
 Outputs:  
 Calls: \*ROAD-SEGMENT-ARRAY\*  
       >map>terrain-vars.lisp  
 Called by: EXPAND-ROAD-ROUTE  
           >saf>cm>road-routes.lisp  
 Description: None

**2.2.4.4.17 ROAD-SEGMENTS-FROM-INTERSECTIONS**

## Definition 17

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (PREV-ITEM ITEM ROUTE)  
 Outputs:  
 Calls: \*ROAD-SEGMENT-ARRAY\*  
       >map>terrain-vars.lisp  
       \*ROAD-INTERSECTION-ARRAY\*  
       >map>terrain-vars.lisp  
       ROUTE  
       >saf>cm>route.lisp  
       ROUTE  
       >saf>cm>route.lisp  
       ROUTE  
       >saf>cm>route.lisp  
 Called by: EXPAND-ROAD-ROUTE  
           >saf>cm>road-routes.lisp  
 Description: None

**2.2.4.4.18 DRAW-EXPANDED-ROUTE**

## Definition 18

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (PREV-POINT ROAD-POINT STREAM ALU)  
 Outputs:  
 Calls: WITH-INTEGGER-CONVERSION-MODE  
       >map>utilities.lisp  
       WITH-MAP-GRAPHICS  
       >map>utilities.lisp  
       WITH-FAST-MAP-GRAPHICS  
       >map>utilities.lisp  
       DRAW-EXPANDED-ROUTE-CORE  
       >saf>cm>road-routes.lisp  
 Called by: GET-ROAD-ROUTE  
           >saf>cm>road-routes.lisp  
 Description: None

**2.2.4.4.19 DRAW-EXPANDED-ROUTE-CORE**

## Definition 19

>saf>cm>road-routes.lisp  
 Type: Function  
 Arguments: (PREV-POINT ROAD-POINT STREAM ALU)  
 Outputs:

Calls: **\*ROAD-SEGMENT-ARRAY\***

>map>terrain-vars.lisp  
 REVERSE-XY  
 >saf>cm>control-measure.lisp  
 EXPAND-ROAD-ROUTE  
 >saf>cm>road-routes.lisp

Called by: **MAKE-ROUTE**

>saf>cm>route.lisp  
 (METHOD INSERT-POINT-BEFORE ROUTE)  
 >saf>cm>route.lisp  
 (METHOD INSERT-POINT-AFTER ROUTE)  
 >saf>cm>route.lisp  
 (METHOD DELETE-POINT ROUTE)  
 >saf>cm>route.lisp  
 (METHOD MOVE-POINT ROUTE)  
 >saf>cm>route.lisp  
 (METHOD PAINT ROUTE)  
 >saf>cm>route.lisp  
 DRAW-EXPANDED-ROUTE  
 >saf>cm>road-routes.lisp

Description: None

#### 2.2.4.4.20 GET-BRIDGE-ROUTE

Definition 20

>saf>cm>road-routes.lisp

Type: Function

Arguments: (PREV-POINT STREAM)

Outputs:

Calls: **WITH-INTEGER-CONVERSION-MODE**

>map>utilities.lisp  
 WITH-MAP-GRAPHICS  
 >map>utilities.lisp  
 SCREEN-TO-WORLD  
 >map>utilities.lisp  
 \*OPFOR-IO\*  
 >saf>sys>vars.lisp  
 SAY  
 >saf>sys>macros.lisp  
 GET-BRIDGE-ROUTE  
 >saf>cm>road-routes.lisp  
 GET-BRIDGE-POINTS  
 >saf>cm>road-routes.lisp  
 FIND-NEAREST-BRIDGE  
 >saf>cm>road-routes.lisp

Called by: **MAKE-ROUTE**

>saf>cm>route.lisp  
 GET-BRIDGE-ROUTE  
 >saf>cm>road-routes.lisp

Description: None

**2.2.4.4.21 GET-BRIDGE-POINTS**

Definition 21

&gt;saf&gt;cm&gt;road-routes.lisp

Type: Function

Arguments: (BRIDGE PREV-POINT STREAM)

Outputs:

Calls: \*ROAD-SEGMENT-ARRAY\*

&gt;map&gt;terrain-vars.lisp

\*ROAD-INTERSECTION-ARRAY\*

&gt;map&gt;terrain-vars.lisp

DISTANCE

&gt;map&gt;utilities.lisp

FIND-INTER-POINT

&gt;map&gt;vectors.lisp

REVERSE-XY

&gt;saf&gt;cm&gt;control-measure.lisp

EXPAND-ROUTE

&gt;saf&gt;cm&gt;road-routes.lisp

MOUSE-ON-BRIDGE-APPROACH-POINT

&gt;saf&gt;cm&gt;road-routes.lisp

SEGMENT-THRU-WATER

&gt;saf&gt;cm&gt;water-check.lisp

Called by: GET-BRIDGE-ROUTE

&gt;saf&gt;cm&gt;road-routes.lisp

Description: None

**2.2.4.4.22 MOUSE-ON-BRIDGE-APPROACH-POINT**

Definition 22

&gt;saf&gt;cm&gt;road-routes.lisp

Type: Function

Arguments: (STREAM)

Outputs:

Calls: SINGLE-POINT

&gt;map&gt;control.lisp

\*OPFOR-IO\*

&gt;saf&gt;sys&gt;vars.lisp

SAY

&gt;saf&gt;sys&gt;macros.lisp

POINT

&gt;saf&gt;interface&gt;model-menu.lisp

POINT

&gt;saf&gt;interface&gt;model-menu.lisp

Called by: GET-BRIDGE-POINTS

&gt;saf&gt;cm&gt;road-routes.lisp

Description: None



**2.2.4.4.23 FIND-NEAREST-BRIDGE**

Definition 23

&gt;saf&gt;cm&gt;road-routes.lisp

Type: Function

Arguments: (X-WC Y-WC STREAM &OPTIONAL (THRESHOLD (MAX (/ 10  
(WINDOW-SCALE STREAM)) 100))  
(QUAD-TREE \*QUAD-TREE\*))

Outputs:

Calls: \*ROAD-SEGMENT-ARRAY\*

&gt;map&gt;terrain-vars.lisp

\*QUAD-TREE\*

&gt;map&gt;terrain-vars.lisp

DISTANCE

&gt;map&gt;utilities.lisp

GET-NEIGHBOR-QUAD-ROADS

&gt;saf&gt;cm&gt;road-routes.lisp

Called by: GET-BRIDGE-ROUTE

&gt;saf&gt;cm&gt;road-routes.lisp

Description: None

**2.2.4.5 CSU cm>route-finder.lisp**

This unit contains the routines that determine the shortest road path between any two road points. An A\* algorithm is used to generate this path; A\* algorithms are discussed in most introductory AI texts, such as Winston's Artificial Intelligence, chapter 4, p. 115. These routines are used by the road route expansion routines.

**2.2.4.5.1 'FIND-ROUTE**

Definition 1

&gt;saf&gt;cm&gt;route-finder.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.4.5.2 FIND-ROUTE**

Definition 2

&gt;saf&gt;cm&gt;route-finder.lisp

Type: Function

Arguments: (FROM-INTERSECTION TO-INTERSECTION &amp;KEY (MAX-DISTANCE 25000) (EXPAND T))

Outputs:

Calls: **\*ROAD-INTERSECTION-ARRAY\***  
 >map>terrain-vars.lisp  
**EXPAND-FIRST-ROUTE**  
 >saf>cm>route-finder.lisp  
**SORT-ROUTE-QUEUE**  
 >saf>cm>route-finder.lisp  
**DISTANCE-BETWEEN-INTERSECTIONS**  
 >saf>cm>route-finder.lisp  
**EXPAND-ROUTE-INTO-POINTS**  
 >saf>cm>route-finder.lisp

Called by: **EXPAND-ROAD-ROUTE**  
 >saf>cm>road-routes.lisp

Description: None  
**2.2.4.5.3 EXPAND-FIRST-ROUTE**  
 Definition 3

>saf>cm>route-finder.lisp  
 Type: Function  
 Arguments: (ROUTE-QUEUE TO-INTERSECTION)

Outputs:  
 Calls: **\*ROAD-SEGMENT-ARRAY\***  
 >map>terrain-vars.lisp  
**\*ROAD-INTERSECTION-ARRAY\***  
 >map>terrain-vars.lisp  
**DISTANCE-BETWEEN-INTERSECTIONS**  
 >saf>cm>route-finder.lisp

Called by: **FIND-ROUTE**  
 >saf>cm>route-finder.lisp

Description: None

**2.2.4.5.4 SORT-ROUTE-QUEUE**  
 Definition 4

>saf>cm>route-finder.lisp  
 Type: Function  
 Arguments: (QUEUE MAX-DISTANCE)

Outputs:  
 Calls: **PARTIAL-SORT**  
 >saf>cm>route-finder.lisp

Called by: **FIND-ROUTE**  
 >saf>cm>route-finder.lisp

Description: None

**2.2.4.5.5 PARTIAL-SORT**  
 Definition 5

>saf>cm>route-finder.lisp  
 Type: Function  
 Arguments: (SEQ)  
 Outputs:

Calls: FIND-SHORTEST  
>saf>cm>route-finder.lisp  
Called by: SORT-ROUTE-QUEUE  
>saf>cm>route-finder.lisp  
Description: None

#### 2.2.4.5.6 FIND-SHORTEST

Definition 6

>saf>cm>route-finder.lisp  
Type: Function  
Arguments: (SEQ)  
Outputs:  
Calls: FIND-SHORTEST  
>saf>cm>route-finder.lisp  
Called by: FIND-SHORTEST  
>saf>cm>route-finder.lisp  
PARTIAL-SORT  
>saf>cm>route-finder.lisp  
Description: None

#### 2.2.4.5.7 TRIM-REDUNDANCY

Definition 7

>saf>cm>route-finder.lisp  
Type: Function  
Arguments: (QUEUE FINAL-POINTS)  
Outputs:  
Calls: TRIM-REDUNDANCY  
>saf>cm>route-finder.lisp  
Called by: TRIM-REDUNDANCY  
>saf>cm>route-finder.lisp  
Description: None

#### 2.2.4.5.8 DISTANCE-BETWEEN-INTERSECTIONS

Definition 8

>saf>cm>route-finder.lisp  
Type: Function  
Arguments: (I0 I1)  
Outputs:  
Calls: \*ROAD-INTERSECTION-ARRAY\*  
>map>terrain-vars.lisp  
DISTANCE  
>map>utilities.lisp  
Called by: EXPAND-FIRST-ROUTE  
>saf>cm>route-finder.lisp  
FIND-ROUTE  
>saf>cm>route-finder.lisp  
Description: None

**2.2.4.5.9 EXPAND-ROUTE-INTO-POINTS**

Definition 9

```

>saf>cm>route-finder.lisp
Type: Function
Arguments: (INTERSECTION-LIST)
Outputs:
Calls: *ROAD-SEGMENT-ARRAY*
       >map>terrain-vars.lisp
       *ROAD-INTERSECTION-ARRAY*
       >map>terrain-vars.lisp
       NEAR
       >map>utilities.lisp
       *OPFOR-IO*
       >saf>sys>vars.lisp
       SAY
       >saf>sys>macros.lisp
       REVERSE-XY
       >saf>cm>control-measure.lisp
       EXPAND-ROUTE-INTO-POINTS
       >saf>cm>route-finder.lisp
Called by: EXPAND-ROUTE-INTO-POINTS
          >saf>cm>route-finder.lisp
          FIND-ROUTE
          >saf>cm>route-finder.lisp
Description: None

```

**2.2.4.6 CSU cm>route.lisp**

This unit contains the definition of the route control measure structure, as well as the routines to manipulate and display them. Like other control measures, the route object has an associated behavior object, route-behavior, as well as make-behavior and review-data methods. Other methods permit routes to be drawn and erased. Each route instance includes a list of route-points in its *points* slot. Route methods permit these points to be moved, deleted and inserted before or after existing points. The function make-route is the top-level code that presents menu options to the user for creating various kinds of routes, including air routes, cross-country routes, road routes, and bridge crossings. It calls route construction functions such as get-road-route and get-bridge-route.

The route control measure differs from other control measures in that its geometrical structure may include more information than the positions of the control measure points of which it is composed. In addition to these points, the geometry of a road-route is also defined by the road paths from one road-point to the next. These paths are not represented by control measure points, but by lists of road segment indices, together with their traversal directions. These lists are placed in the expanded-route slot of the road-points that make up the route control measure instance.

Because of this extra geometrical data, the move-point, delete-point, insert-point-after, and insert-point-before methods for the route control measure are more complex than the corresponding methods for other control measures. In the other control measures, the movement, deletion and insertion of the control-measure points may require erasing and/or drawing of connecting lines. In the route control measure, connecting lines are replaced by connecting road-paths, and they have to be created before they can be drawn. Thus,

whenever a point is positioned, a call to `expand-route` (in CSU `cm>road-routes.lisp`) is made to find new connecting paths, which are then placed in the expanded-route slots of the appropriate road-points. These new paths are drawn by calls to `draw-expanded-route-core`. Old connecting paths are erased by similar calls, using an `erase-alu` instead of a `draw-alu`.

Cross-country routes must be checked for water-crossings as well, so the `move-point` and `insert-point` methods call `check-route-segment`, a route method that not only checks for water crossings, but gives the user an option to modify the route, using the water avoidance algorithms.

#### 2.2.4.6.1 \*ASK-USER\*

Definition 1

```
>saf>cm>route.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: (METHOD CHECK-ROUTE-SEGMENT ROUTE)
>saf>cm>route.lisp
(METHOD CHECK ROUTE)
>saf>cm>route.lisp
(METHOD INSERT-POINT-BEFORE ROUTE)
>saf>cm>route.lisp
(METHOD INSERT-POINT-AFTER ROUTE)
>saf>cm>route.lisp
(METHOD DELETE-POINT ROUTE)
>saf>cm>route.lisp
(METHOD MOVE-POINT ROUTE)
>saf>cm>route.lisp
Description: None
```

#### 2.2.4.6.2 ROUTE

Definition 2

```
>saf>cm>route.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: None
Called by: (WRITE-INSTANCE-VARIABLE (SETF ROUTE) SUB-TASK ROUTE)
No Source File Record
(READ-INSTANCE-VARIABLE ROUTE SUB-TASK ROUTE)
No Source File Record
(LOCATE-INSTANCE-VARIABLE (LOCF ROUTE) CM-POINT-BEHAVIOR
ROUTE)
No Source File Record
(WRITE-INSTANCE-VARIABLE (SETF ROUTE) CM-POINT-BEHAVIOR
ROUTE)
No Source File Record
(READ-INSTANCE-VARIABLE ROUTE CM-POINT-BEHAVIOR ROUTE)
```

No Source File Record  
(METHOD REEXECUTE-SUB-TASK SUB-TASK)  
>saf>ui>subordinate-tasking.lisp  
(METHOD DISPLAY-SUB-TASKING SUB-TASK)  
>saf>ui>subordinate-tasking.lisp  
(METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK)  
>saf>ui>subordinate-tasking.lisp  
(METHOD EXECUTE-SUB-TASK SUB-TASK)  
>saf>ui>subordinate-tasking.lisp  
(METHOD MAKE-INSTANCE SUB-TASK AFTER)  
>saf>ui>subordinate-tasking.lisp  
(METHOD COPY-BEHAVIOR CM-POINT-BEHAVIOR)  
>saf>cm>point.lisp  
(METHOD SEND-BEH-INFO CM-POINT-BEHAVIOR)  
>saf>cm>point.lisp  
(METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK)  
>saf>ui>subordinate-tasking.lisp  
MAKE-POINT-BEHAVIOR  
>saf>cm>point.lisp  
MAKE-ROUTE  
>saf>cm>route.lisp  
CM-ROUTE?  
>saf>cm>route.lisp  
FIND-ROUTE-AROUND-WATER  
>saf>cm>water-avoidance.lisp  
ROAD-SEGMENTS-FROM-INTERSECTIONS  
>saf>cm>road-routes.lisp  
FIND-SHORTEST-ROUTE  
>saf>cm>road-routes.lisp  
EXPAND-ROAD-ROUTE  
>saf>cm>road-routes.lisp  
SEND-EXECUTE-OVERLAY  
>saf>network>commands.lisp  
SEND-ROUTE  
>saf>network>commands.lisp  
SEND-POINT  
>saf>network>commands.lisp  
MAKE-ROUTE  
>saf>cm>route.lisp  
(METHOD COPY ROUTE)  
>saf>cm>route.lisp  
CM-ROUTE?  
>saf>cm>route.lisp  
SUB-TASK  
>saf>ui>subordinate-tasking.lisp  
CM-POINT-BEHAVIOR  
>saf>cm>point.lisp

Description: None

**2.2.4.6.3 CM-ROUTE?**

Definition 3

```
>saf>cm>route.lisp
Type: Function
Arguments: (ROUTE)
Outputs:
Calls: ROUTE
       >saf>cm>route.lisp
       ROUTE
       >saf>cm>route.lisp
       ROUTE
       >saf>cm>route.lisp
       ROUTE
       >saf>cm>route.lisp
       ROUTE
       >saf>cm>route.lisp
       ROUTE
       >saf>cm>route.lisp
       ROUTE
       >saf>cm>route.lisp
       ROUTE
Called by: None
Description: None
```

**2.2.4.6.4 ROUTE-BEHAVIOR**

Definition 4

```
>saf>cm>route.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: STORABLE-MIXIN
       >saf>objects>storable-mixin.lisp
       CONTROL-MEASURE-BEHAVIOR
       >saf>cm>control-measure.lisp
Called by: None
Description: None
```

**2.2.4.6.5 (METHOD COPY-BEHAVIOR ROUTE-BEHAVIOR)**

Definition 5

```
>saf>cm>route.lisp
Type: Method
Arguments: (CM)
Outputs:
Calls: None
Called by: None
Description: None
```

**2.2.4.6.6 (METHOD MAKE-BEHAVIOR ROUTE)**

Definition 6

&gt;saf&gt;cm&gt;route.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.2.4.6.7 (METHOD MAKE-INSTANCE ROUTE AFTER)**

Definition 7

&gt;saf&gt;cm&gt;route.lisp

Type: Method

Arguments: (&amp;REST INIT-ARGS)

Outputs:

Calls: None

Called by: None

Description: None

**2.2.4.6.8 (METHOD INITIALIZE-POINTS ROUTE)**

Definition 8

&gt;saf&gt;cm&gt;route.lisp

Type: Method

Arguments: (POINT-LIST)

Outputs:

Calls: POINT

&gt;saf&gt;interface&gt;model-menu.lisp

SAF

&gt;saf&gt;ui&gt;frame.lisp

CONTROL-MEASURE

&gt;saf&gt;cm&gt;control-measure.lisp

CONTROL-MEASURE

&gt;saf&gt;cm&gt;control-measure.lisp

POINT

&gt;saf&gt;interface&gt;model-menu.lisp

Called by: None

Description: None

**2.2.4.6.9 (METHOD REVIEW-DATA ROUTE)**

Definition 9

&gt;saf&gt;cm&gt;route.lisp

Type: Method

Arguments: ()

Outputs:



Calls: None  
Called by: None  
Description: None

**2.2.4.6.10 (METHOD PAINT-NAME ROUTE)**  
Definition 10

>saf>cm>route.lisp  
Type: Method  
Arguments: (STREAM ALU)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.2.4.6.11 (METHOD PAINT ROUTE)**  
Definition 11

>saf>cm>route.lisp  
Type: Method  
Arguments: (STREAM ALU)  
Outputs:  
Calls: WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
DRAW-EXPANDED-ROUTE-CORE  
>saf>cm>road-routes.lisp  
Called by: None  
Description: None

**2.2.4.6.12 (METHOD DRAW ROUTE)**  
Definition 12

>saf>cm>route.lisp  
Type: Method  
Arguments: (STREAM)  
Outputs:  
Calls: \*OVERLAY-ALU\*  
>map>color-map.lisp  
Called by: None  
Description: None

**2.2.4.6.13 (METHOD ERASE ROUTE)**  
Definition 13

>saf>cm>route.lisp  
Type: Method  
Arguments: (STREAM)  
Outputs:

**Calls: \*ERASE-OVERLAY-ALU\***

>map>color-map.lisp

**Called by: None**

**Description: None**

#### 2.2.4.6.14 (METHOD ORTHOGONALIZE ROUTE)

Definition 14

>saf>cm>route.lisp

**Type: Method**

**Arguments: ()**

**Outputs:**

>saf>interface>model-menu.lisp

POINT

>saf>interface>model-menu.lisp

**Called by: None**

**Description: None**

#### 2.2.4.6.15 (METHOD MOVE-POINT ROUTE)

Definition 15

>saf>cm>route.lisp

**Type: Method**

**Arguments: (POINT)**

**Outputs:**

**Calls: WITH-INTEGER-CONVERSION-MODE**

>map>utilities.lisp

WITH-MAP-GRAPHICS

>map>utilities.lisp

WITH-FAST-MAP-GRAPHICS

>map>utilities.lisp

\*OVERLAY-ALU\*

>map>color-map.lisp

\*ERASE-OVERLAY-ALU\*

>map>color-map.lisp

SINGLE-POINT

>map>control.lisp

\*PVD-DISPLAY\*

>saf>sys>vars.lisp

POINT

>saf>interface>model-menu.lisp

GET-ROAD-POINT

>saf>cm>road-routes.lisp

EXPAND-ROUTE

>saf>cm>road-routes.lisp

DRAW-EXPANDED-ROUTE-CORE

>saf>cm>road-routes.lisp

\*ASK-USER\*

>saf>cm>route.lisp

POINT

>saf>interface>model-menu.lisp

Called by: None  
 Description: None  
**2.2.4.6.16 (METHOD DELETE-POINT ROUTE)**  
 Definition 16

```

    >saf>cm>route.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: WITH-INTEGGER-CONVERSION-MODE
    >map>utilities.lisp
    WITH-MAP-GRAPHICS
    >map>utilities.lisp
    WITH-FAST-MAP-GRAPHICS
    >map>utilities.lisp
    *OVERLAY-ALU*
    >map>color-map.lisp
    *ERASE-OVERLAY-ALU*
    >map>color-map.lisp
    *PVD-DISPLAY*
    >saf>sys>vars.lisp
    POINT
    >saf>interface>model-menu.lisp
    EXPAND-ROUTE
    >saf>cm>road-routes.lisp
    DRAW-EXPANDED-ROUTE-CORE
    >saf>cm>road-routes.lisp
    *ASK-USER*
    >saf>cm>route.lisp
    POINT
    >saf>interface>model-menu.lisp
  
```

Called by: None  
 Description: None

**2.2.4.6.17 (METHOD INSERT-POINT-AFTER ROUTE)**  
 Definition 17

```

    >saf>cm>route.lisp
Type: Method
Arguments: (POINT &OPTIONAL XY PT-TYPE)
Outputs:
Calls: WITH-INTEGGER-CONVERSION-MODE
    >map>utilities.lisp
    WITH-MAP-GRAPHICS
    >map>utilities.lisp
    WITH-FAST-MAP-GRAPHICS
    >map>utilities.lisp
    *OVERLAY-ALU*
    >map>color-map.lisp
    *ERASE-OVERLAY-ALU*
    >map>color-map.lisp
    SINGLE-POINT
    >map>control.lisp
  
```

```

*PVD-DISPLAY*
>saf>sys>vars.lisp
MENU-CHOOSE
>saf>sys>utilities.lisp
POINT
>saf>interface>model-menu.lisp
SAF
>saf>ui>frame.lisp
CONTROL-MEASURE
>saf>cm>control-measure.lisp
CONTROL-MEASURE
>saf>cm>control-measure.lisp
GET-ROAD-SEGMENT-POINT
>saf>cm>road-routes.lisp
EXPAND-ROUTE
>saf>cm>road-routes.lisp
DRAW-EXPANDED-ROUTE-CORE
>saf>cm>road-routes.lisp
*ASK-USER*
>saf>cm>route.lisp
POINT
>saf>interface>model-menu.lisp

```

Called by: None

Description: None

#### 2.2.4.6.18 (METHOD INSERT-POINT-BEFORE ROUTE)

Definition 18

```

>saf>cm>route.lisp
Type: Method
Arguments: (POINT)
Outputs:
Calls: WITH-INTEGGER-CONVERSION-MODE
>map>utilities.lisp
WITH-MAP-GRAPHICS
>map>utilities.lisp
WITH-FAST-MAP-GRAPHICS
>map>utilities.lisp
*OVERLAY-ALU*
>map>color-map.lisp
*ERASE-OVERLAY-ALU*
>map>color-map.lisp
SINGLE-POINT
>map>control.lisp
*PVD-DISPLAY*
>saf>sys>vars.lisp
MENU-CHOOSE
>saf>sys>utilities.lisp
POINT
>saf>interface>model-menu.lisp
SAF
>saf>ui>frame.lisp

```

CONTROL-MEASURE  
 >saf>cm>control-measure.lisp  
 CONTROL-MEASURE  
 >saf>cm>control-measure.lisp  
 GET-ROAD-SEGMENT-POINT  
 >saf>cm>road-routes.lisp  
 EXPAND-ROUTE  
 >saf>cm>road-routes.lisp  
 DRAW-EXPANDED-ROUTE-CORE  
 >saf>cm>road-routes.lisp  
 \*ASK-USER\*  
 >saf>cm>route.lisp  
 POINT  
 >saf>interface>model-menu.lisp

Called by: None  
 Description: None

#### 2.2.4.6.19 (METHOD CHECK ROUTE)

Definition 19

>saf>cm>route.lisp  
 Type: Method  
 Arguments: ()  
 Outputs:  
 Calls: \*ASK-USER\*  
 >saf>cm>route.lisp  
 Called by: None  
 Description: None

#### 2.2.4.6.20 (METHOD CHECK-ROUTE-SEGMENT ROUTE)

Definition 20

>saf>cm>route.lisp  
 Type: Method  
 Arguments: (P1 POINTS)  
 Outputs:  
 Calls: \*OPFOR-IO\*  
 >saf>sys>vars.lisp  
 SAY  
 >saf>sys>macros.lisp  
 MENU-CHOOSE  
 >saf>sys>utilities.lisp  
 SEGMENT-THRU-WATER  
 >saf>cm>water-check.lisp  
 FIND-ROUTE-AROUND-WATER  
 >saf>cm>water-avoidance.lisp  
 \*ASK-USER\*  
 >saf>cm>route.lisp  
 Called by: None  
 Description: None

**2.2.4.6.21 (METHOD SEND-CM-INFO ROUTE)**

Definition 21

```

>saf>cm>route.lisp
Type: Method
Arguments: (BEH)
Outputs:
Calls: None
Called by: None
Description: None

```

**2.2.4.6.22 (METHOD COPY ROUTE)**

Definition 22

```

>saf>cm>route.lisp
Type: Method
Arguments: ()
Outputs:
Calls: NAME
>saf>sysdcl.lisp
POINT
>saf>interface>model-menu.lisp
ROUTE
>saf>cm>route.lisp
CONTROL-MEASURE
>saf>cm>control-measure.lisp
CONTROL-MEASURE
>saf>cm>control-measure.lisp
ROUTE
>saf>cm>route.lisp
ROUTE
>saf>cm>route.lisp
POINT
>saf>interface>model-menu.lisp
Called by: None
Description: None

```

**2.2.4.6.23 ROUTE**

Definition 23

```

>saf>cm>route.lisp
Type: COMPILE-FLAVOR-METHODS
Arguments: ()
Outputs:
Calls: None
Called by: (WRITE-INSTANCE-VARIABLE (SETF ROUTE) SUB-TASK ROUTE)
No Source File Record
(READ-INSTANCE-VARIABLE ROUTE SUB-TASK ROUTE)
No Source File Record
(LOCATE-INSTANCE-VARIABLE (LOCF ROUTE) CM-POINT-BEHAVIOR
ROUTE)
No Source File Record

```

(WRITE-INSTANCE-VARIABLE (SETF ROUTE) CM-POINT-BEHAVIOR  
ROUTE)

No Source File Record

(READ-INSTANCE-VARIABLE ROUTE CM-POINT-BEHAVIOR ROUTE)

No Source File Record

(METHOD REEXECUTE-SUB-TASK SUB-TASK)

>saf>ui>subordinate-tasking.lisp

(METHOD DISPLAY-SUB-TASKING SUB-TASK)

>saf>ui>subordinate-tasking.lisp

(METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK)

>saf>ui>subordinate-tasking.lisp

(METHOD EXECUTE-SUB-TASK SUB-TASK)

>saf>ui>subordinate-tasking.lisp

(METHOD MAKE-INSTANCE SUB-TASK AFTER)

>saf>ui>subordinate-tasking.lisp

(METHOD COPY-BEHAVIOR CM-POINT-BEHAVIOR)

>saf>cm>point.lisp

(METHOD SEND-BEH-INFO CM-POINT-BEHAVIOR)

>saf>cm>point.lisp

(METHOD CHOOSE-SUB-TASK-PARAMETERS SUB-TASK)

>saf>ui>subordinate-tasking.lisp

MAKE-POINT-BEHAVIOR

>saf>cm>point.lisp

MAKE-ROUTE

>saf>cm>route.lisp

CM-ROUTE?

>saf>cm>route.lisp

FIND-ROUTE-AROUND-WATER

>saf>cm>water-avoidance.lisp

ROAD-SEGMENTS-FROM-INTERSECTIONS

>saf>cm>road-routes.lisp

FIND-SHORTEST-ROUTE

>saf>cm>road-routes.lisp

EXPAND-ROAD-ROUTE

>saf>cm>road-routes.lisp

SEND-EXECUTE-OVERLAY

>saf>network>commands.lisp

SEND-ROUTE

>saf>network>commands.lisp

SEND-POINT

>saf>network>commands.lisp

MAKE-ROUTE

>saf>cm>route.lisp

(METHOD COPY ROUTE)

>saf>cm>route.lisp

CM-ROUTE?

>saf>cm>route.lisp

SUB-TASK

>saf>ui>subordinate-tasking.lisp

CM-POINT-BEHAVIOR

>saf>cm>point.lisp

Description: None

**2.2.4.6.24 MAKE-ROUTE**

Definition 24

&gt;saf&gt;cm&gt;route.lisp

Type: Function

Arguments: (OVERLAY STREAM)

Outputs:

Calls: WITH-INTEGER-CONVERSION-MODE

&gt;map&gt;utilities.lisp

WITH-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

WITH-FAST-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

\*OVERLAY-ALU\*

&gt;map&gt;color-map.lisp

RUBBER-LINE

&gt;map&gt;control.lisp

MENU-CHOOSE

&gt;saf&gt;sys&gt;utilities.lisp

ROUTE

&gt;saf&gt;cm&gt;route.lisp

ROUTE

&gt;saf&gt;cm&gt;route.lisp

REVERSE-XY

&gt;saf&gt;cm&gt;control-measure.lisp

XY-LIST-TO-ROUTE-POINTS

&gt;saf&gt;cm&gt;route-point.lisp

GET-ROAD-ROUTE

&gt;saf&gt;cm&gt;road-routes.lisp

DRAW-EXPANDED-ROUTE-CORE

&gt;saf&gt;cm&gt;road-routes.lisp

GET-BRIDGE-ROUTE

&gt;saf&gt;cm&gt;road-routes.lisp

ROUTE

&gt;saf&gt;cm&gt;route.lisp

ROUTE

&gt;saf&gt;cm&gt;route.lisp

ROUTE

&gt;saf&gt;cm&gt;route.lisp

ROUTE

&gt;saf&gt;cm&gt;route.lisp

Called by: (METHOD ADD-NEW-CONTROL-MEASURE OVERLAY)

&gt;saf&gt;cm&gt;overlay.lisp

Description: None



## 2.3 BATTLEMASTER CSC

The Battlemaster CSC implements the interface which the battlemaster uses to select and initialize the SAF forces. The battlemaster screen is password protected so that SAF commanders cannot create new forces on their own. The battlemaster screen is also used to connect the workstation to a simhost. The battlemaster interface allows the selection and placement of vehicles and units either by clicking on a location on the map display for tanks, IFVs, RWAs, and ADAs or by selecting an airport on the edge of the database for FWAs. Once the units the battlemaster wants to place under the control of this workstation have been selected, the battlemaster can choose to save away their positions, or to go ahead and create them on the simhost. Once the battlemaster creates the vehicles, the workstation automatically goes into the commander mode. In addition to the selection and placement of units, the battlemaster screen allows the battlemaster to recall previous selections, restore scenarios, designate whether the workstation display is in omniscient view or commander's view, set unit marksmanship, and designate the site-host address of the stealth providing the out-the-window view. The difference between recalling selections and scenarios is that selection files contain only the positions of units which have been selected (positioned) and stored, while scenarios are snapshots of positions of created vehicles and overlays, including their fuel and ammunition status parameters. Recalling a scenario will automatically cause the vehicles to be created and their status parameters restored, while recalling a selection only loads the vehicles into the battlemaster workspace. Figure 2.3-1 shows the sub-level CSCs of the Battlemaster CSC.

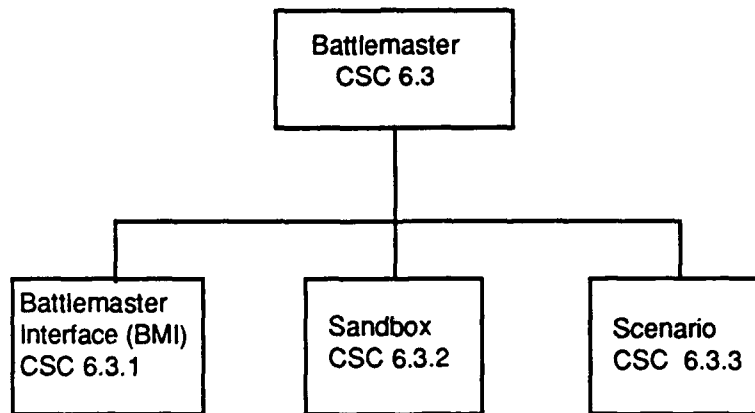


Figure 2.3-1 CSCs of the Battlemaster CSC

### 2.3.1 Battlemaster Interface (BMI) CSC

The BMI CSC defines the frame which comprises the battlemaster screen on the monochrome monitor. It also defines the menus which are used by the battlemaster to issue his commands. The BMI CSC contains the following CSUs:

```
bmi>bmi-frame.lisp csu  
bmi>commands.lisp csu  
bmi>utilities.lisp csu  
bmi>airport.lisp csu  
bmi>presentations-types.lisp csu
```

### 2.3.1.1 CSU `bmi>bmi-frame.lisp`

This unit contains the functions and data-structures used by the battlemaster screen. These include *bmi*, the battlemaster interface flavor, and a user-interface function, *accept-bmi-options*. This function allows the user to specify options for workstation and unit alignment, unit tactics, battle view, marksmanship, and the stealth vehicle site and host numbers. Other functions handle airport lists, display the FWA pane on the battlemaster screen, show connection to a Simhost (bmi method *Display-Connection-State*), and enable creation, storage and removal of sandbox objects.

#### 2.3.1.1.1 BMI

Definition 1

```
>saf>bmi>bmi-frame.lisp
Type: Flavor
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.3.1.1.2 (METHOD ENABLE-MMSHIP-CHANGE BMI)

Definition 2

```
>saf>bmi>bmi-frame.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.3.1.1.3 (METHOD SET-ENABLE-MMSHIP-CHANGE BMI)

Definition 3

```
>saf>bmi>bmi-frame.lisp
Type: Method
Arguments: (ENABLE)
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.3.1.1.4 WORKSTATION-MMSHIP-CHANGE

Definition 4

```
>saf>bmi>bmi-frame.lisp
Type: Function
Arguments: ()
```

Outputs:

Calls: \*BMI-PROGRAM\*

>saf>sys>vars.lisp

Called by: (METHOD SPECIFY-RULES-OF-ENGAGEMENT GUNNER)

>saf>objects>gunner.lisp

Description: None

### 2.3.1.1.5 (METHOD WS-ALIGNMENT BMI)

Definition 5

>saf>bmi>bmi-frame.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

### 2.3.1.1.6 (METHOD SET-WS-ALIGNMENT BMI)

Definition 6

>saf>bmi>bmi-frame.lisp

Type: Method

Arguments: (ALIGN)

Outputs:

Calls: None

Called by: None

Description: None

### 2.3.1.1.7 WORKSTATION-ALIGNMENT

Definition 7

>saf>bmi>bmi-frame.lisp

Type: Function

Arguments: ()

Outputs:

Calls: \*BMI-PROGRAM\*

>saf>sys>vars.lisp

Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)

>saf>bmi>bmi-frame.lisp

(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)

>saf>bmi>bmi-frame.lisp

(DEFUN-IN-FLAVOR ACCEPT-TACTICS-AND-TEAM BMI)

No Source File Record

MAKE-SANDBOX-OBJECT

>saf>sandbox>sandbox-object.lisp

CISS-FOR-CONTROL-MEASURE

>saf>sys>interim-model.lisp

Description: None

**2.3.1.1.8 (METHOD BATTLE-VIEW BMI)**

Definition 8

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.3.1.1.9 (METHOD SET-BATTLE-VIEW BMI)**

Definition 9

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Method

Arguments: (VIEW)

Outputs:

Calls: None

Called by: None

Description: None

**2.3.1.1.10 WORKSTATION-BATTLE-VIEW**

Definition 10

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Function

Arguments: ()

Outputs:

Calls: \*BMI-PROGRAM\*

&gt;saf&gt;sys&gt;vars.lisp

Called by: COM-OMNISCIENT-VIEW

&gt;saf&gt;objects&gt;simnet-agent.lisp

COLOR-SCREEN-MENU

&gt;saf&gt;ui&gt;mouse-interface.lisp

Description: None

**2.3.1.1.11 (METHOD BATTLE-SCHEME BMI)**

Definition 11

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.3.1.1.12 (METHOD SET-BATTLE-SCHEME BMI)**

Definition 12

>saf>bmi>bmi-frame.lisp  
 Type: Method  
 Arguments: (SCHEME)  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

**2.3.1.1.13 WORKSTATION-BATTLE-SCHEME**

Definition 13

>saf>bmi>bmi-frame.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: \*BMI-PROGRAM\*  
 >saf>sys>vars.lisp  
 Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)  
 >saf>bmi>bmi-frame.lisp  
 (METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)  
 >saf>bmi>bmi-frame.lisp  
 (DEFUN-IN-FLAVOR ACCEPT-TACTICS-AND-TEAM BMI)  
 No Source File Record  
 ALIGNMENT-FROM-FORCE-ID  
 >saf>bmi>bmi-frame.lisp  
 Description: None

**2.3.1.1.14 \*DEFAULT-BATTALION-NUMBER\***

Definition 14

>saf>bmi>bmi-frame.lisp  
 Type: Parameter  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: LOAD-SCENARIO  
 >saf>sys>new-storage.lisp  
 STORE-SCENARIO  
 >saf>sys>new-storage.lisp  
 (METHOD DISPLAY-TITLE SUBORDINATE-UNIT-TASKING)  
 >saf>ui>subordinate-tasking.lisp  
 (METHOD TOP-LEVEL SAF)  
 >saf>ui>frame.lisp  
 (METHOD ACCEPT-BMI-OPTIONS BMI)  
 >saf>bmi>bmi-frame.lisp  
 GET-BATTALION-NUMBER  
 >saf>bmi>bmi-frame.lisp  
 Description: None

**2.3.1.1.15 GET-BATTALION-NUMBER**

Definition 15

>saf>bmi>bmi-frame.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*DEFAULT-BATTALION-NUMBER\*  
>saf>bmi>bmi-frame.lisp  
Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
DISPLAY-WORKSTATION-BATTALION  
>saf>ui>task-org.lisp  
Description: None

**2.3.1.1.16 (METHOD ACCEPT-BMI-OPTIONS BMI)**

Definition 16

>saf>bmi>bmi-frame.lisp  
Type: Method  
Arguments: (STREAM)  
Outputs:  
Calls: \*VIEW-VEHICLE-ID\*  
>saf>sys>vars.lisp  
\*STEALTH-SITE-NUMBER\*  
>saf>sys>vars.lisp  
\*STEALTH-HOST-NUMBER\*  
>saf>sys>vars.lisp  
\*DEFAULT-BATTALION-NUMBER\*  
>saf>bmi>bmi-frame.lisp  
BATTALION-BUMPER  
>saf>bmi>presentation-types.lisp  
Called by: None  
Description: None

**2.3.1.1.17 (METHOD AFTER-PROGRAM-FRAME-SELECTION-HANDLER BMI)**

Definition 17

>saf>bmi>bmi-frame.lisp  
Type: Method  
Arguments: (FRAME)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.1.18 (METHOD BMI-SANDBOX BMI)**

Definition 18

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.3.1.1.19 (METHOD BMI-SET-SANDBOX BMI)**

Definition 19

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Method

Arguments: (NEW-SANDBOX)

Outputs:

Calls: None

Called by: None

Description: None

**2.3.1.1.20 (METHOD BMI-REMOVE-SANDBOX-OBJECT BMI)**

Definition 20

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Method

Arguments: (OBJECT)

Outputs:

Calls: \*PVD-DISPLAY\*

&gt;saf&gt;sys&gt;vars.lisp

ERASE-SANDBOX-OBJECT

&gt;saf&gt;sandbox&gt;sandbox-object.lisp

Called by: None

Description: None

**2.3.1.1.21 (METHOD BMI-CLEAR-SANDBOX BMI)**

Definition 21

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.3.1.1.22 (METHOD BMI-ADD-SANDBOX-OBJECT BMI)**  
Definition 22

>saf>bmi>bmi-frame.lisp  
Type: Method  
Arguments: (OBJECT)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.1.23 (METHOD BMI-AIRPORTS BMI)**  
Definition 23

>saf>bmi>bmi-frame.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.1.24 (METHOD BMI-SET-AIRPORTS BMI)**  
Definition 24

>saf>bmi>bmi-frame.lisp  
Type: Method  
Arguments: (NEW-AIRPORTS)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.1.25 (METHOD BMI-ADD-AIRPORT BMI)**  
Definition 25

>saf>bmi>bmi-frame.lisp  
Type: Method  
Arguments: (NEW-AIRPORT)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.1.26 (METHOD FIND-AIRPORT BMI)**  
Definition 26

>saf>bmi>bmi-frame.lisp  
Type: Method  
Arguments: (LOCATION)

---



Outputs:  
 Calls: None  
 Called by: None  
 Description: None

### 2.3.1.1.27 (METHOD DISPLAY-CONNECTION-STATE BMI)

Definition 27

```
>saf>bmi>bmi-frame.lisp
Type: Method
Arguments: (&OPTIONAL (STREAM NIL))
Outputs:
Calls: *BMI-PROGRAM*
       >saf>sys>vars.lisp
       SIMULATION-HOST
       >saf>network>vars.lisp
       STANDALONEP
       >saf>network>connection.lisp
       NO-CONNECTION
       >saf>bmi>presentation-types.lisp
       CONNECTION
       >saf>bmi>presentation-types.lisp
Called by: None
Description: None
```

### 2.3.1.1.28 (METHOD CREATE-MOCK-UNITS BMI)

Definition 28

```
>saf>bmi>bmi-frame.lisp
Type: Method
Arguments: ()
Outputs:
Calls: REALLY-MAKE-SANDBOX-OBJECT
       >saf>bmi>bmi-frame.lisp
Called by: None
Description: None
```

### 2.3.1.1.29 REALLY-MAKE-SANDBOX-OBJECT

Definition 29

```
>saf>bmi>bmi-frame.lisp
Type: Function
Arguments: (OBJECT)
Outputs:
Calls: NEW-SBX-UNIQUE-UNIT-ID
       >saf>sys>vars.lisp
       ADD-SANDBOX-TO-ALIST
       >saf>sys>vars.lisp
       MAP-ECHELON-TO-NUMBER
       >saf>sys>interim-model.lisp
       MAP-ECHELON-TYPE-TO-NUMBER
```

```

>saf>sys>interim-model.lisp
CREATE
>saf>network>vars.lisp
NET-MSG
>saf>rudp>outgoing.lisp
RETURN-FORCE-AND-COUNTRY-D-AND-O
>saf>bmi>bmi-frame.lisp
Called by: (METHOD CREATE-MOCK-UNITS BMI)
>saf>bmi>bmi-frame.lisp
Description: None

```

### 2.3.1.1.30 RETURN-FORCE-AND-COUNTRY-D-AND-O

Definition 30

```

>saf>bmi>bmi-frame.lisp
Type: Function
Arguments: (ALIGN)
Outputs:
Calls: ALIGNED-OFFENSE
>saf>sys>vars.lisp
ALIGNED-DEFENSE
>saf>sys>vars.lisp
ALIGNED-USSR
>saf>sys>vars.lisp
ALIGNED-US
>saf>sys>vars.lisp
COUNTRY-US
>saf>sys>vars.lisp
COUNTRY-USSR
>saf>sys>vars.lisp
DISTINGUISHED-FORCE
>saf>sys>vars.lisp
OTHER-FORCE
>saf>sys>vars.lisp
Called by: CREATE-STORED-INSTANCE
>saf>sys>new-storage.lisp
REALLY-MAKE-SANDBOX-OBJECT
>saf>bmi>bmi-frame.lisp
Description: None

```

### 2.3.1.1.31 ALIGNMENT-FROM-FORCE-ID

Definition 31

```

>saf>bmi>bmi-frame.lisp
Type: Function
Arguments: (FORCE-ID)
Outputs:
Calls: ALIGNED-OFFENSE
>saf>sys>vars.lisp
ALIGNED-DEFENSE
>saf>sys>vars.lisp
ALIGNED-USSR

```

```

>saf>sys>vars.lisp
ALIGNED-US
>saf>sys>vars.lisp
COUNTRY-US
>saf>sys>vars.lisp
DISTINGUISHED-FORCE
>saf>sys>vars.lisp
WORKSTATION-BATTLE-SCHEME
>saf>bmi>bmi-frame.lisp
Called by: PROCESS-VEHICLE-PAE-PKT
>saf>rudp>handle-incoming.lisp
PROCESS-VEHICLE-APPEARANCE-PKT
>saf>rudp>handle-incoming.lisp
Description: None

```

### 2.3.1.1.32 (ACCEPT-TACTICS-AND-TEAM BMI)

Definition 32

```

>saf>bmi>bmi-frame.lisp
Type: DEFSUBST-IN-FLAVOR
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

### 2.3.1.1.33 ACCEPT-PARAMETER-FROM-SEQUENCE

Definition 33

```

>saf>bmi>bmi-frame.lisp
Type: Macro
Arguments: (PARAMETER SEQUENCE PROMPT STREAM &OPTIONAL
QUERYID)
Outputs:
Calls: ACCEPT-PARAMETER-FROM-SEQUENCE
>saf>bmi>bmi-frame.lisp
Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)
>saf>bmi>bmi-frame.lisp
(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)
>saf>bmi>bmi-frame.lisp
ACCEPT-PARAMETER-FROM-SEQUENCE
>saf>bmi>bmi-frame.lisp
Description: None

```

### 2.3.1.1.34 ALL-ECHELONS

Definition 34

```

>saf>bmi>bmi-frame.lisp
Type: Function
Arguments: (TACTICS)

```

## Outputs:

Calls: GET-VEHICLE-ECHELONS-AND-TYPES

&gt;saf&gt;sys&gt;interim-model.lisp

Called by: (METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Description: None

**2.3.1.1.35 GET-ECHELON-TYPES**

Definition 35

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Function

Arguments: (ECHELON TACTICS)

## Outputs:

Calls: GET-VEHICLE-ECHELONS-AND-TYPES

&gt;saf&gt;sys&gt;interim-model.lisp

Called by: (METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Description: None

**2.3.1.1.36 BMI-FIND-FORMATIONS**

Definition 36

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Subst

Arguments: (ECHELON TYPE TACTICS)

## Outputs:

Calls: FIND-FORMATIONS

&gt;saf&gt;sys&gt;interim-model.lisp

Called by: (METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Description: None

**2.3.1.1.37 (METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)**

Definition 37

&gt;saf&gt;bmi&gt;bmi-frame.lisp

Type: Method

Arguments: (X Y BEARING &amp;OPTIONAL (STREAM NIL))

## Outputs:

Calls:  $\pi$ 

&gt;saf&gt;sys&gt;constants.lisp

 $2\pi$ 

&gt;saf&gt;sys&gt;constants.lisp

90DEG

&gt;saf&gt;sys&gt;constants.lisp

RAD-TO-MIL

&gt;saf&gt;sys&gt;constants.lisp

MIL-TO-RAD

&gt;saf&gt;sys&gt;constants.lisp

RADIANS-COMPASS-TO-RADIANS-MATH

```
>saf>sys>macros.lisp
RADIANS-MATH-TO-RADIANS-COMPASS
>saf>sys>macros.lisp
RADIANS-COMPASS-TO-MILS
>saf>sys>macros.lisp
RADIANS-MATH-TO-MILS
>saf>sys>macros.lisp
MILS-TO-RADIANS-COMPASS
>saf>sys>macros.lisp
MILS-TO-RADIANS-MATH
>saf>sys>macros.lisp
FIND-FORMATIONS
>saf>sys>interim-model.lisp
OPFOR
>saf>network>vars.lisp
BLUEFOR
>saf>network>vars.lisp
*MARKSMAN*
>saf>objects>gunner.lisp
*COMPETENT*
>saf>objects>gunner.lisp
*NOVICE*
>saf>objects>gunner.lisp
VEHICLE
>saf>objects>vehicle.lisp
VEHICLE
>saf>objects>vehicle.lisp
WORKSTATION-ALIGNMENT
>saf>bmi>bmi-frame.lisp
WORKSTATION-BATTLE-SCHEME
>saf>bmi>bmi-frame.lisp
GET-BATTALION-NUMBER
>saf>bmi>bmi-frame.lisp
ACCEPT-PARAMETER-FROM-SEQUENCE
>saf>bmi>bmi-frame.lisp
ALL-ECHELONS
>saf>bmi>bmi-frame.lisp
GET-ECHELON-TYPES
>saf>bmi>bmi-frame.lisp
BMI-FIND-FORMATIONS
>saf>bmi>bmi-frame.lisp
SANDBOX-OBJECT
>saf>bmi>presentation-types.lisp
CONVERT-UNIT-SIZE
>saf>bmi>utilities.lisp
CONVERT-ALIGNMENT
>saf>bmi>utilities.lisp
OPFOR-SYMBOL
>saf>bmi>utilities.lisp
MAYBE-LOAD-FORMATION-DATA
>saf>bmi>utilities.lisp
SANDBOX-OBJECT
>saf>bmi>presentation-types.lisp
TACTICS
```

>saf>bmi>presentation-types.lisp  
MILS

>saf>bmi>presentation-types.lisp  
COMPANY-BUMPER

>saf>bmi>presentation-types.lisp  
PLATOON-BUMPER

>saf>bmi>presentation-types.lisp

Called by: None

Description: None

### 2.3.1.1.38 FIND-ALL-FWA-ECHELONS

Definition 38

>saf>bmi>bmi-frame.lisp

Type: Function

Arguments: (TACTICS)

Outputs:

Calls: GET-VEHICLE-ECHELONS-AND-TYPES

>saf>sys>interim-model.lisp

Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)

>saf>bmi>bmi-frame.lisp

Description: None

### 2.3.1.1.39 (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)

Definition 39

>saf>bmi>bmi-frame.lisp

Type: Method

Arguments: (AIRPORT)

Outputs:

Calls: FIND-FORMATIONS

>saf>sys>interim-model.lisp

OPFOR

>saf>network>vars.lisp

BLUEFOR

>saf>network>vars.lisp

\*MARKSMAN\*

>saf>objects>gunner.lisp

\*COMPETENT\*

>saf>objects>gunner.lisp

\*NOVICE\*

>saf>objects>gunner.lisp

VEHICLE

>saf>objects>vehicle.lisp

VEHICLE

>saf>objects>vehicle.lisp

WORKSTATION-ALIGNMENT

>saf>bmi>bmi-frame.lisp

WORKSTATION-BATTLE-SCHEME

>saf>bmi>bmi-frame.lisp

GET-BATTALION-NUMBER

```

>saf>bmi>bmi-frame.lisp
ACCEPT-PARAMETER-FROM-SEQUENCE
>saf>bmi>bmi-frame.lisp
FIND-ALL-FWA-ECHELONS
>saf>bmi>bmi-frame.lisp
SANDBOX-OBJECT
>saf>bmi>presentation-types.lisp
CONVERT-UNIT-SIZE
>saf>bmi>utilities.lisp
CONVERT-ALIGNMENT
>saf>bmi>utilities.lisp
OPFOR-SYMBOL
>saf>bmi>utilities.lisp
MAYBE-LOAD-FORMATION-DATA
>saf>bmi>utilities.lisp
SANDBOX-OBJECT
>saf>bmi>presentation-types.lisp
TACTICS
>saf>bmi>presentation-types.lisp
COMPANY-BUMPER
>saf>bmi>presentation-types.lisp
PLATOON-BUMPER
>saf>bmi>presentation-types.lisp

```

Called by: None

Description: None

#### 2.3.1.1.40 (METHOD DISPLAY-FWA-PANE BMI)

Definition 40

```

>saf>bmi>bmi-frame.lisp
Type: Method
Arguments: (FWA-PANE)
Outputs:
Calls: *QUAD-TREE*
>map>terrain-vars.lisp
MAKE-AIRPORTS
>saf>bmi>airport.lisp

```

Called by: None

Description: None

#### 2.3.1.1.41 (METHOD DISPLAY-TOTALS-PANE BMI)

Definition 41

```

>saf>bmi>bmi-frame.lisp
Type: Method
Arguments: (STREAM)
Outputs:
Calls: SANDBOX-OBJECT
>saf>bmi>presentation-types.lisp
SANDBOX-OBJECT
>saf>bmi>presentation-types.lisp

```

Called by: None  
Description: None

#### 2.3.1.1.42 (METHOD REDISPLAY-TOTALS-PANE BMI) Definition 42

```
>saf>bmi>bmi-frame.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.3.1.1.43 (METHOD REDISPLAY-OPTIONS-PANE BMI) Definition 43

```
>saf>bmi>bmi-frame.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

### 2.3.1.2 CSU bmi>commands.lisp

This unit defines the menu and Command Processor (CP) commands used by the battlemaster screen. These include selecting, loading, saving and creating units, and the command that accepts the battlemaster password.

The symbolics function cp:define-command allows user-defined commands to be added to the CP command table; see the Symbolics manuals for details.

#### 2.3.1.2.1 BATTLEMASTER-SCREEN-P Definition 1

```
>saf>bmi>commands.lisp
Type: Subst
Arguments: ()
Outputs:
Calls: *OPFOR-FRAME*
>saf>sys>vars.lisp
Called by: (METHOD COM-ADD-AIRCRAFT-INTERNAL SAF)
No Source File Record
(METHOD COM-CREATE-UNITS-INTERNAL SAF)
No Source File Record
(METHOD COM-LOAD-SELECTIONS-INTERNAL SAF)
No Source File Record
(METHOD COM-SAVE-SELECTIONS-INTERNAL SAF)
```



No Source File Record  
(METHOD COM-RESTORE-EXERCISE-INTERNAL SAF)  
No Source File Record  
(METHOD COM-CLEAR-SELECTIONS-INTERNAL SAF)  
No Source File Record  
(METHOD COM-SELECT-UNITS-INTERNAL SAF)  
No Source File Record

Description: None

**2.3.1.2.2 (COM-SELECT-UNITS MENU-ACCELERATOR Select Units  
MENU-LEVEL BATTLEMASTER)**

Definition 2

>saf>bmi>commands.lisp  
Type: DEFINE-SAF-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.2.3 (COM-CLEAR-SELECTIONS MENU-ACCELERATOR Clear  
Selections MENU-LEVEL BATTLEMASTER)**

Definition 3

>saf>bmi>commands.lisp  
Type: DEFINE-SAF-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.2.4 (COM-RESTORE-EXERCISE MENU-ACCELERATOR Restore  
Exercise MENU-LEVEL BATTLEMASTER)**

Definition 4

>saf>bmi>commands.lisp  
Type: DEFINE-SAF-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.2.5 (COM-SAVE-SELECTIONS MENU-ACCELERATOR Save Selections MENU-LEVEL BATTLEMASTER)**

## Definition 5

>saf>bmi>commands.lisp  
Type: DEFINE-SAF-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.2.6 (COM-LOAD-SELECTIONS MENU-ACCELERATOR Load Selections MENU-LEVEL BATTLEMASTER)**

## Definition 6

>saf>bmi>commands.lisp  
Type: DEFINE-SAF-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.2.7 (COM-CREATE-UNITS MENU-ACCELERATOR Create Units MENU-LEVEL BATTLEMASTER)**

## Definition 7

>saf>bmi>commands.lisp  
Type: DEFINE-SAF-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.2.8 (COM-SHOW-SANDBOX)**

## Definition 8

>saf>bmi>commands.lisp  
Type: DEFINE-SAF-COMMAND  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.2.9 (COM-ADD-AIRCRAFT)**

Definition 9

```
>saf>bmi>commands.lisp
Type: DEFINE-SAF-COMMAND
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.3.1.2.10 COM-BATTLEMASTER**

Definition 10

```
>saf>bmi>commands.lisp
Type: CP Command
Arguments: ()
Outputs:
Calls: *BATTLEMASTER-PASSWORD*
>saf>sys>vars.lisp
Called by: None
Description: None
```

**2.3.1.2.11 COM-COMMANDER**

Definition 11

```
>saf>bmi>commands.lisp
Type: CP Command
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.3.1.3 CSU bmi>utilities.lisp**

This unit contains a few utilities used by the battlemaster code, for retrieving a sandbox from disk, converting unit size and alignment, making sandbox objects, locating a suitable local file server, and loading formation data from a file server.

**2.3.1.3.1 USER-CHOOSE**

Definition 1

```
>saf>bmi>utilities.lisp
Type: Function
Arguments: (CHOICES LABEL &OPTIONAL (STYLE '(DUTCH BOLD LARGE)))
Outputs:
Calls: None
```

Called by: (PRESENTATION-MOUSE-HANDLER SANDBOX-OBJECT-GESTURE)

No Source File Record  
(PRESENTATION-MOUSE-HANDLER MAKE-CONNECTION)  
No Source File Record  
RETRIEVE-A-SANDBOX  
>saf>bmi>utilities.lisp

Description: None

### 2.3.1.3.2 RETRIEVE-A-SANDBOX

Definition 2

>saf>bmi>utilities.lisp

Type: Function

Arguments: ()

Outputs:

Calls: \*BMI-PROGRAM\*

>saf>sys>vars.lisp

\*ACTIVE-SANDBOXES\*

>saf>sys>vars.lisp

DRAW-SANDBOX

>saf>sandbox>sandbox.lisp

ERASE-SANDBOX

>saf>sandbox>sandbox.lisp

ALL-SANDBOXES-AS-MENU-ITEMS

>saf>sandbox>utilities.lisp

USER-CHOOSE

>saf>bmi>utilities.lisp

Called by: (METHOD COM-LOAD-SELECTIONS-INTERNAL SAF)

No Source File Record

Description: None

### 2.3.1.3.3 CONVERT-UNIT-SIZE

Definition 3

>saf>bmi>utilities.lisp

Type: Function

Arguments: (UNIT-SIZE)

Outputs:

Calls: SAF

>saf>ui>frame.lisp

Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)

>saf>bmi>bmi-frame.lisp

(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)

>saf>bmi>bmi-frame.lisp

Description: None

**2.3.1.3.4 CONVERT-ALIGNMENT**

## Definition 4

>saf>bmi>utilities.lisp  
Type: Function  
Arguments: (ALIGNMENT)  
Outputs:  
Calls: SAF  
>saf>ui>frame.lisp  
Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
Description: None

**2.3.1.3.5 BMI-MAKE-SANDBOX-OBJECT**

## Definition 5

>saf>bmi>utilities.lisp  
Type: Function  
Arguments: (X Y BEARING)  
Outputs:  
Calls: \*BMI-PROGRAM\*  
>saf>sys>vars.lisp  
Called by: (METHOD COM-SELECT-UNITS-INTERNAL SAF)  
No Source File Record  
Description: None

**2.3.1.3.6 OPFOR-SYMBOL**

## Definition 6

>saf>bmi>utilities.lisp  
Type: Function  
Arguments: (SYMBOL)  
Outputs:  
Calls: SAF  
>saf>ui>frame.lisp  
Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
Description: None

**2.3.1.3.7 FIND-GOOD-LOCAL-FILE-SERVER**

## Definition 7

>saf>bmi>utilities.lisp  
Type: Function  
Arguments: ()

Outputs:  
Calls: None  
Called by: MAYBE-LOAD-FORMATION-DATA  
          >saf>bmi>utilities.lisp  
Description: None

#### 2.3.1.3.8 MAYBE-LOAD-FORMATION-DATA

Definition 8

          >saf>bmi>utilities.lisp  
Type: Function  
Arguments: (&OPTIONAL (HOST (FIND-GOOD-LOCAL-FILE-SERVER)))  
Outputs:  
Calls: GET-FORMATION-DATA  
          >saf>sys>interim-model.lisp  
          SAF  
          >saf>ui>frame.lisp  
          FIND-GOOD-LOCAL-FILE-SERVER  
          >saf>bmi>utilities.lisp  
Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)  
          >saf>bmi>bmi-frame.lisp  
          (METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)  
          >saf>bmi>bmi-frame.lisp  
Description: None

#### 2.3.1.4 CSU bmi>airport.lisp

This unit contains the data-structures and routines that implement the off-database airport concept. This allows aircraft to be created at airports that are located off the terrain. Off-terrain airports are located at 8 points of the compass, with a rectangular offset of 2500 meters from the terrain rectangle. These locations are shown at the left of the battlemaster screen. Lisp forms in this unit define the airport flavor, methods for creating and drawing airports, and functions to add airports to the battlemaster interface.

#### 2.3.1.4.1 AIRPORT-DATA

Definition 1

          >saf>bmi>airport.lisp  
Type: Function  
Arguments: (TERRAIN-X-EXTENT TERRAIN-Y-EXTENT)  
Outputs:  
Calls: None  
Called by: MAKE-AIRPORTS  
          >saf>bmi>airport.lisp  
Description: None

**2.3.1.4.2 DRAW-AIRPORT-LOCATION**

## Definition 2

>saf>bmi>airport.lisp

Type: Function

Arguments: (LOCATION XE YE N)

Outputs:

Calls: None

Called by: (METHOD DRAW AIRPORT)

>saf>bmi>airport.lisp

Description: None

**2.3.1.4.3 AIRPORT**

## Definition 3

>saf>bmi>airport.lisp

Type: Flavor

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.3.1.4.4 (METHOD MAKE-INSTANCE AIRPORT AFTER)**

## Definition 4

>saf>bmi>airport.lisp

Type: Method

Arguments: (&REST INIT-ARGS)

Outputs:

Calls: None

Called by: None

Description: None

**2.3.1.4.5 (METHOD DRAW AIRPORT)**

## Definition 5

>saf>bmi>airport.lisp

Type: Method

Arguments: (XE YE N STREAM)

Outputs:

Calls: DRAW-AIRPORT-LOCATION

>saf>bmi>airport.lisp

AIRPORT

>saf>bmi>presentation-types.lisp

Called by: None

Description: None

### 2.3.1.4.6 (METHOD MAKE-FWA-SANDBOX-OBJECT AIRPORT) Definition 6

>saf>bmi>airport.lisp  
 Type: Method  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

### 2.3.1.4.7 MAKE-AIRPORT Definition 7

>saf>bmi>airport.lisp  
 Type: Function  
 Arguments: (DATA)  
 Outputs:  
 Calls: AIRPORT  
 >saf>bmi>presentation-types.lisp  
 Called by: MAKE-AIRPORTS  
 >saf>bmi>airport.lisp  
 Description: None

### 2.3.1.4.8 MAKE-AIRPORTS Definition 8

>saf>bmi>airport.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: \*QUAD-TREE\*  
 >map>terrain-vars.lisp  
 AIRPORT-DATA  
 >saf>bmi>airport.lisp  
 MAKE-AIRPORT  
 >saf>bmi>airport.lisp  
 Called by: (METHOD DISPLAY-FWA-PANE BMI)  
 >saf>bmi>bmi-frame.lisp  
 Description: None

### 2.3.1.5 CSU bmi>presentation-types.lisp

This unit defines presentation types, actions, and command translators for the battlemaster screen. These presentations create mouse-sensitive regions for handling connection state, airports, tactics and simnet team, and sandbox objects.

See the Symbolics manuals for more information on functions such as *define-presentation-type* and *define-presentation-action*.



**2.3.1.5.1 NO-CONNECTION**

## Definition 1

>saf>bmi>presentation-types.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (PRESENTATION-MOUSE-HANDLER MAKE-CONNECTION)  
No Source File Record  
(METHOD DISPLAY-CONNECTION-STATE BMI)  
>saf>bmi>bmi-frame.lisp  
(PROPERTY NO-CONNECTION DEFTYPE)  
No Source File Record  
Description: None

**2.3.1.5.2 MAKE-CONNECTION**

## Definition 2

>saf>bmi>presentation-types.lisp  
Type: DEFINE-PRESENTATION-ACTION  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.3.1.5.3 CONNECTION**

## Definition 3

>saf>bmi>presentation-types.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (PRESENTATION-MOUSE-HANDLER END-CONNECTION)  
No Source File Record  
SAF PROGRAM-FRAME-OPTIONS  
>saf>ui>frame.lisp  
(METHOD DISPLAY-CONNECTION-STATE BMI)  
>saf>bmi>bmi-frame.lisp  
(PROPERTY CONNECTION DEFTYPE)  
No Source File Record  
Description: None

**2.3.1.5.4 END-CONNECTION**

## Definition 4

>saf>bmi>presentation-types.lisp  
Type: DEFINE-PRESENTATION-ACTION  
Arguments: ()

**Outputs:**

Calls: None

Called by: None

Description: None

**2.3.1.5.5 AIRPORT**

## Definition 5

&gt;saf&gt;bmi&gt;presentation-types.lisp

Type: DEFINE-PRESENTATION-TYPE

Arguments: ()

**Outputs:**

Calls: None

Called by: (PRESENTATION-MOUSE-HANDLER ADD-AIRCRAFT)

No Source File Record

(METHOD COM-ADD-AIRCRAFT-PARSER SAF)

No Source File Record

MAKE-AIRCRAFT

&gt;saf&gt;bmi&gt;airport.lisp

(METHOD DRAW AIRPORT)

&gt;saf&gt;bmi&gt;airport.lisp

Description: None

**2.3.1.5.6 ADD-AIRCRAFT**

## Definition 6

&gt;saf&gt;bmi&gt;presentation-types.lisp

Type: DEFINE-PRESENTATION-TO-COMMAND-TRANSLATOR

Arguments: ()

**Outputs:**

Calls: None

Called by: None

Description: None

**2.3.1.5.7 SANDBOX-OBJECT**

## Definition 7

&gt;saf&gt;bmi&gt;presentation-types.lisp

Type: DEFINE-PRESENTATION-TYPE

Arguments: ()

**Outputs:**

Calls: None

Called by: (WRITE-INSTANCE-VARIABLE (SETF SANDBOX-OBJECT)

SIMNET-AGENT SANDBOX-OBJECT)

No Source File Record

(READ-INSTANCE-VARIABLE SANDBOX-OBJECT SIMNET-AGENT

SANDBOX-OBJECT)

No Source File Record

(PRESENTATION-MOUSE-HANDLER SANDBOX-OBJECT-GESTURE)

No Source File Record

CREATE-STORED-INSTANCE

&gt;saf&gt;sys&gt;new-storage.lisp

```

FILTERED-SAVE-INSTANCE
>saf>sys>new-storage.lisp
MAKE-AGENT
>saf>simnet-objects>vehicle-tracking.lisp
(METHOD DISPLAY-TOTALS-PANE BMI)
>saf>bmi>bmi-frame.lisp
(METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)
>saf>bmi>bmi-frame.lisp
(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)
>saf>bmi>bmi-frame.lisp
DRAW-SANDBOX-OBJECT
>saf>sandbox>sandbox-object.lisp
COPY-SANDBOX-OBJECT
>saf>sandbox>sandbox-object.lisp
(PROPERTY SANDBOX-OBJECT NAMED-STRUCTURE-INVOKE)
No Source File Record
SANDBOX-OBJECT-P
>saf>sandbox>sandbox-object.lisp
MAKE-SANDBOX-OBJECT
>saf>sandbox>sandbox-object.lisp
FIND-FORMATION-INFO
>saf>sandbox>sandbox.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp

```

Description: None

### 2.3.1.5.8 SANDBOX-OBJECT-GESTURE

Definition 8

```

>saf>bmi>presentation-types.lisp
Type: DEFINE-PRESENTATION-ACTION
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

### 2.3.1.5.9 TACTICS

Definition 9

```

>saf>bmi>presentation-types.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: (WRITE-INSTANCE-VARIABLE (SETF TACTICS) SIMNET-AGENT
TACTICS)
No Source File Record
(READ-INSTANCE-VARIABLE TACTICS SIMNET-AGENT TACTICS)
No Source File Record
(METHOD UPDATE-APPEARANCE SIMNET-AGENT)
>saf>objects>simnet-agent.lisp

```

```

(METHOD GET-TEMPLATE SIMNET-AGENT)
>saf>objects>simnet-agent.lisp
(METHOD MAKE-UNIT-NAME SIMNET-NAME-MIXIN)
>saf>objects>simnet-name-mixin.lisp
(METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)
>saf>bmi>bmi-frame.lisp
(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)
>saf>bmi>bmi-frame.lisp
(DEFUN-IN-FLAVOR ACCEPT-TACTICS-AND-TEAM BMI)
No Source File Record
(PROPERTY TACTICS DEFTYPE)
No Source File Record
SIMNET-AGENT
>saf>objects>simnet-agent.lisp

```

Description: None

### 2.3.1.5.10 SIMNET-TEAM

Definition 10

```

>saf>bmi>presentation-types.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: (PROPERTY SIMNET-TEAM DEFTYPE)
No Source File Record

```

Description: None

### 2.3.1.5.11 MILS

Definition 11

```

>saf>bmi>presentation-types.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None
Called by: (METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)
>saf>bmi>bmi-frame.lisp
(PROPERTY MILS DEFTYPE)
No Source File Record

```

Description: None

### 2.3.1.5.12 BATTALION-BUMPER

Definition 12

```

>saf>bmi>presentation-types.lisp
Type: DEFINE-PRESENTATION-TYPE
Arguments: ()
Outputs:
Calls: None

```

Called by: (METHOD ACCEPT-BMI-OPTIONS BMI)  
>saf>bmi>bmi-frame.lisp  
(PROPERTY BATTALION-BUMPER DEFTYPE)  
No Source File Record

Description: None

### 2.3.1.5.13 COMPANY-BUMPER

Definition 13

>saf>bmi>presentation-types.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
(PROPERTY COMPANY-BUMPER DEFTYPE)  
No Source File Record

Description: None

### 2.3.1.5.14 PLATOON-BUMPER

Definition 14

>saf>bmi>presentation-types.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: (METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)  
>saf>bmi>bmi-frame.lisp  
(PROPERTY PLATOON-BUMPER DEFTYPE)  
No Source File Record

Description: None

## 2.3.2 Sandbox CSC

This CSC contains the code required to save away the units and positions which have been selected in battlemaster mode but not created. By means of this code a battlemaster can select and position his units without being connected to a simulation host. He can save these units away in a file for future use. This CSC contains the following CSUs:

```
sandbox>sandbox.lisp csu  
sandbox>sandbox-object.lisp csu  
sandbox>utilities.lisp csu
```

### 2.3.2.1 CSU sandbox>sandbox.lisp

This unit contains the sandbox data-structure and its associated functions.

The sandbox construct gives the battlemaster a place to put selected units, change their positions, save and retrieve them, etc., before actually creating them on the PVD.

Functions in this unit allow for copying, displaying, and saving sandbox structures.

Because data about the formation of a unit is not part of the SAF command protocol, this information is cached on the workstation when units are created on the Simhost. The way RUDP works, there is a period of time when a unit-creation message has been sent to the Simhost, but an acknowledgement has not yet been received; this is the interval during which formation data has to be cached. Once an acknowledgement is received, the unit entity is created on the Symbolics, and the formation data is recorded there; at this point the cached data is no longer needed. The functions *formation-cache-entry*, *cache-formation-data*, and *find-formation-info* allow formation data to be stored and retrieved as required.

#### 2.3.2.1.1 SANDBOX

Definition 1

```

>saf>sandbox>sandbox.lisp
Type: DEFSTRUCT
Arguments: ()
Outputs:
Calls: None
Called by: (METHOD COM-CREATE-UNITS-INTERNAL SAF)
           No Source File Record
           (METHOD COM-CLEAR-SELECTIONS-INTERNAL SAF)
           No Source File Record
           (METHOD MAKE-INSTANCE SAF AFTER)
>saf>ui>frame.lisp
COPY-SANDBOX
>saf>sandbox>sandbox.lisp
(PROPERTY SANDBOX NAMED-STRUCTURE-INVOKE)
No Source File Record
SANDBOX-P
>saf>sandbox>sandbox.lisp
MAKE-SANDBOX
>saf>sandbox>sandbox.lisp
Description: None

```

#### 2.3.2.1.2 COPY-SANDBOX

Definition 2

```

>saf>sandbox>sandbox.lisp
Type: Function
Arguments: (SANDBOX)
Outputs:

```

Calls: SANDBOX  
       >saf>sandbox>sandbox.lisp  
       COPY-SANDBOX-OBJECT  
       >saf>sandbox>sandbox-object.lisp  
 Called by: None  
 Description: None

### 2.3.2.1.3 DRAW-SANDBOX

Definition 3

      >saf>sandbox>sandbox.lisp  
 Type: Function  
 Arguments: (SANDBOX)  
 Outputs:  
 Calls: DRAW-SANDBOX-OBJECT  
       >saf>sandbox>sandbox-object.lisp  
 Called by: DRAW-MAP  
       >saf>sys>update-process.lisp  
       RETRIEVE-A-SANDBOX  
       >saf>bmi>utilities.lisp  
 Description: None

### 2.3.2.1.4 ERASE-SANDBOX

Definition 4

      >saf>sandbox>sandbox.lisp  
 Type: Function  
 Arguments: (SANDBOX)  
 Outputs:  
 Calls: ERASE-SANDBOX-OBJECT  
       >saf>sandbox>sandbox-object.lisp  
 Called by: (METHOD COM-CREATE-UNITS-INTERNAL SAF)  
           No Source File Record  
           (METHOD COM-CLEAR-SELECTIONS-INTERNAL SAF)  
           No Source File Record  
           RETRIEVE-A-SANDBOX  
           >saf>bmi>utilities.lisp  
 Description: None

### 2.3.2.1.5 STORE-SANDBOX

Definition 5

      >saf>sandbox>sandbox.lisp  
 Type: Function  
 Arguments: (SANDBOX)  
 Outputs:  
 Calls: \*ACTIVE-SANDBOXES\*  
       >saf>sys>vars.lisp  
       WRITE-SANDBOX  
       >saf>sandbox>sandbox.lisp  
 Called by: (METHOD COM-SAVE-SELECTIONS-INTERNAL SAF)  
           No Source File Record  
 Description: None

**2.3.2.1.6 WRITE-SANDBOX**

Definition 6

```

>saf>sandbox>sandbox.lisp
Type: Function
Arguments: (SANDBOX)
Outputs:
Calls: OPFOR-CHOOSE-VARIABLE-VALUES
       >saf>sys>cl-tv-patches.lisp
Called by: STORE-SANDBOX
          >saf>sandbox>sandbox.lisp
Description: None

```

**2.3.2.1.7 FORMATION-CACHE-ENTRY**

Definition 7

```

>saf>sandbox>sandbox.lisp
Type: DEFSTRUCT
Arguments: ()
Outputs:
Calls: None
Called by: CACHE-FORMATION-INFO
          >saf>sandbox>sandbox.lisp
          FORMATION-CACHE-ENTRY-P
          >saf>sandbox>sandbox.lisp
          COPY-FORMATION-CACHE-ENTRY
          >saf>sandbox>sandbox.lisp
          MAKE-FCE
          >saf>sandbox>sandbox.lisp
Description: None

```

**2.3.2.1.8 \*FORMATION-CACHE\***

Definition 8

```

>saf>sandbox>sandbox.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: FIND-FORMATION-INFO
          >saf>sandbox>sandbox.lisp
          CACHE-FORMATION-INFO
          >saf>sandbox>sandbox.lisp
Description: None

```

**2.3.2.1.9 \*DEBUG-FCE\***

Definition 9

```

>saf>sandbox>sandbox.lisp
Type: Variable
Arguments: ()

```



Outputs:  
Calls: None  
Called by: FIND-FORMATION-INFO  
>saf>sandbox>sandbox.lisp  
Description: None

### 2.3.2.1.10 CACHE-FORMATION-INFO

Definition 10

>saf>sandbox>sandbox.lisp  
Type: Function  
Arguments: (OBJECT)  
Outputs:  
Calls: VEH-MAIN-BATTLE-TANK  
>saf>network>vars.lisp  
VEH-PERSONNEL-CARRIER  
>saf>network>vars.lisp  
VEH-MORTAR-CARRIER  
>saf>network>vars.lisp  
VEH-SP-HOWITZER  
>saf>network>vars.lisp  
VEH-ATTACK-HELICOPTER  
>saf>network>vars.lisp  
VEH-FIGHTER-BOMBER  
>saf>network>vars.lisp  
VEH-ANTI-AIRCRAFT  
>saf>network>vars.lisp  
FORMATION-CACHE-ENTRY  
>saf>sandbox>sandbox.lisp  
\*FORMATION-CACHE\*  
>saf>sandbox>sandbox.lisp  
Called by: None  
Description: None

### 2.3.2.1.11 FIND-FORMATION-INFO

Definition 11

>saf>sandbox>sandbox.lisp  
Type: Function  
Arguments: (THING)  
Outputs:  
Calls: DISTANCE  
>map>utilities.lisp  
\*OPFOR-IO\*  
>saf>sys>vars.lisp  
SAY  
>saf>sys>macros.lisp  
\*FORMATION-CACHE\*  
>saf>sandbox>sandbox.lisp  
\*DEBUG-FCE\*  
>saf>sandbox>sandbox.lisp  
SANDBOX-OBJECT

```

>saf>bmi>presentation-types.lisp
FORMATION
>saf>cm>control-measure.lisp
SANDBOX-OBJECT
>saf>bmi>presentation-types.lisp
Called by:    None
Description:  None

```

### 2.3.2.2 CSU sandbox>sandbox-object.lisp

This unit contains the definition for the *sandbox-object* data-structure and its associated functions. A sandbox-object is a unit or vehicle that can be placed, copied, displayed, erased, etc. in a sandbox. Functions in this CSU provide those capabilities.

#### 2.3.2.2.1 SANDBOX-OBJECT

Definition 1

```

>saf>sandbox>sandbox-object.lisp
Type: DEFSTRUCT
Arguments: ()
Outputs:
Calls: None
Called by: (WRITE-INSTANCE-VARIABLE (SETF SANDBOX-OBJECT)
SIMNET-AGENT SANDBOX-OBJECT)
No Source File Record
(READ-INSTANCE-VARIABLE SANDBOX-OBJECT SIMNET-AGENT
SANDBOX-OBJECT)
No Source File Record
(PRESENTATION-MOUSE-HANDLER SANDBOX-OBJECT-GESTURE)
No Source File Record
CREATE-STORED-INSTANCE
>saf>sys>new-storage.lisp
FILTERED-SAVE-INSTANCE
>saf>sys>new-storage.lisp
MAKE-AGENT
>saf>simnet-objects>vehicle-tracking.lisp
(METHOD DISPLAY-TOTALS-PANE BMI)
>saf>bmi>bmi-frame.lisp
(METHOD MAKE-FWA-SANDBOX-OBJECT-INTERNAL BMI)
>saf>bmi>bmi-frame.lisp
(METHOD MAKE-SANDBOX-OBJECT-INTERNAL BMI)
>saf>bmi>bmi-frame.lisp
DRAW-SANDBOX-OBJECT
>saf>sandbox>sandbox-object.lisp
COPY-SANDBOX-OBJECT
>saf>sandbox>sandbox-object.lisp
(PROPERTY SANDBOX-OBJECT NAMED-STRUCTURE-INVOKE)
No Source File Record
SANDBOX-OBJECT-P
>saf>sandbox>sandbox-object.lisp

```

MAKE-SANDBOX-OBJECT  
>saf>sandbox>sandbox-object.lisp  
FIND-FORMATION-INFO  
>saf>sandbox>sandbox.lisp  
SIMNET-AGENT  
>saf>objects>simnet-agent.lisp

Description: None

### 2.3.2.2.2 COPY-SANDBOX-OBJECT

Definition 2

>saf>sandbox>sandbox-object.lisp  
Type: Function  
Arguments: (STRUCTURE)  
Outputs:  
Calls: SANDBOX-OBJECT  
>saf>bmi>presentation-types.lisp  
SANDBOX-OBJECT  
>saf>bmi>presentation-types.lisp  
Called by: COPY-SANDBOX  
>saf>sandbox>sandbox.lisp  
Description: None

### 2.3.2.2.3 SANDBOX-OBJECT-ALU

Definition 3

>saf>sandbox>sandbox-object.lisp  
Type: Function  
Arguments: (OBJECT)  
Outputs:  
Calls: ALIGNED-FOE  
>saf>sys>vars.lisp  
ALIGNED-OFFENSE  
>saf>sys>vars.lisp  
ALIGNED-DEFENSE  
>saf>sys>vars.lisp  
ALIGNED-FRIEND  
>saf>sys>vars.lisp  
ALIGNED-USSR  
>saf>sys>vars.lisp  
ALIGNED-US  
>saf>sys>vars.lisp  
\*FRIEND-ALLIANCE\*  
>saf>sys>vars.lisp  
\*FOE-ALLIANCE\*  
>saf>sys>vars.lisp  
\*OFFENSE-ALU\*  
>saf>sys>vars.lisp  
\*DEFENSE-ALU\*  
>saf>sys>vars.lisp

Called by: DRAW-SANDBOX-UNIT  
 >saf>sandbox>sandbox-object.lisp  
 DRAW-SANDBOX-OBJECT  
 >saf>sandbox>sandbox-object.lisp  
 Description: None

#### 2.3.2.2.4 SANDBOX-OBJECT-COUNTRY

Definition 4

>saf>sandbox>sandbox-object.lisp  
 Type: Function  
 Arguments: (OBJECT)  
 Outputs:  
 Calls: ALIGNED-FOE  
 >saf>sys>vars.lisp  
 ALIGNED-OFFENSE  
 >saf>sys>vars.lisp  
 ALIGNED-DEFENSE  
 >saf>sys>vars.lisp  
 ALIGNED-FRIEND  
 >saf>sys>vars.lisp  
 ALIGNED-USSR  
 >saf>sys>vars.lisp  
 ALIGNED-US  
 >saf>sys>vars.lisp  
 \*FRIEND-ALLIANCE\*  
 >saf>sys>vars.lisp  
 \*FOE-ALLIANCE\*  
 >saf>sys>vars.lisp  
 Called by: ERASE-SANDBOX-OBJECT  
 >saf>sandbox>sandbox-object.lisp  
 DRAW-SANDBOX-OBJECT  
 >saf>sandbox>sandbox-object.lisp  
 Description: None

#### 2.3.2.2.5 DRAW-SANDBOX-OBJECT

Definition 5

>saf>sandbox>sandbox-object.lisp  
 Type: Function  
 Arguments: (OBJECT &OPTIONAL (WINDOW \*PVD-DISPLAY\*) (SENSITIVE-  
 TYPE 'SANDBOX-OBJECT))  
 Outputs:  
 Calls: \*PVD-DISPLAY\*  
 >saf>sys>vars.lisp  
 \*TRIM-ALU\*  
 >saf>sys>vars.lisp  
 \*PAINT-VEHICLES-AS-ICONS\*  
 >saf>sys>vars.lisp  
 MAP-ÉCHELON-TYPE-TO-ICON  
 >saf>sys>interim-model.lisp

```

LOCAL
>saf>network>vars.lisp
VEHICLE
>saf>objects>vehicle.lisp
VEHICLE
>saf>objects>vehicle.lisp
DRAW-IMAGE
>saf>simnet-objects>draw-vehicles.lisp
IMAGE-FOR-VEHICLE
>saf>simnet-objects>draw-vehicles.lisp
DRAW-VEHICLE
>saf>simnet-objects>new-draw-vehicles.lisp
SAF
>saf>ui>frame.lisp
SANDBOX-OBJECT
>saf>bmi>presentation-types.lisp
SANDBOX-OBJECT-ALU
>saf>sandbox>sandbox-object.lisp
SANDBOX-OBJECT-COUNTRY
>saf>sandbox>sandbox-object.lisp
DRAW-SANDBOX-UNIT
>saf>sandbox>sandbox-object.lisp
SANDBOX-OBJECT
>saf>bmi>presentation-types.lisp
Called by: (METHOD COM-SELECT-UNITS-INTERNAL SAF)
No Source File Record
DRAW-SANDBOX
>saf>sandbox>sandbox.lisp
Description: None

```

### 2.3.2.2.6 ERASE-SANDBOX-OBJECT

Definition 6

```

>saf>sandbox>sandbox-object.lisp
Type: Function
Arguments: (OBJECT &OPTIONAL (WINDOW *PVD-DISPLAY*))
Outputs:
Calls: *PVD-DISPLAY*
>saf>sys>vars.lisp
*ERASE-VEHICLES-ALU*
>saf>sys>vars.lisp
*PAINT-VEHICLES-AS-ICONS*
>saf>sys>vars.lisp
MAP-ÉCHELON-TYPE-TO-ICON
>saf>sys>interim-model.lisp
LOCAL
>saf>network>vars.lisp
VEHICLE
>saf>objects>vehicle.lisp
VEHICLE
>saf>objects>vehicle.lisp
ERASE-IMAGE
>saf>simnet-objects>draw-vehicles.lisp

```

```

IMAGE-FOR-VEHICLE
>saf>simnet-objects>draw-vehicles.lisp
DRAW-VEHICLE
>saf>simnet-objects>new-draw-vehicles.lisp
SAF
>saf>ui>frame.lisp
SANDBOX-OBJECT-COUNTRY
>saf>sandbox>sandbox-object.lisp
DRAW-SANDBOX-UNIT
>saf>sandbox>sandbox-object.lisp
Called by: (METHOD BMI-REMOVE-SANDBOX-OBJECT BMI)
>saf>bmi>bmi-frame.lisp
ERASE-SANDBOX
>saf>sandbox>sandbox.lisp
Description: None

```

### 2.3.2.2.7 DRAW-SANDBOX-UNIT

Definition 7

```

>saf>sandbox>sandbox-object.lisp
Type: Function
Arguments: (OBJECT DRAW-OR-ERASE STREAM)
Outputs:
Calls: *ERASE-VEHICLES-ALU*
>saf>sys>vars.lisp
MAP-ECHOLON-TYPE-TO-ICON
>saf>sys>interim-model.lisp
DRAW-UNIT
>saf>simnet-objects>draw-units.lisp
SAF
>saf>ui>frame.lisp
SANDBOX-OBJECT-ALU
>saf>sandbox>sandbox-object.lisp
Called by: ERASE-SANDBOX-OBJECT
>saf>sandbox>sandbox-object.lisp
DRAW-SANDBOX-OBJECT
>saf>sandbox>sandbox-object.lisp
Description: None

```

### 2.3.2.3 CSU sandbox>utilities.lisp

This CSU contains several utility functions used by the sandbox code. Functions for menu presentation create lists of sandboxes to be displayed as menu items, such as the list of all active sandboxes, all sandboxes on disk, or all sandboxes.

Object creation no longer occurs in this file; sandbox objects are created by the constructor *make-sandbox-object*, which is called from other files, such as *bmi>bmi-frame.lisp*.

**2.3.2.3.1 ACTIVE SANDBOXES-AS-MENU-ITEMS**

## Definition 1

>saf>sandbox>utilities.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*ACTIVE-SANDBOXES\*  
>saf>sys>vars.lisp  
Called by: ALL-SANDBOXES-AS-MENU-ITEMS  
>saf>sandbox>utilities.lisp  
Description: None

**2.3.2.3.2 NAMES-OF-DISK-SANDBOXES**

## Definition 2

>saf>sandbox>utilities.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: NAME  
>saf>sysdcl.lisp  
Called by: ALL-SANDBOXES-AS-MENU-ITEMS  
>saf>sandbox>utilities.lisp  
Description: None

**2.3.2.3.3 SYMBOL-VS-CAR-LIST-TEST**

## Definition 3

>saf>sandbox>utilities.lisp  
Type: Function  
Arguments: (A B)  
Outputs:  
Calls: None  
Called by: ALL-SANDBOXES-AS-MENU-ITEMS  
>saf>sandbox>utilities.lisp  
Description: None

**2.3.2.3.4 ALL-SANDBOXES-AS-MENU-ITEMS**

## Definition 4

>saf>sandbox>utilities.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: ACTIVE-SANDBOXES-AS-MENU-ITEMS  
>saf>sandbox>utilities.lisp  
NAMES-OF-DISK-SANDBOXES  
>saf>sandbox>utilities.lisp  
SYMBOL-VS-CAR-LIST-TEST  
>saf>sandbox>utilities.lisp

Called by: RETRIEVE-A-SANDBOX  
 >saf>bmi>utilities.lisp  
 Description: None

### 2.3.2.3.5 'GET-LOCATION-AND-BEARING Definition 5

>saf>sandbox>utilities.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

### 2.3.2.3.6 GET-LOCATION-AND-BEARING Definition 6

>saf>sandbox>utilities.lisp  
 Type: Function  
 Arguments: (STREAM)  
 Outputs:  
 Calls: WITH-INTEGER-CONVERSION-MODE  
 >map>utilities.lisp  
 WITH-MAP-GRAPHICS  
 >map>utilities.lisp  
 SCREEN-TO-WORLD  
 >map>utilities.lisp  
 $\pi$   
 >saf>sys>constants.lisp  
 $2\pi$   
 >saf>sys>constants.lisp  
 90DEG  
 >saf>sys>constants.lisp  
 RAD-TO-MIL  
 >saf>sys>constants.lisp  
 RADIANS-COMPASS-TO-RADIANS-MATH  
 >saf>sys>macros.lisp  
 RADIANS-MATH-TO-RADIANS-COMPASS  
 >saf>sys>macros.lisp  
 COMPASS-ANGLE  
 >saf>sys>macros.lisp  
 MATH-ANGLE  
 >saf>sys>macros.lisp  
 RADIANS-COMPASS-TO-MILS  
 >saf>sys>macros.lisp  
 RADIANS-MATH-TO-MILS  
 >saf>sys>macros.lisp  
 Called by: (METHOD COM-SELECT-UNITS-INTERNAL SAF)  
 No Source File Record  
 Description: None



### 2.3.3 scenario CSC

This CSC contains the code which allow scenarios to be stored and retrieve at any time during an exercise. The CSUs in this CSC are:

```
objects>storable-mixin.lisp csu
sys>new-storage.lisp csu
```

#### 2.3.3.1 CSU objects>storable-mixin.lisp

This csu defines a mixin called *storable-mixin*. This flavor object defines an "instance-name" slot that is used by the storage facility *save-top-level-and-inferiors*, defined in *sys>new-storage.lisp*. For an object to be storable in this way, it must inherit the instance-name slot from *storable-mixin* in the flavor hierarchy. For more information on mixins in the flavor hierarchy, see the Symbolics manuals.

##### 2.3.3.1.1 STORABLE-MIXIN

Definition 1

```
>saf>objects>storable-mixin.lisp
Type: DEFOBJECT
Arguments: ()
Outputs:
Calls: None
Called by: SCENARIO
>saf>sys>new-storage.lisp
SUB-TASK
>saf>ui>subordinate-tasking.lisp
UNIT-TASK
>saf>ui>subordinate-tasking.lisp
OVERLAY
>saf>cm>overlay.lisp
CONTROL-MEASURE-POINT
>saf>cm>control-measure-point.lisp
ROUTE-POINT
>saf>cm>route-point.lisp
CONTROL-MEASURE-BEHAVIOR
>saf>cm>control-measure.lisp
ZONE-BEHAVIOR
>saf>cm>zone.lisp
AREA-BEHAVIOR
>saf>cm>area.lisp
LINE-BEHAVIOR
>saf>cm>line.lisp
CM-POINT-BEHAVIOR
>saf>cm>point.lisp
ROUTE-BEHAVIOR
>saf>cm>route.lisp
CONTROL-MEASURE
>saf>cm>control-measure.lisp
GENERIC-AREA
>saf>cm>generic-area.lisp
```

```

ZONE
>saf>cm>zone.lisp
AREA
>saf>cm>area.lisp
LINE
>saf>cm>line.lisp
CM-POINT
>saf>cm>point.lisp
ROUTE
>saf>cm>route.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
VEHICLE
>saf>objects>vehicle.lisp
COMPOSITE-OBJECT
>saf>objects>composite-object.lisp
SAVE-INSTANCE
>saf>sys>new-storage.lisp
MAKE-OBJECT-LIST-RECURSIVE
>saf>sys>new-storage.lisp

```

Description: None

### 2.3.3.2 CSU `sys>new-storage.lisp`

This unit contains code for saving certain kinds of lisp objects in a text file. It is used primarily to store and retrieve SAF unit and control measures information.

Symbolics provides a function that saves lisp objects in a *binary* file. The storage code in this CSU saves objects in a *text* file that can be conveniently edited when necessary.

The basic save function is called *save-top-level-and-inferiors*. It has been designed to handle, in a systematic way, the fact that objects to be saved often point to other objects, which must also be saved. Not only that, but the graph of pointer-connections usually contains closed cycles, in the intended application. Because of these cycles, a standard nested-list print representation of the objects may not exist, so a simpler storage approach based on the Lisp reader can't be used.

The storage function starts by a recursive descent, starting from given top-level objects, to find all the objects pointed to; this happens in the function *make-object-list-recursive*.

To simplify the storage and retrieval processes, the pointer links (the values in the slot-value pairs of the flavor instances to be stored) are replaced, before storage, by *instance-names*, standardized names, created using the function *return-iterated-symbol*, that serve as tags to identify the flavor-instances pointed to. A hash table is created linking the instances with their instance names, to make the replacement process fast. The instance's own instance-name is written into a special slot it inherits from the flavor mixin *storable-mixin*, defined in `objects>storable-mixin.lisp`. Instance-names have the form DB-INSTANCE-[number], to make them easy to spot in the text file.

On retrieval, instances are created first in memory with no pointer links, and a hash table linking instance-names back to instances is built. When all the instances have been made, the hash table is used to replace each instance-name by an actual value, reconstructing the original lisp pointer structure.

Objects stored are restricted to be flavor instances, and must mixin *storable-mixin*, which provides a slot to record the instance's instance-name before it's written to disk. Other restrictions on objects to be stored are described in the source-file's comment header.

### 2.3.3.2.1 \*DBASE-FILE\*

Definition 1

```
>saf>sys>new-storage.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: READ-AND-MAKE-INSTANCES
>saf>sys>new-storage.lisp
SAVE-IN-DATABASE
>saf>sys>new-storage.lisp
SAVE-TOP-LEVEL-AND-INFERIORS
>saf>sys>new-storage.lisp
Description: None
```

### 2.3.3.2.2 COERCE-STRING

Definition 2

```
>saf>sys>new-storage.lisp
Type: Function
Arguments: (X)
Outputs:
Calls: None
Called by: MKATOM
>saf>sys>new-storage.lisp
Description: None
```

### 2.3.3.2.3 MKATOM

Definition 3

```
>saf>sys>new-storage.lisp
Type: Function
Arguments: (ITEM)
Outputs:
Calls: SAF
>saf>ui>frame.lisp
COERCE-STRING
>saf>sys>new-storage.lisp
Called by: (PRESENTATION-MOUSE-HANDLER INSERT-POINT-1)
No Source File Record
(PRESENTATION-MOUSE-HANDLER INSERT-POINT)
No Source File Record
REMOVE-THIS-WINDOW-FROM-THE-CONFIGURATION
>saf>interface>object-menu.lisp
TRUNCATE-IF-NECESSARY
>saf>interface>object-menu.lisp
```

```
UNDO-ALL-CHANGES
>saf>interface>object-menu.lisp
(METHOD UNDO-LAST-CHANGE-INTERNAL OBJECT-MS-PANE)
>saf>interface>object-menu.lisp
(METHOD SAVE-ALL-OBJECT-INFORMATION OBJECT-MS-PANE)
>saf>interface>object-menu.lisp
WRITE-DIRECT-FIRE-DAMAGE-DATA-FILE
>saf>interface>model-menu.lisp
WRITE-INDIRECT-FIRE-DAMAGE-DATA-FILE
>saf>interface>model-menu.lisp
MAKE-Dictionary-PAIRS
>saf>interface>model-menu.lisp
(METHOD ALTER-PROBABILITY POINT)
>saf>interface>model-menu.lisp
(METHOD ALTER-RANGE POINT)
>saf>interface>model-menu.lisp
(METHOD DRAG-UP-DOWN POINT)
>saf>interface>model-menu.lisp
SELECT-AND-DRAG-UP-DOWN-POINT
>saf>interface>model-menu.lisp
GO-BACK-TO-PREVIOUS-STEP
>saf>interface>model-menu.lisp
UPDATE-BACKTRACKING-CAPABILITY
>saf>interface>model-menu.lisp
SELECT-AND-DRAG-POINT
>saf>interface>model-menu.lisp
GET-POINT
>saf>interface>model-menu.lisp
(METHOD ERASE-POINT-AND-LINES POINT)
>saf>interface>model-menu.lisp
(METHOD DRAG POINT)
>saf>interface>model-menu.lisp
ADD-NEW-POINT
>saf>interface>model-menu.lisp
RECORD-NEW-POINT
>saf>interface>model-menu.lisp
ADD-CORRESPONDING-NEW-POINT-IN-POINT-LIST
>saf>interface>model-menu.lisp
UPDATE-POINT-LIST
>saf>interface>model-menu.lisp
FIND-SURROUNDING-POINTS
>saf>interface>model-menu.lisp
DELETE-POINT-IF-THERE
>saf>interface>model-menu.lisp
(METHOD EXPUNGE POINT)
>saf>interface>model-menu.lisp
DRAW-X-TICKS
>saf>interface>model-menu.lisp
GET-CURRENT-LINE-POINTS
>saf>interface>model-menu.lisp
GET-CURRENT-GRAPH-POINTS
>saf>interface>model-menu.lisp
LOAD-SCENARIO
>saf>sys>new-storage.lisp
```

**LOAD-OVERLAY**

>saf>sys>new-storage.lisp  
 GIMME-VAR-NAME-OF-CURRENT-POINTS  
 >saf>interface>model-menu.lisp  
 GIMME-VAR-NAME-OF-CURRENT-GRAPH  
 >saf>interface>model-menu.lisp

Description: None

**2.3.3.2.4 GET-INSTANCE-VARIABLES**

Definition 4

>saf>sys>new-storage.lisp  
 Type: Macro  
 Arguments: (INSTANCE)  
 Outputs:  
 Calls: GET-INSTANCE-VARIABLES  
 >saf>sys>new-storage.lisp  
 Called by: REMOVE-LEFTOVER-INSTANCE-NAMES  
 >saf>sys>new-storage.lisp  
 READ-AND-MAKE-INSTANCES  
 >saf>sys>new-storage.lisp  
 SAVE-INSTANCE  
 >saf>sys>new-storage.lisp  
 MAKE-OBJECT-LIST-RECURSIVE  
 >saf>sys>new-storage.lisp  
 GET-INSTANCE-VARIABLES  
 >saf>sys>new-storage.lisp

Description: None

**2.3.3.2.5 ITERATED-SYMBOL**

Definition 5

>saf>sys>new-storage.lisp  
 Type: Subst  
 Arguments: (NUM)  
 Outputs:  
 Calls: None  
 Called by: RETURN-ITERATED-SYMBOL  
 >saf>sys>new-storage.lisp

Description: None

**2.3.3.2.6 RETURN-ITERATED-SYMBOL**

Definition 6

>saf>sys>new-storage.lisp  
 Type: Function  
 Arguments: (NUM)  
 Outputs:  
 Calls: ITERATED-SYMBOL  
 >saf>sys>new-storage.lisp

Called by: SAVE-IN-DATABASE  
>saf>sys>new-storage.lisp  
Description: None

### 2.3.3.2.7 GET-VALUE-SUBST

Definition 7

>saf>sys>new-storage.lisp  
Type: Subst  
Arguments: (SLOT INSTANCE)  
Outputs:  
Calls: None  
Called by: GET-VALUE  
>saf>sys>new-storage.lisp  
Description: None

### 2.3.3.2.8 GET-VALUE

Definition 8

>saf>sys>new-storage.lisp  
Type: Function  
Arguments: (SLOT INSTANCE)  
Outputs:  
Calls: GET-VALUE-SUBST  
>saf>sys>new-storage.lisp  
Called by: REMOVE-LEFTOVER-INSTANCE-NAMES  
>saf>sys>new-storage.lisp  
READ-AND-MAKE-INSTANCES  
>saf>sys>new-storage.lisp  
SAVE-INSTANCE  
>saf>sys>new-storage.lisp  
MAKE-OBJECT-LIST-RECURSIVE  
>saf>sys>new-storage.lisp  
Description: None

### 2.3.3.2.9 REPLACE-VALUE-SUBST

Definition 9

>saf>sys>new-storage.lisp  
Type: Subst  
Arguments: (SLOT INSTANCE VALUE)  
Outputs:  
Calls: None  
Called by: REPLACE-VALUE  
>saf>sys>new-storage.lisp  
Description: None

**2.3.3.2.10 REPLACE-VALUE**

Definition 10

```

>saf>sys>new-storage.lisp
Type: Function
Arguments: (SLOT INSTANCE VALUE)
Outputs:
Calls: REPLACE-VALUE-SUBST
       >saf>sys>new-storage.lisp
Called by: REMOVE-LEFTOVER-INSTANCE-NAMES
          >saf>sys>new-storage.lisp
          READ-AND-MAKE-INSTANCES
          >saf>sys>new-storage.lisp
          SAVE-IN-DATABASE
          >saf>sys>new-storage.lisp
Description: None

```

**2.3.3.2.11 SAVE-TOP-LEVEL-AND-INFERIORS**

Definition 11

```

>saf>sys>new-storage.lisp
Type: Function
Arguments: (LIST-OF-TOP-LEVEL-OBJECTS &OPTIONAL (FILENAME *DBASE-
FILE*))
Outputs:
Calls: *DBASE-FILE*
       >saf>sys>new-storage.lisp
       MAKE-OBJECT-LIST-RECURSIVE
       >saf>sys>new-storage.lisp
       SAVE-IN-DATABASE
       >saf>sys>new-storage.lisp
Called by: (METHOD STORE SCENARIO)
          >saf>sys>new-storage.lisp
Description: None

```

**2.3.3.2.12 MAKE-OBJECT-LIST-RECURSIVE**

Definition 12

```

>saf>sys>new-storage.lisp
Type: Function
Arguments: (STUFF-TO-SAVE OBJECT-LIST)
Outputs:
Calls: STORABLE-MIXIN
       >saf>objects>storable-mixin.lisp
       GET-INSTANCE-VARIABLES
       >saf>sys>new-storage.lisp
       GET-VALUE
       >saf>sys>new-storage.lisp
       MAKE-OBJECT-LIST-RECURSIVE
       >saf>sys>new-storage.lisp

```

Called by: MAKE-OBJECT-LIST-RECURSIVE

>saf>sys>new-storage.lisp  
 SAVE-TOP-LEVEL-AND-INFERIORS  
 >saf>sys>new-storage.lisp

Description: None

### 2.3.3.2.13 SAVE-IN-DATABASE

Definition 13

>saf>sys>new-storage.lisp

Type: Function

Arguments: (LIST-OF-INSTANCES &OPTIONAL (FILENAME \*DBASE-FILE\*))

Outputs:

Calls: \*DBASE-FILE\*

>saf>sys>new-storage.lisp  
 RETURN-ITERATED-SYMBOL  
 >saf>sys>new-storage.lisp  
 REPLACE-VALUE  
 >saf>sys>new-storage.lisp  
 FILTERED-SAVE-INSTANCE  
 >saf>sys>new-storage.lisp

Called by: SAVE-TOP-LEVEL-AND-INFERIORS

>saf>sys>new-storage.lisp

Description: None

### 2.3.3.2.14 SAVE-INSTANCE

Definition 14

>saf>sys>new-storage.lisp

Type: Function

Arguments: (INSTANCE STREAM INST-HASH-TABLE)

Outputs:

Calls: STORABLE-MIXIN

>saf>objects>storable-mixin.lisp  
 GET-INSTANCE-VARIABLES  
 >saf>sys>new-storage.lisp  
 GET-VALUE  
 >saf>sys>new-storage.lisp  
 REPLACE-SLOT-VALUE-OBJECTS  
 >saf>sys>new-storage.lisp

Called by: FILTERED-SAVE-INSTANCE

>saf>sys>new-storage.lisp

Description: None

### 2.3.3.2.15 REPLACE-SLOT-VALUE-OBJECTS

Definition 15

>saf>sys>new-storage.lisp

Type: Function

Arguments: (SLOT-VALUE INST-HASH-TABLE)

Outputs:



Calls: REPLACE-SLOT-VALUE-OBJECTS  
       >saf>sys>new-storage.lisp  
 Called by: REPLACE-SLOT-VALUE-OBJECTS  
           >saf>sys>new-storage.lisp  
           SAVE-INSTANCE  
           >saf>sys>new-storage.lisp  
 Description: None

### 2.3.3.2.16 READ-AND-MAKE-INSTANCES

Definition 16

      >saf>sys>new-storage.lisp  
 Type: Function  
 Arguments: (&OPTIONAL (FILENAME \*DBASE-FILE\*))  
 Outputs:  
 Calls: \*DB-INSTANCES\*  
       >saf>sys>vars.lisp  
       SAF  
       >saf>ui>frame.lisp  
       \*DBASE-FILE\*  
       >saf>sys>new-storage.lisp  
       GET-INSTANCE-VARIABLES  
       >saf>sys>new-storage.lisp  
       GET-VALUE  
       >saf>sys>new-storage.lisp  
       REPLACE-VALUE  
       >saf>sys>new-storage.lisp  
       REPLACE-SLOT-VALUE-INSTANCE-NAMES  
       >saf>sys>new-storage.lisp  
       REMOVE-LEFTOVER-DB-INSTANCES  
       >saf>sys>new-storage.lisp  
 Called by: LOAD-SCENARIO  
           >saf>sys>new-storage.lisp  
           LOAD-OVERLAY  
           >saf>sys>new-storage.lisp  
 Description: None

### 2.3.3.2.17 REPLACE-SLOT-VALUE-INSTANCE-NAMES

Definition 17

      >saf>sys>new-storage.lisp  
 Type: Function  
 Arguments: (SLOT-VALUE INST-HASH-TABLE)  
 Outputs:  
 Calls: REPLACE-SLOT-VALUE-INSTANCE-NAMES  
       >saf>sys>new-storage.lisp  
 Called by: REPLACE-SLOT-VALUE-INSTANCE-NAMES  
           >saf>sys>new-storage.lisp  
           READ-AND-MAKE-INSTANCES  
           >saf>sys>new-storage.lisp  
 Description: None

**2.3.3.2.18 REMOVE-LEFTOVER-SLOT-VALUE-INSTANCE-NAMES**

Definition 18

>saf>sys>new-storage.lisp  
 Type: Function  
 Arguments: (SLOT-VALUE)  
 Outputs:  
 Calls: REMOVE-LEFTOVER-SLOT-VALUE-INSTANCE-NAMES  
       >saf>sys>new-storage.lisp  
 Called by: REMOVE-LEFTOVER-INSTANCE-NAMES  
           >saf>sys>new-storage.lisp  
           REMOVE-LEFTOVER-SLOT-VALUE-INSTANCE-NAMES  
           >saf>sys>new-storage.lisp  
 Description: None

**2.3.3.2.19 REMOVE-LEFTOVER-INSTANCE-NAMES**

Definition 19

>saf>sys>new-storage.lisp  
 Type: Function  
 Arguments: (INSTANCE)  
 Outputs:  
 Calls: GET-INSTANCE-VARIABLES  
       >saf>sys>new-storage.lisp  
       GET-VALUE  
       >saf>sys>new-storage.lisp  
       REPLACE-VALUE  
       >saf>sys>new-storage.lisp  
       REMOVE-LEFTOVER-SLOT-VALUE-INSTANCE-NAMES  
       >saf>sys>new-storage.lisp  
 Called by: REMOVE-LEFTOVER-DB-INSTANCES  
           >saf>sys>new-storage.lisp  
 Description: None

**2.3.3.2.20 REMOVE-LEFTOVER-DB-INSTANCES**

Definition 20

>saf>sys>new-storage.lisp  
 Type: Function  
 Arguments: (&OPTIONAL (INST-LIST \*DB-INSTANCES\*))  
 Outputs:  
 Calls: \*DB-INSTANCES\*  
       >saf>sys>vars.lisp  
       REMOVE-LEFTOVER-INSTANCE-NAMES  
       >saf>sys>new-storage.lisp  
 Called by: READ-AND-MAKE-INSTANCES  
           >saf>sys>new-storage.lisp  
 Description: None

**2.3.3.2.21 \*SCENARIO\***

Definition 21

>saf>sys>new-storage.lisp  
Type: Parameter  
Arguments: ()  
Outputs:  
Calls: None  
Called by: STORE-SCENARIO  
>saf>sys>new-storage.lisp  
Description: None

**2.3.3.2.22 \*SAVE-INSTANCE-FILTER\***

Definition 22

>saf>sys>new-storage.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: RETURN-SCENARIO-OBJECT-LIST  
>saf>sys>new-storage.lisp  
STORE-SCENARIO  
>saf>sys>new-storage.lisp  
NAME-AND-STORE-SCENARIO  
>saf>sys>new-storage.lisp  
NAME-AND-STORE-OVERLAY  
>saf>sys>new-storage.lisp  
FILTERED-SAVE-INSTANCE  
>saf>sys>new-storage.lisp  
Description: None

**2.3.3.2.23 SAVE-FOR-TASKING-P**

Definition 23

>saf>sys>new-storage.lisp  
Type: Function  
Arguments: (AGENT-INST)  
Outputs:  
Calls: SIMNET-AGENT  
>saf>objects>simnet-agent.lisp  
SIMNET-AGENT  
>saf>objects>simnet-agent.lisp  
SIMNET-AGENT  
>saf>objects>simnet-agent.lisp  
\*TOP-LEVEL-TASKING\*  
>saf>ui>subordinate-tasking.lisp  
UNIT-TASK  
>saf>ui>subordinate-tasking.lisp  
UNIT  
>saf>cm>control-measure.lisp

Called by: FILTERED-SAVE-INSTANCE  
 >saf>sys>new-storage.lisp  
 Description: None

#### 2.3.3.2.24 FILTERED-SAVE-INSTANCE

Definition 24

>saf>sys>new-storage.lisp  
 Type: Function  
 Arguments: (INSTANCE STREAM INST-HASH-TABLE LIST-OF-INSTANCES)  
 Outputs:  
 Calls: SIMNET-AGENT  
 >saf>objects>simnet-agent.lisp  
 SIMNET-AGENT  
 >saf>objects>simnet-agent.lisp  
 SIMNET-AGENT  
 >saf>objects>simnet-agent.lisp  
 UNIT-TASK  
 >saf>ui>subordinate-tasking.lisp  
 SUB-TASK  
 >saf>ui>subordinate-tasking.lisp  
 SAVE-INSTANCE  
 >saf>sys>new-storage.lisp  
 \*SAVE-INSTANCE-FILTER\*  
 >saf>sys>new-storage.lisp  
 SAVE-FOR-TASKING-P  
 >saf>sys>new-storage.lisp  
 SANDBOX-OBJECT  
 >saf>bmi>presentation-types.lisp  
 CONTROL-MEASURE-BEHAVIOR  
 >saf>cm>control-measure.lisp  
 SANDBOX-OBJECT  
 >saf>bmi>presentation-types.lisp  
 Called by: SAVE-IN-DATABASE  
 >saf>sys>new-storage.lisp  
 Description: None

#### 2.3.3.2.25 CONCATLIST

Definition 25

>saf>sys>new-storage.lisp  
 Type: Subst  
 Arguments: (ARGS)  
 Outputs:  
 Calls: None  
 Called by: CONCAT  
 >saf>sys>new-storage.lisp  
 Description: None

**2.3.3.2.26 CONCAT**

## Definition 26

```

>saf>sys>new-storage.lisp
Type: Function
Arguments: (&REST ARGS)
Outputs:
Calls: CONCATLIST
>saf>sys>new-storage.lisp
Called by: (PRESENTATION-MOUSE-HANDLER INSERT-POINT-1)
No Source File Record
(PRESENTATION-MOUSE-HANDLER INSERT-POINT)
No Source File Record
(METHOD DRAW-ALTITUDE FORMATION-OBJECT)
>saf>interface>formations.lisp
(METHOD DRAW FORMATION-OBJECT)
>saf>interface>formations.lisp
REMOVE-THIS-WINDOW-FROM-THE-CONFIGURATION
>saf>interface>object-menu.lisp
TRUNCATE-IF-NECESSARY
>saf>interface>object-menu.lisp
FIND-FIELD-DESCRIPTOR
>saf>interface>object-menu.lisp
READ-OBJECT-FILE
>saf>interface>object-menu.lisp
FETCH-DF-DATA
>saf>interface>model-menu.lisp
READ-DIRECT-FIRE-DAMAGE-DATA
>saf>interface>model-menu.lisp
FETCH-IF-DATA
>saf>interface>model-menu.lisp
READ-INDIRECT-FIRE-DAMAGE-DATA
>saf>interface>model-menu.lisp
MAKE-DICTIONARY-PAIRS
>saf>interface>model-menu.lisp
(METHOD ALTER-PROBABILITY POINT)
>saf>interface>model-menu.lisp
(METHOD ALTER-RANGE POINT)
>saf>interface>model-menu.lisp
(METHOD DRAG-UP-DOWN POINT)
>saf>interface>model-menu.lisp
SELECT-AND-DRAG-UP-DOWN-POINT
>saf>interface>model-menu.lisp
SELECT-HOST
>saf>interface>model-menu.lisp
SWITCH-HIGHLIGHT
>saf>interface>model-menu.lisp
GO-BACK-TO-PREVIOUS-STEP
>saf>interface>model-menu.lisp
UPDATE-BACKTRACKING-CAPABILITY
>saf>interface>model-menu.lisp
SELECT-AND-DRAG-POINT
>saf>interface>model-menu.lisp
GET-POINT

```

```

>saf>interface>model-menu.lisp
(METHOD ERASE-POINT-AND-LINES POINT)
>saf>interface>model-menu.lisp
(METHOD DRAG POINT)
>saf>interface>model-menu.lisp
ADD-NEW-POINT
>saf>interface>model-menu.lisp
RECORD-NEW-POINT
>saf>interface>model-menu.lisp
ADD-CORRESPONDING-NEW-POINT-IN-POINT-LIST
>saf>interface>model-menu.lisp
UPDATE-POINT-LIST
>saf>interface>model-menu.lisp
FIND-SURROUNDING-POINTS
>saf>interface>model-menu.lisp
DELETE-POINT-IF-THERE
>saf>interface>model-menu.lisp
(METHOD EXPUNGE POINT)
>saf>interface>model-menu.lisp
DRAW-X-TICKS
>saf>interface>model-menu.lisp
GET-CURRENT-LINE-POINTS
>saf>interface>model-menu.lisp
GET-CURRENT-GRAPH-POINTS
>saf>interface>model-menu.lisp
(METHOD STORE SCENARIO)
>saf>sys>new-storage.lisp
GIMME-VAR-NAME-OF-CURRENT-POINTS
>saf>interface>model-menu.lisp
GIMME-VAR-NAME-OF-CURRENT-GRAPH
>saf>interface>model-menu.lisp

```

Description: None

### 2.3.3.2.27 SCENARIO

Definition 27

```

>saf>sys>new-storage.lisp
Type: Flavor
Arguments: ()
Outputs:
Calls: STORABLE-MIXIN
>saf>objects>storable-mixin.lisp
Called by: None
Description: None

```

### 2.3.3.2.28 CLOSE-ENOUGH

Definition 28

```

>saf>sys>new-storage.lisp
Type: Function
Arguments: (GOAL ATTEMPT &OPTIONAL (TOLERANCE 0.1))
Outputs:

```

Calls: None  
 Called by: (METHOD ADJUST-VIEWPORT SCENARIO)  
           >saf>sys>new-storage.lisp  
 Description: used to compare numbers when equality is too strict

### 2.3.3.2.29 (METHOD ADJUST-VIEWPORT SCENARIO)

Definition 29

```

    >saf>sys>new-storage.lisp
  Type: Method
  Arguments: ()
  Outputs:
  Calls: *ZOOM-LEVELS*
         >map>zoom-levels.lisp
         *ZOOM-LEVELS*
         >map>zoom-levels.lisp
         *PVD-DISPLAY*
         >saf>sys>vars.lisp
         *INTERFACE-TO-UPDATE-PROCESS-QUEUE*
         >saf>sys>vars.lisp
         ADD-TO-UPDATE-QUEUE
         >saf>sys>macros.lisp
         CLOSE-ENOUGH
         >saf>sys>new-storage.lisp
  Called by: None
  Description: None
  
```

### 2.3.3.2.30 GET-SCREEN-PARAMETERS

Definition 30

```

    >saf>sys>new-storage.lisp
  Type: Function
  Arguments: ()
  Outputs:
  Calls: *PVD-DISPLAY*
         >saf>sys>vars.lisp
  Called by: STORE-SCENARIO
         >saf>sys>new-storage.lisp
  Description: None
  
```

### 2.3.3.2.31 \*OVERLAY-TO-SAVE\*

Definition 31

```

    >saf>sys>new-storage.lisp
  Type: Variable
  Arguments: ()
  Outputs:
  Calls: None
  
```

Called by: RETURN-SCENARIO-OBJECT-LIST

```
>saf>sys>new-storage.lisp
NAME-AND-STORE-OVERLAY
>saf>sys>new-storage.lisp
```

Description: None

### 2.3.3.2.32 NAME-AND-STORE-OVERLAY

Definition 32

```
>saf>sys>new-storage.lisp
Type: Function
Arguments: ()
Outputs:
>saf>sys>vars.lisp
*ALL-OVERLAYS*
>saf>sys>vars.lisp
SAY
>saf>sys>macros.lisp
*SAVE-INSTANCE-FILTER*
>saf>sys>new-storage.lisp
*OVERLAY-TO-SAVE*
>saf>sys>new-storage.lisp
REMOVE-DOTS-FROM-STRING
>saf>sys>new-storage.lisp
STORE-SCENARIO
>saf>sys>new-storage.lisp
```

Called by: SAVE-OR-LOAD-OVERLAYS

```
>saf>sys>new-storage.lisp
```

Description: None

### 2.3.3.2.33 SAVE-OR-LOAD-OVERLAYS

Definition 33

```
>saf>sys>new-storage.lisp
Type: Function
Arguments: ()
Outputs:
Calls: NAME-AND-STORE-OVERLAY
>saf>sys>new-storage.lisp
LOAD-OVERLAY
>saf>sys>new-storage.lisp
```

Called by: (METHOD COM-SELECT-BUTTON-INTERNAL SAF)

No Source File Record

Description: None

### 2.3.3.2.34 NAME-AND-STORE-SCENARIO

Definition 34

```
>saf>sys>new-storage.lisp
Type: Function
Arguments: ()
```



## Outputs:

Calls: \*OPFOR-IO\*

&gt;saf&gt;sys&gt;vars.lisp

SAY

&gt;saf&gt;sys&gt;macros.lisp

\*SAVE-INSTANCE-FILTER\*

&gt;saf&gt;sys&gt;new-storage.lisp

REMOVE-DOTS-FROM-STRING

&gt;saf&gt;sys&gt;new-storage.lisp

STORE-SCENARIO

&gt;saf&gt;sys&gt;new-storage.lisp

Called by: (METHOD COM-SELECT-BUTTON-INTERNAL SAF)

No Source File Record

COM-STORE-SCENARIO

&gt;saf&gt;ui&gt;commands.lisp

COM-SAVE-SCENARIO

&gt;saf&gt;ui&gt;commands.lisp

Description: None

**2.3.3.2.35 REMOVE-DOTS-FROM-STRING**

Definition 35

&gt;saf&gt;sys&gt;new-storage.lisp

Type: Function

Arguments: (STRING)

Outputs:

Calls: REMOVE-DOTS-FROM-STRING

&gt;saf&gt;sys&gt;new-storage.lisp

Called by: REMOVE-DOTS-FROM-STRING

&gt;saf&gt;sys&gt;new-storage.lisp

NAME-AND-STORE-SCENARIO

&gt;saf&gt;sys&gt;new-storage.lisp

NAME-AND-STORE-OVERLAY

&gt;saf&gt;sys&gt;new-storage.lisp

Description: None

**2.3.3.2.36 STORE-SCENARIO**

Definition 36

&gt;saf&gt;sys&gt;new-storage.lisp

Type: Function

Arguments: (NAME &amp;KEY (FILENAME NIL))

Outputs:

Calls: NAME

&gt;saf&gt;sysdcl.lisp

\*BMI-PROGRAM\*

&gt;saf&gt;sys&gt;vars.lisp

\*DB-INSTANCES\*

&gt;saf&gt;sys&gt;vars.lisp

\*REAPPEAR-LATENCY\*

&gt;saf&gt;rudp&gt;vars.lisp

\*RANGE-THRESHOLD\*

```

>saf>rudp>vars.lisp
*UPDATE-RATE*
>saf>rudp>vars.lisp
*CLUSTER-DISTANCE*
>saf>rudp>vars.lisp
*DEFAULT-BATTALION-NUMBER*
>saf>bmi>bmi-frame.lisp
*TOP-LEVEL-TASKING*
>saf>ui>subordinate-tasking.lisp
*SCENARIO*
>saf>sys>new-storage.lisp
*SAVE-INSTANCE-FILTER*
>saf>sys>new-storage.lisp
GET-SCREEN-PARAMETERS
>saf>sys>new-storage.lisp
RETURN-SCENARIO-OBJECT-LIST
>saf>sys>new-storage.lisp
Called by:  NAME-AND-STORE-SCENARIO
>saf>sys>new-storage.lisp
NAME-AND-STORE-OVERLAY
>saf>sys>new-storage.lisp
Description:  None

```

### 2.3.3.2.37 RETURN-SCENARIO-OBJECT-LIST

Definition 37

```

>saf>sys>new-storage.lisp
Type: Function
Arguments:  ()
Outputs:
Calls: *ALL-OVERLAYS*
>saf>sys>vars.lisp
GET-SUBORDINATES-INSTANCES
>saf>objects>simnet-agent.lisp
ALL-LOCAL-VEHICLES
>saf>simnet-objects>vehicle-tracking.lisp
*SAVE-INSTANCE-FILTER*
>saf>sys>new-storage.lisp
*OVERLAY-TO-SAVE*
>saf>sys>new-storage.lisp
GET-CURRENT-TOP-UNITS
>saf>sys>new-storage.lisp
Called by:  STORE-SCENARIO
>saf>sys>new-storage.lisp
Description:  None

```

### 2.3.3.2.38 GET-CURRENT-TOP-UNITS

Definition 38

```

>saf>sys>new-storage.lisp
Type: Function
Arguments:  ()

```

## Outputs:

Calls: LOCAL

&gt;saf&gt;network&gt;vars.lisp

TOP-LEVEL-UNITS

&gt;saf&gt;simnet-objects&gt;vehicle-tracking.lisp

Called by: RETURN-SCENARIO-OBJECT-LIST

&gt;saf&gt;sys&gt;new-storage.lisp

Description: None

**2.3.3.2.39 \*SCENARIO-DIRECTORY\***

Definition 39

&gt;saf&gt;sys&gt;new-storage.lisp

Type: Parameter

Arguments: ()

Outputs:

Calls: None

Called by: LOAD-SCENARIO

&gt;saf&gt;sys&gt;new-storage.lisp

(METHOD STORE SCENARIO)

&gt;saf&gt;sys&gt;new-storage.lisp

Description: None

**2.3.3.2.40 \*OVERLAY-DIRECTORY\***

Definition 40

&gt;saf&gt;sys&gt;new-storage.lisp

Type: Parameter

Arguments: ()

Outputs:

Calls: None

Called by: LOAD-OVERLAY

&gt;saf&gt;sys&gt;new-storage.lisp

(METHOD STORE SCENARIO)

&gt;saf&gt;sys&gt;new-storage.lisp

Description: None

**2.3.3.2.41 (METHOD STORE SCENARIO)**

Definition 41

&gt;saf&gt;sys&gt;new-storage.lisp

Type: Method

Arguments: (&amp;KEY (FILENAME NIL))

Outputs:

Calls: SAVE-TOP-LEVEL-AND-INFERIORS

&gt;saf&gt;sys&gt;new-storage.lisp

CONCAT

&gt;saf&gt;sys&gt;new-storage.lisp

**\*SCENARIO-DIRECTORY\***

>saf>sys>new-storage.lisp

**\*OVERLAY-DIRECTORY\***

>saf>sys>new-storage.lisp

Called by: None

Description: None

### 2.3.3.2.42 LOAD-OVERLAY

Definition 42

>saf>sys>new-storage.lisp

Type: Function

Arguments: ()

Outputs:

Calls: NAME

>saf>sysdcl.lisp

**\*PVD-DISPLAY\***

>saf>sys>vars.lisp

**\*ALL-OVERLAYS\***

>saf>sys>vars.lisp

**\*DB-INSTANCES\***

>saf>sys>vars.lisp

MKATOM

>saf>sys>new-storage.lisp

READ-AND-MAKE-INSTANCES

>saf>sys>new-storage.lisp

**\*OVERLAY-DIRECTORY\***

>saf>sys>new-storage.lisp

OVERLAY

>saf>cm>overlay.lisp

OVERLAY

>saf>cm>overlay.lisp

Called by: SAVE-OR-LOAD-OVERLAYS

>saf>sys>new-storage.lisp

Description: None

### 2.3.3.2.43 LOAD-SCENARIO

Definition 43

>saf>sys>new-storage.lisp

Type: Function

Arguments: ()

Outputs:

Calls: NAME

>saf>sysdcl.lisp

**\*PVD-DISPLAY\***

>saf>sys>vars.lisp

**\*OPFOR-IO\***

>saf>sys>vars.lisp

**\*BMI-PROGRAM\***

>saf>sys>vars.lisp

**\*ALL-OVERLAYS\***

```

>saf>sys>vars.lisp
*DB-INSTANCES*
>saf>sys>vars.lisp
SAY
>saf>sys>macros.lisp
*REAPPEAR-LATENCY*
>saf>rudp>vars.lisp
*RANGE-THRESHOLD*
>saf>rudp>vars.lisp
*UPDATE-RATE*
>saf>rudp>vars.lisp
*CLUSTER-DISTANCE*
>saf>rudp>vars.lisp
STANDALONEP
>saf>network>connection.lisp
SEND-AN-IVIS-FINE-CONTROL-PACKET
>saf>network>commands.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
SIMNET-AGENT
>saf>objects>simnet-agent.lisp
*DEFAULT-BATTALION-NUMBER*
>saf>bmi>bmi-frame.lisp
*TOP-LEVEL-TASKING*
>saf>ui>subordinate-tasking.lisp
MKATOM
>saf>sys>new-storage.lisp
READ-AND-MAKE-INSTANCES
>saf>sys>new-storage.lisp
*SCENARIO-DIRECTORY*
>saf>sys>new-storage.lisp
CREATE-STORED-INSTANCE
>saf>sys>new-storage.lisp
OVERLAY
>saf>cm>overlay.lisp
OVERLAY
>saf>cm>overlay.lisp

```

Called by: (METHOD COM-RESTORE-EXERCISE-INTERNAL SAF)

No Source File Record

Description: None

#### 2.3.3.2.44 CREATE-STORED-INSTANCE

Definition 44

```
>saf>sys>new-storage.lisp
```

Type: Function

Arguments: (INSTANCE)

Outputs:

**Calls:** NEW-SBX-UNIQUE-UNIT-ID  
 >saf>sys>vars.lisp  
 MAP-ECHELON-TO-NUMBER  
 >saf>sys>interim-model.lisp  
 MAP-ECHELON-TYPE-TO-NUMBER  
 >saf>sys>interim-model.lisp  
 GET-LOCAL-HOST-SAF-PORT  
 >saf>network>vars.lisp  
 CREATE  
 >saf>network>vars.lisp  
 NET-MSG  
 >saf>rudp>outgoing.lisp  
 RETURN-FORCE-AND-COUNTRY-D-AND-O  
 >saf>bmi>bmi-frame.lisp  
 SET-INFERIORS-PORT-AND-SUPERIOR-ID  
 >saf>sys>new-storage.lisp  
 SANDBOX-OBJECT  
 >saf>bmi>presentation-types.lisp  
 SANDBOX-OBJECT  
 >saf>bmi>presentation-types.lisp  
**Called by:** LOAD-SCENARIO  
 >saf>sys>new-storage.lisp  
**Description:** None

#### 2.3.3.2.45 SET-INFERIORS-PORT-AND-SUPERIOR-ID

Definition 45

>saf>sys>new-storage.lisp  
**Type:** Function  
**Arguments:** (INSTANCE PORT UNIQ-ID)  
**Outputs:**  
**Calls:** SIMNET-AGENT  
 >saf>objects>simnet-agent.lisp  
 SIMNET-AGENT  
 >saf>objects>simnet-agent.lisp  
 SIMNET-AGENT  
 >saf>objects>simnet-agent.lisp  
 SET-INFERIORS-PORT-AND-SUPERIOR-ID  
 >saf>sys>new-storage.lisp  
**Called by:** SET-INFERIORS-PORT-AND-SUPERIOR-ID  
 >saf>sys>new-storage.lisp  
 CREATE-STORED-INSTANCE  
 >saf>sys>new-storage.lisp  
**Description:** None

#### 2.3.3.2.46 COPY-RELEVANT-IVS

Definition 46

>saf>sys>new-storage.lisp  
**Type:** Function  
**Arguments:** (FROM-INST TO-INST)  
**Outputs:**



Outputs:

Calls: NAME

>saf>sysdcl.lisp

MULTIPLE-MENU-CHOOSE

>saf>sys>new-storage.lisp

Called by: COM-DELETE-EXERCISES

>saf>ui>commands.lisp

COM-DELETE-SCENARIOS

>saf>ui>commands.lisp

Description: None

### 2.3.3.2.50 CHOOSE-OVERLAYS-TO-DELETE

Definition 50

>saf>sys>new-storage.lisp

Type: Function

Arguments: (&OPTIONAL (ALL NIL))

Outputs:

Calls: NAME

>saf>sysdcl.lisp

MULTIPLE-MENU-CHOOSE

>saf>sys>new-storage.lisp

Called by: COM-DELETE-OVERLAYS

>saf>ui>commands.lisp

Description: None



## 2.4 MAP DISPLAY CSC

The map display CSC provides a plan view display of the battlefield. It allows the user to see the battlefield at different zoom levels, at different locations, and with different terrain features displayed. The drawing of the terrain is handled by the update process. A separate process is used because drawing the terrain is a very compute intensive process and you don't want to tie up the user or RUDP process until it finishes. It also allows new terrain drawing commands to interrupt older ones. In addition to displaying the battlefield terrain, the color display shows the position and state of own and other forces on the battlefield and fire activity. By clicking on the units on the map display, you can invoke most of the commands available via the task organization display. What vehicles are displayed is controlled by the current map display view. In the omniscient view, all vehicles are displayed. In the commander's view, all the vehicles on your side plus the enemy vehicles which can be seen by the own vehicles are displayed. Figure 2.4-1 shows the sub-level CSCs of the Map Display CSC.

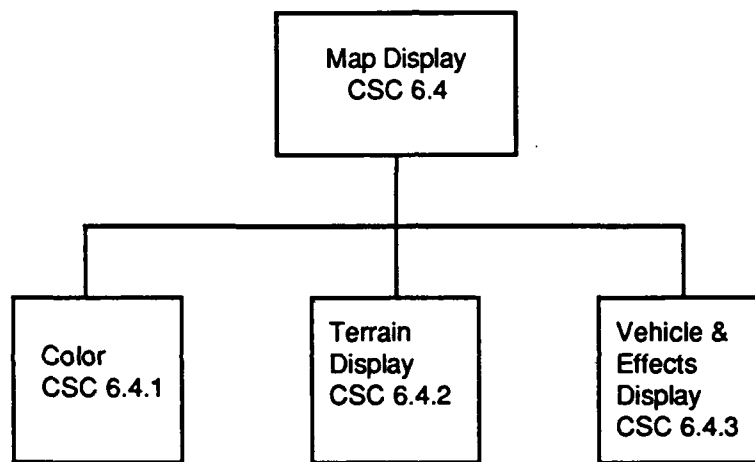


Figure 2.4-1 CSCs of the Map Display CSC

### 2.4.1 Color CSC

This CSC contains the code for the update process which draws the terrain and refreshes the vehicles when the screen is redrawn. In addition, the update process handles poll requesting by timing when the last request went out and then making sure that the last request has been filled. This CSC also contains the code for the color display menu bar. This CSC contains the following CSUs:

```

sys>update-process.lisp csu
ui>frame-utils.lisp csu
  
```

#### 2.4.1.1 CSU sys>update-process.lisp

This unit contains the main loop for the update process. The update process repeatedly checks the message queues used for interprocess communication, looking for messages from the network process (RUDP), or the user process. It handles these messages by drawing the requested features on the PVD.

The message queues are implemented as the global variables *\*update-process-queue\**, for messages from the user process, and *\*rudp-receive-queue\**, for messages from RUDP. (*\*network-to-update-process-queue\** is no longer used; the update process generates a *error* if this queue is found to be non-empty.)

The main loop of the update process handles drawing operations in the following order:

- (1) Input from the user process
- (2) Draw vehicles that need updates
- (3) Draw new effects
- (4) Erase old effects
- (5) Draw some terrain

Terrain is drawn one quadtree node at a time, by the function *draw-map*, except when the *entire* map needs to be drawn; then the quadtree is not needed and terrain is drawn feature by feature, to save time.

Notice that, if there is no terrain to draw, the update process puts itself to sleep with a call to the Symbolics function *process-wait*, at the end of *update-top-level-aux*. The function *update-process-wakeup* revives it.

#### 2.4.1.1.1 \*TERRAIN-TO-DRAW\*

Definition 1

```
>saf>sys>update-process.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: DRAW-ANOTHER-TERRAIN-QUAD
           >saf>sys>update-process.lisp
           DRAW-MAP
           >saf>sys>update-process.lisp
           UPDATE-TOP-LEVEL-AUX
           >saf>sys>update-process.lisp
Description: None
```

#### 2.4.1.1.2 \*TERRAIN-CONTOURS-TO-DRAW\*

Definition 2

```
>saf>sys>update-process.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: DRAW-ANOTHER-TERRAIN-QUAD
           >saf>sys>update-process.lisp
           DRAW-MAP
           >saf>sys>update-process.lisp
           UPDATE-TOP-LEVEL-AUX
           >saf>sys>update-process.lisp
Description: None
```

**2.4.1.1.3 \*EFFECTS-ERASE-TIME\***

## Definition 3

>saf>sys>update-process.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: UPDATE-TOP-LEVEL-AUX  
>saf>sys>update-process.lisp  
Description: number of seconds effects remain on screen before erasure

**2.4.1.1.4 \*UPDATE-PROCESS-WAIT-TIME\***

## Definition 4

>saf>sys>update-process.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: UPDATE-TOP-LEVEL-AUX  
>saf>sys>update-process.lisp  
UPDATE-PROCESS-WAKE-UP  
>saf>sys>update-process.lisp  
Description: when the update process put itself into a wait state

**2.4.1.1.5 \*UPDATE-PROCESS-LAST-CYCLE\***

## Definition 5

>saf>sys>update-process.lisp  
Type: Parameter  
Arguments: ()  
Outputs:  
Calls: None  
Called by: UPDATE-TOP-LEVEL-AUX  
>saf>sys>update-process.lisp  
(METHOD MAKE-INSTANCE SAF AFTER)  
>saf>ui>frame.lisp  
MAKE-UPDATE-PROCESS  
>saf>ui>processes.lisp  
Description: None

**2.4.1.1.6 \*UPDATE-PROCESS-MAX-WAIT-TIME\***

## Definition 6

>saf>sys>update-process.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None

Called by: UPDATE-PROCESS-WAKE-UP  
>saf>sys>update-process.lisp  
Description: max number of seconds update process will sleep

#### 2.4.1.1.7 \*TIME-LAST-POLLED\*

Definition 7

>saf>sys>update-process.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: UPDATE-TOP-LEVEL-AUX  
>saf>sys>update-process.lisp  
Description: time the butterfly was last polled to get vehicle positions

#### 2.4.1.1.8 UPDATE-PROCESS-WAKE-UP

Definition 8

>saf>sys>update-process.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*INTERFACE-TO-UPDATE-PROCESS-QUEUE\*  
>saf>sys>vars.lisp  
\*RUDP-RECEIVE-QUEUE\*  
>saf>rudp>vars.lisp  
\*UPDATE-PROCESS-WAIT-TIME\*  
>saf>sys>update-process.lisp  
\*UPDATE-PROCESS-MAX-WAIT-TIME\*  
>saf>sys>update-process.lisp  
Called by: UPDATE-TOP-LEVEL-AUX  
>saf>sys>update-process.lisp  
Description: decides when to make the update process runnable

#### 2.4.1.1.9 UPDATE-TOP-LEVEL

Definition 9

>saf>sys>update-process.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*WHERE-ARE-THEY-POLL-WAIT\*  
>saf>sys>vars.lisp  
\*STOP-UPDATE-PROCESS\*  
>saf>sys>vars.lisp  
UPDATE-TOP-LEVEL-AUX  
>saf>sys>update-process.lisp

Called by: (METHOD MAKE-INSTANCE SAF AFTER)

```
>saf>ui>frame.lisp
MAKE-UPDATE-PROCESS
>saf>ui>processes.lisp
```

Description: top level function for the update process

#### 2.4.1.1.10 UPDATE-TOP-LEVEL-AUX

Definition 10

```
>saf>sys>update-process.lisp
```

Type: Function

Arguments: ()

Outputs:

Calls: \*QUAD-TREE\*

```
>map>terrain-vars.lisp
```

```
*VIEW-VEHICLE-ID*
```

```
>saf>sys>vars.lisp
```

```
GODS-EYE-VIEW
```

```
>saf>sys>vars.lisp
```

```
NON-GODS-EYE-VIEW
```

```
>saf>sys>vars.lisp
```

```
COMMANDERS-EYE-VIEW
```

```
>saf>sys>vars.lisp
```

```
*WHERE-ARE-THEY-POLL-WAIT*
```

```
>saf>sys>vars.lisp
```

```
*WHERE-ARE-THEY-POLL-FREQUENCY*
```

```
>saf>sys>vars.lisp
```

```
*INTERFACE-TO-UPDATE-PROCESS-QUEUE*
```

```
>saf>sys>vars.lisp
```

```
*NETWORK-TO-UPDATE-PROCESS-QUEUE*
```

```
>saf>sys>vars.lisp
```

```
DEQUEUE
```

```
>saf>sys>macros.lisp
```

```
POLL
```

```
>saf>network>vars.lisp
```

```
STANDALONEP
```

```
>saf>network>connection.lisp
```

```
PROCESS-RECEIVED-PACKETS
```

```
>saf>rudp>incoming.lisp
```

```
NET-MSG
```

```
>saf>rudp>outgoing.lisp
```

```
GET-VEHICLE
```

```
>saf>simnet-objects>vehicle-tracking.lisp
```

```
ERASE-ELASPED-EFFECTS
```

```
>saf>simnet-objects>draw-effects.lisp
```

```
*TERRAIN-TO-DRAW*
```

```
>saf>sys>update-process.lisp
```

```
*TERRAIN-CONTOURS-TO-DRAW*
```

```
>saf>sys>update-process.lisp
```

```
*EFFECTS-ERASE-TIME*
```

```
>saf>sys>update-process.lisp
```

```
*UPDATE-PROCESS-WAIT-TIME*
```

```
>saf>sys>update-process.lisp
```

```

*UPDATE-PROCESS-LAST-CYCLE*
>saf>sys>update-process.lisp
*TIME-LAST-POLLED*
>saf>sys>update-process.lisp
UPDATE-PROCESS-WAKE-UP
>saf>sys>update-process.lisp
PROCESS-USER-COMMAND
>saf>sys>update-process.lisp
PROCESS-NETWORK-COMMAND
>saf>sys>update-process.lisp
DRAW-ANOTHER-TERRAIN-QUAD
>saf>sys>update-process.lisp
Called by: UPDATE-TOP-LEVEL
>saf>sys>update-process.lisp
Description: the guts of the update process top level processing

```

#### 2.4.1.1.11 PROCESS-USER-COMMAND

Definition 11

```

>saf>sys>update-process.lisp
Type: Function
Arguments: (COMMAND ARGUMENTS)
Outputs:
Calls: *ZOOM-LEVELS*
>map>zoom-levels.lisp
*ZOOM-LEVELS*
>map>zoom-levels.lisp
*PVD-DISPLAY*
>saf>sys>vars.lisp
*PVD-LEGEND*
>saf>sys>vars.lisp
PROCESS-NEW-MAP-OPTIONS
>saf>sys>update-process.lisp
DRAW-MAP
>saf>sys>update-process.lisp
Called by: UPDATE-TOP-LEVEL-AUX
>saf>sys>update-process.lisp
Description: top level function for the update process

```

#### 2.4.1.1.12 POLL-COMPLETED

Definition 12

```

>saf>sys>update-process.lisp
Type: Function
Arguments: ()
Outputs:
Calls: *MAX-VEHICLE-ID*
>saf>sys>constants.lisp
*WHERE-ARE-THEY-POLL-WAIT*
>saf>sys>vars.lisp
ACCESS-VEHICLE-INSTANCE
>saf>simnet-objects>macros.lisp

```

```

ACCESS-NEW-FLAG
>saf>simnet-objects>macros.lisp
SET-NEW-FLAG
>saf>simnet-objects>macros.lisp
ACCESS-PAINTED-FLAG
>saf>simnet-objects>macros.lisp
GET-VEHICLE-HOLDER
>saf>simnet-objects>vehicle-tracking.lisp

```

Called by: PROCESS-VEHICLE-POSITION-POLL-COMPLETED-PKT  
>saf>rudp>handle-incoming.lisp

Description: None

#### 2.4.1.1.13 PROCESS-NETWORK-COMMAND

Definition 13

```
>saf>sys>update-process.lisp
```

Type: Function

Arguments: (COMMAND ARGUMENTS)

Outputs:

Calls: None

Called by: UPDATE-TOP-LEVEL-AUX

```
>saf>sys>update-process.lisp
```

Description: Process a request from the network process

#### 2.4.1.1.14 PROCESS-NEW-MAP-OPTIONS

Definition 14

```
>saf>sys>update-process.lisp
```

Type: Function

Arguments: (MAP-OPTIONS)

Outputs:

Calls: \*CURRENT-ZOOM-LEVEL\*

```
>map>zoom-levels.lisp
```

```
CURRENT-SCALE
```

```
>map>zoom-levels.lisp
```

```
*CURRENT-ZOOM-LEVEL*
```

```
>map>zoom-levels.lisp
```

```
*TERRAIN-OPTIONS*
```

```
>saf>sys>vars.lisp
```

```
DRAW-MAP
```

```
>saf>sys>update-process.lisp
```

Called by: PROCESS-USER-COMMAND

```
>saf>sys>update-process.lisp
```

Description: None

**2.4.1.1.15 \*SOIL-TYPES\***

Definition 15

```

>saf>sys>update-process.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: DRAW-ANOTHER-TERRAIN-QUAD
           >saf>sys>update-process.lisp
           DRAW-MAP
           >saf>sys>update-process.lisp
Description: None

```

**2.4.1.1.16 DRAW-MAP**

Definition 16

```

>saf>sys>update-process.lisp
Type: Function
Arguments: (&OPTIONAL (NEW-SCALE NIL))
Outputs:
Calls: *QUAD-TREE*
       >map>terrain-vars.lisp
       *ZOOM-LEVELS*
       >map>zoom-levels.lisp
       *CURRENT-ZOOM-LEVEL*
       >map>zoom-levels.lisp
       *ZOOM-LEVELS*
       >map>zoom-levels.lisp
       *CURRENT-ZOOM-LEVEL*
       >map>zoom-levels.lisp
       QUADS-TO-DRAW
       >map>quadtree-search.lisp
       *PVD-DISPLAY*
       >saf>sys>vars.lisp
       *PVD-LEGEND*
       >saf>sys>vars.lisp
       *BMI-PROGRAM*
       >saf>sys>vars.lisp
       *TERRAIN-OPTIONS*
       >saf>sys>vars.lisp
       REDRAW-VEHICLES
       >saf>simnet-objects>vehicle-tracking.lisp
       *TERRAIN-TO-DRAW*
       >saf>sys>update-process.lisp
       *TERRAIN-CONTOURS-TO-DRAW*
       >saf>sys>update-process.lisp
       *SOIL-TYPES*
       >saf>sys>update-process.lisp
       DRAW-SANDBOX
       >saf>sandbox>sandbox.lisp
       REDRAW-OVERLAYS
       >saf>cm>overlay.lisp

```



Called by: PROCESS-NEW-MAP-OPTIONS

>saf>sys>update-process.lisp  
 PROCESS-USER-COMMAND  
 >saf>sys>update-process.lisp

Description: None

#### 2.4.1.1.17 DRAW-ANOTHER-TERRAIN-QUAD

Definition 17

>saf>sys>update-process.lisp

Type: Function

Arguments: ()

Outputs:

Calls: \*ZOOM-LEVELS\*  
 >map>zoom-levels.lisp  
 \*CURRENT-ZOOM-LEVEL\*  
 >map>zoom-levels.lisp  
 \*ZOOM-LEVELS\*  
 >map>zoom-levels.lisp  
 \*CURRENT-ZOOM-LEVEL\*  
 >map>zoom-levels.lisp  
 DRAW-ALL-TERRAIN  
 >map>draw-terrain.lisp  
 DRAW-TERRAIN  
 >map>draw-terrain.lisp  
 \*PVD-DISPLAY\*  
 >saf>sys>vars.lisp  
 \*TERRAIN-TO-DRAW\*  
 >saf>sys>update-process.lisp  
 \*TERRAIN-CONTOURS-TO-DRAW\*  
 >saf>sys>update-process.lisp  
 \*SOIL-TYPES\*  
 >saf>sys>update-process.lisp

Called by: UPDATE-TOP-LEVEL-AUX

>saf>sys>update-process.lisp

Description: None

#### 2.4.1.2 CSU ui>frame-utils.lisp

This unit defines some pane-types for use in the color display window and some routines to handle highlighting of menu-bar items.

##### 2.4.1.2.1 MAP-WINDOW

Definition 1

>saf>ui>frame-utils.lisp

Type: Flavor

Arguments: ()

Outputs:

Calls: UTM-GRID-MIXIN  
 >map>utm-grid-mixin.lisp

Called by: None  
Description: None

#### 2.4.1.2.2 MAP-WINDOW Definition 2

>saf>ui>frame-utils.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: PVD PROGRAM-FRAME-OPTIONS  
>saf>ui>frame.lisp  
Description: None

#### 2.4.1.2.3 MAP-LEGEND Definition 3

>saf>ui>frame-utils.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.1.2.4 MAP-LEGEND Definition 4

>saf>ui>frame-utils.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: PVD PROGRAM-FRAME-OPTIONS  
>saf>ui>frame.lisp  
Description: None

#### 2.4.1.2.5 HIGHLIGHT-BUTTON Definition 5

>saf>ui>frame-utils.lisp  
Type: Function  
Arguments: (PROGRAM MENU BUTTON-STRING)  
Outputs:  
Calls: HIGHLIGHT-BUTTON-1  
>saf>ui>frame-utils.lisp

Called by: (METHOD COM-TERRAIN-OPTIONS-INTERNAL PVD)

No Source File Record  
 (METHOD COM-REFRESH-INTERNAL PVD)  
 No Source File Record  
 (METHOD COM-RESCALE-INTERNAL PVD)  
 No Source File Record  
 (METHOD COM-ZOOM-OUT-INTERNAL PVD)  
 No Source File Record  
 (METHOD COM-PAN-INTERNAL PVD)  
 No Source File Record  
 (METHOD COM-ZOOM-IN-INTERNAL PVD)  
 No Source File Record  
 DEFINE-PVD-MENU-COMMAND  
 >saf>ui>commands.lisp

Description: None

#### 2.4.1.2.6 HIGHLIGHT-BUTTON-1

Definition 6

>saf>ui>frame-utils.lisp

Type: Function

Arguments: (WINDOW BUTTON-STRING)

Outputs:

Calls: None

Called by: SWITCH-ORIENTATION-HIGHLIGHT

>saf>interface>formations.lisp  
 HIGHLIGHT-SELECTION  
 >saf>interface>model-menu.lisp  
 SWITCH-HIGHLIGHT  
 >saf>interface>model-menu.lisp  
 HIGHLIGHT-BUTTON  
 >saf>ui>frame-utils.lisp

Description: None

#### 2.4.2 Terrain Display CSC

This CSC contains the code to draw the terrain and to access the quadtree terrain representation. The access routines are also used by the route code. This CSC contains the following CSUs:

map>clip.lisp csu  
 map>color-map.lisp csu  
 map>control.lisp csu  
 map>draw-wide-curve.lisp  
 map>grids.lisp csu  
 map>intersection.lisp csu  
 map>legend.lisp csu  
 map>quadtree-search.lisp csu  
 map>scalable-window.lisp csu  
 map>terrain-vars.lisp csu

```
map>utilities.lisp csu
map>utm-grid-mixin.lisp csu
map>vectors.lisp csu
map>zoom-levels.lisp csu
```

### 2.4.2.1 CSU map>clip.lisp

This unit contains the line clipping routines which are used by the drawing routines as well as the search routines. The functions calculate the intersection of a line segment with given endpoints and a rectangle with given corners. These routines correspond to the Nicholl, Lee, Nicholl algorithm from the 1987 SIGGRAPH proceedings. The lines forming the rectangle divide the plane into 9 regions, like a tic-tac-toe board. Functions determine if endpoints of the segment lie in certain combinations of these 9 regions; this information is then used to calculate the intersection segment. The algorithm uses rotations and reflections to reduce the number of cases that have to be handled.

#### 2.4.2.1.1 ROTATE-90-C

Definition 1

```
>map>clip.lisp
Type: Macro
Arguments: (X Y)
Outputs:
Calls: ROTATE-90-C
       >map>clip.lisp
Called by: P2-LEFT
          >map>clip.lisp
          CENTER-COLUMN
          >map>clip.lisp
          ROTATE-90-C
          >map>clip.lisp
Description: None
```

#### 2.4.2.1.2 ROTATE-180-C

Definition 2

```
>map>clip.lisp
Type: Macro
Arguments: (X Y)
Outputs:
Calls: ROTATE-180-C
       >map>clip.lisp
Called by: INSIDE
          >map>clip.lisp
          CLIP
          >map>clip.lisp
          ROTATE-180-C
          >map>clip.lisp
Description: None
```

**2.4.2.1.3 ROTATE-270-C**

## Definition 3

>map>clip.lisp  
Type: Macro  
Arguments: (X Y)  
Outputs:  
Calls: ROTATE-270-C  
>map>clip.lisp  
Called by: P2-LEFT  
>map>clip.lisp  
CENTER-COLUMN  
>map>clip.lisp  
ROTATE-270-C  
>map>clip.lisp  
Description: None

**2.4.2.1.4 REFLECT-X-MINUS-Y**

## Definition 4

>map>clip.lisp  
Type: Macro  
Arguments: (X Y)  
Outputs:  
Calls: REFLECT-X-MINUS-Y  
>map>clip.lisp  
Called by: TOP-LEFT-CORNER  
>map>clip.lisp  
REFLECT-X-MINUS-Y  
>map>clip.lisp  
Description: None

**2.4.2.1.5 REFLECT-X-AXIS**

## Definition 5

>map>clip.lisp  
Type: Macro  
Arguments: (Y)  
Outputs:  
Calls: REFLECT-X-AXIS  
>map>clip.lisp  
Called by: LEFT-EDGE  
>map>clip.lisp  
LEFT-COLUMN  
>map>clip.lisp  
REFLECT-X-AXIS  
>map>clip.lisp  
Description: None

**2.4.2.1.6 \*DISPLAY\***

## Definition 6

>map>clip.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INSIDE  
>map>clip.lisp  
P2-BOTTOM  
>map>clip.lisp  
LEFT-EDGE  
>map>clip.lisp  
LEFT-BOTTOM-REGION  
>map>clip.lisp  
TOP-LEFT-CORNER  
>map>clip.lisp  
LEFT-COLUMN  
>map>clip.lisp  
CLIP  
>map>clip.lisp  
Description: None

**2.4.2.1.7 CLIP**

## Definition 7

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT Y-TOP X-RIGHT Y-BOTTOM X1 Y1 X2 Y2)  
Outputs:  
Calls: ROTATE-180-C  
>map>clip.lisp  
\*DISPLAY\*  
>map>clip.lisp  
LEFT-COLUMN  
>map>clip.lisp  
CENTER-COLUMN  
>map>clip.lisp  
Called by: CLIP-POINTS-TO-WINDOW  
>map>draw-terrain.lisp  
GET-THIS-NODE  
>map>quadtree-search.lisp  
POSSIBLE-INTERSECTION  
>map>intersection.lisp  
FIND-NEAREST-ROAD-SEGMENT  
>saf>cm>road-routes.lisp  
Description: None

**2.4.2.1.8 LEFT-COLUMN**

Definition 8

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT Y-TOP X-RIGHT Y-BOTTOM X1 Y1 X2 Y2)  
Outputs:  
Calls: REFLECT-X-AXIS  
>map>clip.lisp  
\*DISPLAY\*  
>map>clip.lisp  
TOP-LEFT-CORNER  
>map>clip.lisp  
LEFT-EDGE  
>map>clip.lisp  
Called by: CLIP  
>map>clip.lisp  
Description: None

**2.4.2.1.9 TOP-LEFT-CORNER**

Definition 9

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT Y-TOP X-RIGHT Y-BOTTOM X1 Y1 X2 Y2)  
Outputs:  
Calls: REFLECT-X-MINUS-Y  
>map>clip.lisp  
\*DISPLAY\*  
>map>clip.lisp  
LEFT-BOTTOM-REGION  
>map>clip.lisp  
Called by: LEFT-COLUMN  
>map>clip.lisp  
Description: None

**2.4.2.1.10 LEFT-BOTTOM-REGION**

Definition 10

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT IGNORE X-RIGHT Y-BOTTOM X1 Y1 X2 Y2 REL-X2 REL-Y2 LEFT-PRODUCT)  
Outputs:  
Calls: \*DISPLAY\*  
>map>clip.lisp  
Called by: TOP-LEFT-CORNER  
>map>clip.lisp  
Description: None

**2.4.2.1.11 LEFT-EDGE**

## Definition 11

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT Y-TOP X-RIGHT Y-BOTTOM X1 Y1 X2 Y2)  
Outputs:  
Calls: REFLECT-X-AXIS  
>map>clip.lisp  
\*DISPLAY\*  
>map>clip.lisp  
P2-BOTTOM  
>map>clip.lisp  
Called by: CENTER-COLUMN  
>map>clip.lisp  
LEFT-COLUMN  
>map>clip.lisp  
Description: None

**2.4.2.1.12 P2-BOTTOM**

## Definition 12

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT IGNORE X-RIGHT Y-BOTTOM X1 Y1 X2 Y2)  
Outputs:  
Calls: \*DISPLAY\*  
>map>clip.lisp  
Called by: LEFT-EDGE  
>map>clip.lisp  
Description: None

**2.4.2.1.13 CENTER-COLUMN**

## Definition 13

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT Y-TOP X-RIGHT Y-BOTTOM X1 Y1 X2 Y2)  
Outputs:  
Calls: ROTATE-90-C  
>map>clip.lisp  
ROTATE-270-C  
>map>clip.lisp  
LEFT-EDGE  
>map>clip.lisp  
INSIDE  
>map>clip.lisp  
Called by: CLIP  
>map>clip.lisp  
Description: None



**2.4.2.1.14 P2-LEFT-TOP**

Definition 14

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT Y-TOP IGNORE IGNORE X1 Y1 X2 Y2)  
Outputs:  
Calls: None  
Called by: P2-LEFT  
>map>clip.lisp  
Description: None

**2.4.2.1.15 P2-LEFT**

Definition 15

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT Y-TOP X-RIGHT Y-BOTTOM X1 Y1 X2 Y2)  
Outputs:  
Calls: ROTATE-90-C  
>map>clip.lisp  
ROTATE-270-C  
>map>clip.lisp  
P2-LEFT-TOP  
>map>clip.lisp  
Called by: INSIDE  
>map>clip.lisp  
Description: None

**2.4.2.1.16 INSIDE**

Definition 16

>map>clip.lisp  
Type: Function  
Arguments: (X-LEFT Y-TOP X-RIGHT Y-BOTTOM X1 Y1 X2 Y2)  
Outputs:  
Calls: ROTATE-180-C  
>map>clip.lisp  
\*DISPLAY\*  
>map>clip.lisp  
P2-LEFT  
>map>clip.lisp  
Called by: CENTER-COLUMN  
>map>clip.lisp  
Description: None

**2.4.2.2 CSU map>color-map.lisp**

This unit contains the routines that generate the color alus for the Symbolics color system for display of the terrain. The alu information is stored as part of the terrain quadtree and the alus are generated when the color window is created or a new terrain database is loaded.

Terrain, vehicles, effects and overlays are displayed in separate image planes, each corresponding to a bitfield within the 8 bit color system pixels. The terrain is displayed in the lower 3 bits. See CSU color-window>color-alus.lisp for details on the image plane approach. Color alus are discussed in detail in the Symbolics documentation.

#### 2.4.2.2.1 **\*OVERLAY-ALU\***

Definition 1

```
>map>color-map.lisp
Type: EXPORT
Arguments:  ()
Outputs:
Calls: None
Called by:  None
Description: None
```

#### 2.4.2.2.2 **\*ERASE-OVERLAY-ALU\***

Definition 2

```
>map>color-map.lisp
Type: EXPORT
Arguments:  ()
Outputs:
Calls: None
Called by:  None
Description: None
```

#### 2.4.2.2.3 **\*OVERLAY-ALU\***

Definition 3

```
>map>color-map.lisp
Type: Variable
Arguments:  ()
Outputs:
Calls: None
Called by:  SELECT-POLYGON
           >map>control.lisp
           RUBBER-LINE
           >map>control.lisp
           ROTATABLE-RECTANGLE
           >map>control.lisp
           (PRESENTATION-FUNCTION CONTROL-MEASURE PRINTER)
           No Source File Record
           DRAW-BRIDGES
           >map>draw-terrain.lisp
           (METHOD DRAW-GRIDS UTM-GRID-MIXIN)
           >map>grids.lisp
           MAKE-COLOR-ALUS
           >map>color-map.lisp
           SETUP-COLOR-ALUS
           >map>color-map.lisp
```

```
SETUP-COLOR-ALUS
>saf>color-window>color-alus.lisp
(METHOD INSERT-POINT-AFTER GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD DELETE-POINT GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD MOVE-POINT GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD DRAW GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD INSERT-POINT-BEFORE LINE)
>saf>cm>line.lisp
(METHOD INSERT-POINT-AFTER LINE)
>saf>cm>line.lisp
(METHOD DELETE-POINT LINE)
>saf>cm>line.lisp
(METHOD MOVE-POINT LINE)
>saf>cm>line.lisp
(METHOD DRAW LINE)
>saf>cm>line.lisp
(METHOD DRAW CM-POINT)
>saf>cm>point.lisp
MAKE-ROUTE
>saf>cm>route.lisp
(METHOD INSERT-POINT-BEFORE ROUTE)
>saf>cm>route.lisp
(METHOD INSERT-POINT-AFTER ROUTE)
>saf>cm>route.lisp
(METHOD DELETE-POINT ROUTE)
>saf>cm>route.lisp
(METHOD MOVE-POINT ROUTE)
>saf>cm>route.lisp
(METHOD DRAW ROUTE)
>saf>cm>route.lisp
GET-ROAD-ROUTE
>saf>cm>road-routes.lisp
(METHOD DRAW CONTROL-MEASURE-POINT)
>saf>cm>control-measure-point.lisp
```

Description: None

#### 2.4.2.2.4 \*ERASE-OVERLAY-ALU\*

Definition 4

```
>map>color-map.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
```

Called by: SELECT-POLYGON

>map>control.lisp  
RUBBER-LINE  
>map>control.lisp  
ROTATABLE-RECTANGLE  
>map>control.lisp  
DRAW-UNIT-SYMBOL  
>map>control.lisp  
(METHOD EDIT CONTROL-MEASURE)  
>map>control.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>saf>color-window>color-alus.lisp  
(METHOD INSERT-POINT-AFTER GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD DELETE-POINT GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD MOVE-POINT GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD ERASE GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD INSERT-POINT-BEFORE LINE)  
>saf>cm>line.lisp  
(METHOD INSERT-POINT-AFTER LINE)  
>saf>cm>line.lisp  
(METHOD DELETE-POINT LINE)  
>saf>cm>line.lisp  
(METHOD MOVE-POINT LINE)  
>saf>cm>line.lisp  
(METHOD ERASE LINE)  
>saf>cm>line.lisp  
(METHOD ERASE CM-POINT)  
>saf>cm>point.lisp  
(METHOD INSERT-POINT-BEFORE ROUTE)  
>saf>cm>route.lisp  
(METHOD INSERT-POINT-AFTER ROUTE)  
>saf>cm>route.lisp  
(METHOD DELETE-POINT ROUTE)  
>saf>cm>route.lisp  
(METHOD MOVE-POINT ROUTE)  
>saf>cm>route.lisp  
(METHOD ERASE ROUTE)  
>saf>cm>route.lisp  
GET-ROAD-ROUTE  
>saf>cm>road-routes.lisp  
(METHOD ERASE CONTROL-MEASURE-POINT)  
>saf>cm>control-measure-point.lisp  
(METHOD DRAW-AS-FIRST-POINT CONTROL-MEASURE-POINT)  
>saf>cm>control-measure-point.lisp

Description: None

**2.4.2.2.5 \*SOIL-ALU\***

## Definition 5

>map>color-map.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-WATER-OR-LAND-TRIANGLES  
>map>draw-terrain.lisp  
DRAW-WATER-OR-LAND-TRIANGLES-MAYBE  
>map>draw-terrain.lisp  
(METHOD DRAW-GRIDS UTM-GRID-MIXIN)  
>map>grids.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp  
Description: None

**2.4.2.2.6 \*OBJECT-ALU\***

## Definition 6

>map>color-map.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-OBJECTS  
>map>draw-terrain.lisp  
DRAW-LEGEND-BUILDINGS  
>map>legend.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp  
Description: None

**2.4.2.2.7 \*TREE-ALU\***

## Definition 7

>map>color-map.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-ALL-CANOPIES  
>map>draw-terrain.lisp  
DRAW-CANOPY-TRIANGLES  
>map>draw-terrain.lisp  
DRAW-TREES  
>map>draw-terrain.lisp

(METHOD DRAW-LEGEND LEGEND-WINDOW)

>map>legend.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp

Description: None

#### 2.4.2.2.8 \*SOIL-ROAD-ALU\*

Definition 8

>map>color-map.lisp

Type: Variable

Arguments: ()

Outputs:

Calls: None

Called by: DRAW-ALL-ROADS

>map>draw-terrain.lisp  
DRAW-ROADS  
>map>draw-terrain.lisp  
(METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp

Description: None

#### 2.4.2.2.9 \*SOIL-RAIL-ALU\*

Definition 9

>map>color-map.lisp

Type: Variable

Arguments: ()

Outputs:

Calls: None

Called by: DRAW-ALL-RAILS

>map>draw-terrain.lisp  
DRAW-RAILS  
>map>draw-terrain.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp

Description: None

**2.4.2.2.10 \*SOIL-WATER-ALU\***

Definition 10

>map>color-map.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-ALL-RIVERS  
>map>draw-terrain.lisp  
DRAW-WATER  
>map>draw-terrain.lisp  
(METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp  
Description: None

**2.4.2.2.11 \*SOIL-MUCK-ALU\***

Definition 11

>map>color-map.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-ALL-RIVERS  
>map>draw-terrain.lisp  
DRAW-WATER-OR-LAND-TRIANGLES  
>map>draw-terrain.lisp  
DRAW-WATER-OR-LAND-TRIANGLES-MAYBE  
>map>draw-terrain.lisp  
DRAW-WATER  
>map>draw-terrain.lisp  
(METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp  
Description: None

**2.4.2.2.12 \*LOW-CONTOUR-ALU\***

Definition 12

>map>color-map.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None

Called by: DRAW-ALL-CONTOURS  
>map>draw-terrain.lisp  
DRAW-CONTOURS  
>map>draw-terrain.lisp  
(METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp  
Description: None

#### 2.4.2.2.13 \*HIGH-CONTOUR-ALU\*

Definition 13

>map>color-map.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-ALL-CONTOURS  
>map>draw-terrain.lisp  
DRAW-CONTOURS  
>map>draw-terrain.lisp  
(METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp  
Description: None

#### 2.4.2.2.14 \*LEGEND-TEXT-ALU\*

Definition 14

>map>color-map.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-LEGEND-BOX-AND-LINE  
>map>legend.lisp  
(METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp  
Description: None



**2.4.2.2.15 'MAKE-AN-ALU**

Definition 15

>map>color-map.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.2.16 MAKE-AN-ALU**

Definition 16

>map>color-map.lisp  
Type: Function  
Arguments: (SIZE POSITION FILL-WITHIN-FIELD)  
Outputs:  
Calls: None  
Called by: MAKE-COLOR-ALUS  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>map>color-map.lisp  
MAKE-ALU-AND-SET-COLOR-MAP  
>map>color-map.lisp  
SETUP-COLOR-ALUS  
>saf>color-window>color-alus.lisp  
Description: None

**2.4.2.2.17 'MAKE-ALU-AND-SET-COLOR-MAP**

Definition 17

>map>color-map.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.2.18 MAKE-ALU-AND-SET-COLOR-MAP**

Definition 18

>map>color-map.lisp  
Type: Function  
Arguments: (SIZE POSITION FILL-WITHIN-FIELD DEC-RED DEC-GREEN DEC-BLUE)  
Outputs:  
Calls: MAKE-AN-ALU  
>map>color-map.lisp

Called by: SETUP-COLOR-ALUS  
           >map>color-map.lisp  
           SETUP-COLOR-ALUS  
           >saf>color-window>color-alus.lisp  
 Description: None

#### 2.4.2.2.19 SETUP-COLOR-ALUS

Definition 19

          >map>color-map.lisp  
 Type: Function  
 Arguments: (&OPTIONAL (DATABASE ft-knox))  
 Outputs:  
 Calls: \*OVERLAY-ALU\*  
       >map>color-map.lisp  
       \*ERASE-OVERLAY-ALU\*  
       >map>color-map.lisp  
       \*SOIL-ALU\*  
       >map>color-map.lisp  
       \*OBJECT-ALU\*  
       >map>color-map.lisp  
       \*TREE-ALU\*  
       >map>color-map.lisp  
       \*SOIL-ROAD-ALU\*  
       >map>color-map.lisp  
       \*SOIL-RAIL-ALU\*  
       >map>color-map.lisp  
       \*SOIL-WATER-ALU\*  
       >map>color-map.lisp  
       \*SOIL-MUCK-ALU\*  
       >map>color-map.lisp  
       \*LOW-CONTOUR-ALU\*  
       >map>color-map.lisp  
       \*HIGH-CONTOUR-ALU\*  
       >map>color-map.lisp  
       \*LEGEND-TEXT-ALU\*  
       >map>color-map.lisp  
       MAKE-AN-ALU  
       >map>color-map.lisp  
       MAKE-ALU-AND-SET-COLOR-MAP  
       >map>color-map.lisp  
 Called by: None  
 Description: None

#### 2.4.2.2.20 MAKE-COLOR-ARRAY

Definition 20

          >map>color-map.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: None

Called by: None  
 Description: None

#### 2.4.2.2.21 MAKE-COLOR-ALUS

Definition 21

>map>color-map.lisp  
 Type: Function  
 Arguments: (COLOR-MAP)  
 Outputs:  
 Calls: \*OVERLAY-ALU\*  
 >map>color-map.lisp  
 \*ERASE-OVERLAY-ALU\*  
 >map>color-map.lisp  
 \*SOIL-ALU\*  
 >map>color-map.lisp  
 \*OBJECT-ALU\*  
 >map>color-map.lisp  
 \*TREE-ALU\*  
 >map>color-map.lisp  
 \*SOIL-ROAD-ALU\*  
 >map>color-map.lisp  
 \*SOIL-RAIL-ALU\*  
 >map>color-map.lisp  
 \*SOIL-WATER-ALU\*  
 >map>color-map.lisp  
 \*SOIL-MUCK-ALU\*  
 >map>color-map.lisp  
 \*LOW-CONTOUR-ALU\*  
 >map>color-map.lisp  
 \*HIGH-CONTOUR-ALU\*  
 >map>color-map.lisp  
 \*LEGEND-TEXT-ALU\*  
 >map>color-map.lisp  
 MAKE-AN-ALU  
 >map>color-map.lisp  
 SET-COLOR-MAP  
 >map>color-map.lisp  
 Called by: (METHOD UPDATE SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 Description: None

#### 2.4.2.2.22 SET-COLOR-MAP

Definition 22

>map>color-map.lisp  
 Type: Function  
 Arguments: (COLOR-ARRAY)  
 Outputs:  
 Calls: None

Called by: MAKE-COLOR-ALUS  
>map>color-map.lisp  
Description: None

### 2.4.2.3 CSU map>control.lisp

This unit contains routines for generation of default map control measures. The SAF system no longer uses these control measures, as it now creates its own, more specific, measures, using the control-measure object defined in CSU control-measure.lisp, and related code in the control measures CSC.

#### 2.4.2.3.1 \*UNIT-TYPES\*

Definition 1

>map>control.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: UNIT-BOUNDARY  
>map>control.lisp  
BATTLE-POSITION  
>map>control.lisp  
Description: None

#### 2.4.2.3.2 \*AREA-TYPES\*

Definition 2

>map>control.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: AREA-CONTROL-MEASURE  
>map>control.lisp  
Description: None

#### 2.4.2.3.3 \*LINE-TYPES\*

Definition 3

>map>control.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: LINE-CONTROL-MEASURE  
>map>control.lisp  
Description: None

#### 2.4.2.3.4 \*CONTROL-MEASURE-MENU-ITEMS\*

Definition 4

>map>control.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: AREA-CONTROL-MEASURE  
>map>control.lisp  
BATTLE-POSITION  
>map>control.lisp  
LINE-CONTROL-MEASURE  
>map>control.lisp  
UNIT-BOUNDARY  
>map>control.lisp  
ARROW-CONTROL-MEASURE  
>map>control.lisp  
EDIT-CONTROL-MEASURES  
>map>control.lisp  
Called by: CONTROL-MEASURES-MENU  
>map>control.lisp  
Description: None

#### 2.4.2.3.5 \*CONTROL-MEASURES\*

Definition 5

>map>control.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-ALL-CONTROL-MEASURES  
>map>control.lisp  
(METHOD EDIT CONTROL-MEASURE)  
>map>control.lisp  
(METHOD INIT CONTROL-MEASURE AFTER)  
>map>control.lisp  
Description: None

#### 2.4.2.3.6 CONTROL-MEASURE

Definition 6

>map>control.lisp  
Type: DEFINE-PRESENTATION-TYPE  
Arguments: ()  
Outputs:  
Calls: None  
Called by: LINE-CONTROL-MEASURE  
>map>control.lisp  
ARROW-CONTROL-MEASURE  
>map>control.lisp  
UNIT-BOUNDARY

```
>map>control.lisp  
AREA-CONTROL-MEASURE  
>map>control.lisp  
BATTLE-POSITION  
>map>control.lisp  
DRAW-ALL-CONTROL-MEASURES  
>map>control.lisp  
EDIT-CONTROL-MEASURES  
>map>control.lisp  
ARROW-CONTROL-MEASURE  
>map>control.lisp  
UNIT-BOUNDARY  
>map>control.lisp  
LINE-CONTROL-MEASURE  
>map>control.lisp  
BATTLE-POSITION  
>map>control.lisp  
AREA-CONTROL-MEASURE  
>map>control.lisp
```

Description: None

#### 2.4.2.3.7 'CONTROL-MEASURE

Definition 7

```
>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None
```

#### 2.4.2.3.8 CONTROL-MEASURE

Definition 8

```
>map>control.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None
```

#### 2.4.2.3.9 'AREA-CONTROL-MEASURE

Definition 9

```
>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:
```

Calls: None  
Called by: None  
Description: None

#### 2.4.2.3.10 AREA-CONTROL-MEASURE

Definition 10

>map>control.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: CONTROL-MEASURE  
>map>control.lisp  
Called by: None  
Description: None

#### 2.4.2.3.11 'BATTLE-POSITION

Definition 11

>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.3.12 BATTLE-POSITION

Definition 12

>map>control.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: CONTROL-MEASURE  
>map>control.lisp  
AREA-CONTROL-MEASURE  
>map>control.lisp  
Called by: None  
Description: None

#### 2.4.2.3.13 'LINE-CONTROL-MEASURE

Definition 13

>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:

Calls: None  
Called by: None  
Description: None

#### 2.4.2.3.14 LINE-CONTROL-MEASURE

Definition 14

>map>control.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: CONTROL-MEASURE  
>map>control.lisp  
Called by: None  
Description: None

#### 2.4.2.3.15 'UNIT-BOUNDARY

Definition 15

>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.3.16 UNIT-BOUNDARY

Definition 16

>map>control.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: CONTROL-MEASURE  
>map>control.lisp  
LINE-CONTROL-MEASURE  
>map>control.lisp  
Called by: None  
Description: None

#### 2.4.2.3.17 'ARROW-CONTROL-MEASURE

Definition 17

>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:



Calls: None  
Called by: None  
Description: None

#### 2.4.2.3.18 ARROW-CONTROL-MEASURE

Definition 18

>map>control.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: CONTROL-MEASURE  
>map>control.lisp  
LINE-CONTROL-MEASURE  
>map>control.lisp  
Called by: None  
Description: None

#### 2.4.2.3.19 (METHOD INIT CONTROL-MEASURE AFTER)

Definition 19

>map>control.lisp  
Type: Method  
Arguments: (IGNORE) .  
Outputs:  
Calls: \*CONTROL-MEASURES\*  
>map>control.lisp  
Called by: None  
Description: None

#### 2.4.2.3.20 (METHOD EDIT CONTROL-MEASURE)

Definition 20

>map>control.lisp  
Type: Method  
Arguments: (WINDOW)  
Outputs:  
Calls: \*ERASE-OVERLAY-ALU\*  
>map>color-map.lisp  
\*MAP-OPTIONS\*  
>map>draw-terrain.lisp  
\*CONTROL-MEASURES\*  
>map>control.lisp  
DRAW-ALL-CONTROL-MEASURES  
>map>control.lisp  
Called by: None  
Description: None

**2.4.2.3.21 (METHOD DRAW AREA-CONTROL-MEASURE)**

Definition 21

>map>control.lisp  
Type: Method  
Arguments: (WINDOW ALU)  
Outputs:  
Calls: WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
Called by: None  
Description: None

**2.4.2.3.22 (METHOD ENTER-NEW-CONTROL-MEASURE AREA-CONTROL-MEASURE)**

Definition 22

>map>control.lisp  
Type: Method  
Arguments: (WINDOW)  
Outputs:  
Calls: AREA-CONTROL-MEASURE  
>map>control.lisp  
Called by: None  
Description: None

**2.4.2.3.23 (METHOD DRAW BATTLE-POSITION AFTER)**

Definition 23

>map>control.lisp  
Type: Method  
Arguments: (WINDOW ALU)  
Outputs:  
Calls: VEC-ROTATE  
>map>vectors.lisp  
VEC-ADD  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
DRAW-UNIT-SYMBOL  
>map>control.lisp  
Called by: None  
Description: None

**2.4.2.3.24 (METHOD ENTER-NEW-CONTROL-MEASURE BATTLE-POSITION)**

Definition 24

>map>control.lisp  
Type: Method  
Arguments: (WINDOW)  
Outputs:  
Calls: BATTLE-POSITION  
>map>control.lisp  
Called by: None  
Description: None

**2.4.2.3.25 (METHOD DRAW LINE-CONTROL-MEASURE)**

Definition 25

>map>control.lisp  
Type: Method  
Arguments: (WINDOW ALU)  
Outputs:  
Calls: WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
VEC-NORMALIZE  
>map>vectors.lisp  
VEC-ADD  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
VEC-SCALE  
>map>vectors.lisp  
DRAW-1-SCALLOPED-LINE  
>map>control.lisp  
DRAW-2-SCALLOPED-LINES  
>map>control.lisp  
Called by: None  
Description: None

**2.4.2.3.26 (METHOD ENTER-NEW-CONTROL-MEASURE LINE-CONTROL-MEASURE)**

Definition 26

>map>control.lisp  
Type: Method  
Arguments: (WINDOW)  
Outputs:  
Calls: LINE-CONTROL-MEASURE  
>map>control.lisp

Called by: None  
Description: None

**2.4.2.3.27 (METHOD DRAW UNIT-BOUNDARY AFTER)**  
Definition 27

>map>control.lisp  
Type: Method  
Arguments: (WINDOW ALU)  
Outputs:  
Calls: DRAW-UNIT-SYMBOL  
>map>control.lisp  
FIND-CENTER-POINT  
>map>control.lisp  
Called by: None  
Description: None

**2.4.2.3.28 (METHOD ENTER-NEW-CONTROL-MEASURE UNIT-BOUNDARY)**  
Definition 28

>map>control.lisp  
Type: Method  
Arguments: (WINDOW)  
Outputs:  
Calls: UNIT-BOUNDARY  
>map>control.lisp  
Called by: None  
Description: None

**2.4.2.3.29 (METHOD DRAW ARROW-CONTROL-MEASURE)**  
Definition 29

>map>control.lisp  
Type: Method  
Arguments: (WINDOW ALU)  
Outputs:  
Calls: WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
FIND-CENTER-POINT  
>map>control.lisp  
DRAW-ARROW  
>map>control.lisp  
Called by: None  
Description: None

**2.4.2.3.30 (METHOD ENTER-NEW-CONTROL-MEASURE ARROW-CONTROL-MEASURE)**

Definition 30

>map>control.lisp  
 Type: Method  
 Arguments: (WINDOW)  
 Outputs:  
 Calls: ARROW-CONTROL-MEASURE  
 >map>control.lisp  
 Called by: None  
 Description: None

**2.4.2.3.31 AREA-CONTROL-MEASURE**

Definition 31

>map>control.lisp  
 Type: Function  
 Arguments: (WINDOW &OPTIONAL OLD-NAME (TYPE (CDAR \*AREA-TYPES\*)))  
 (OWN-WINDOW NIL))  
 Outputs:  
 Calls: ROTATABLE-RECTANGLE  
 >map>control.lisp  
 Called by: BATTLE-POSITION  
 >map>control.lisp  
 (METHOD ENTER-NEW-CONTROL-MEASURE AREA-CONTROL-  
 MEASURE)  
 >map>control.lisp  
 AREA-CONTROL-MEASURE  
 >map>control.lisp  
 \*CONTROL-MEASURE-MENU-ITEMS\*  
 >map>control.lisp  
 Description: Objectives, Firing Positions, Assembly Areas

**2.4.2.3.32 BATTLE-POSITION**

Definition 32

>map>control.lisp  
 Type: Function  
 Arguments: (WINDOW &OPTIONAL (UNIT (CDAR \*UNIT-TYPES\*))) (OWN-  
 WINDOW NIL))  
 Outputs:  
 Calls: ROTATABLE-RECTANGLE  
 >map>control.lisp  
 Called by: (METHOD ENTER-NEW-CONTROL-MEASURE BATTLE-POSITION)  
 >map>control.lisp  
 BATTLE-POSITION  
 >map>control.lisp  
 \*CONTROL-MEASURE-MENU-ITEMS\*  
 >map>control.lisp  
 Description: Battle Position

**2.4.2.3.33 LINE-CONTROL-MEASURE**

## Definition 33

`>map>control.lisp`

Type: Function

Arguments: (WINDOW &amp;OPTIONAL NAME (TYPE (CDAR \*LINE-TYPES\*))) (OWN-WINDOW NIL))

Outputs:

Calls: RUBBER-LINE

`>map>control.lisp`

Called by: ARROW-CONTROL-MEASURE

`>map>control.lisp`

UNIT-BOUNDARY

`>map>control.lisp`

(METHOD ENTER-NEW-CONTROL-MEASURE LINE-CONTROL-MEASURE)

`>map>control.lisp`

LINE-CONTROL-MEASURE

`>map>control.lisp`

\*CONTROL-MEASURE-MENU-ITEMS\*

`>map>control.lisp`

Description: Phase Line, Objective, EFLOT, Line of Contact, Line of Departure

**2.4.2.3.34 UNIT-BOUNDARY**

## Definition 34

`>map>control.lisp`

Type: Function

Arguments: (WINDOW &amp;OPTIONAL (UNIT (CDAR \*UNIT-TYPES\*))) (OWN-WINDOW NIL))

Outputs:

Calls: RUBBER-LINE

`>map>control.lisp`

Called by: (METHOD ENTER-NEW-CONTROL-MEASURE UNIT-BOUNDARY)

`>map>control.lisp`

UNIT-BOUNDARY

`>map>control.lisp`

\*CONTROL-MEASURE-MENU-ITEMS\*

`>map>control.lisp`

Description: Unit Boundaries

**2.4.2.3.35 ARROW-CONTROL-MEASURE**

## Definition 35

`>map>control.lisp`

Type: Function

Arguments: (WINDOW &amp;OPTIONAL NAME-DEFAULT TYPE-DEFAULT DRAW-TYPE-DEFAULT (OWN-WINDOW NIL))

Outputs:

Calls: RUBBER-LINE

`>map>control.lisp`

Called by: (METHOD ENTER-NEW-CONTROL-MEASURE ARROW-CONTROL-MEASURE)

>map>control.lisp  
 ARROW-CONTROL-MEASURE  
 >map>control.lisp  
 \*CONTROL-MEASURE-MENU-ITEMS\*  
 >map>control.lisp

Description: Avenue of Approach, Axis of Advance

#### 2.4.2.3.36 'WITH-COLOR-MOUSE

Definition 36

>map>control.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.3.37 WITH-COLOR-MOUSE

Definition 37

>map>control.lisp  
 Type: Macro  
 Arguments: (WINDOW WINDOW-X WINDOW-Y MOUSE-STRING CHARACTER  
 OLD-MOUSE-FORM NEW-MOUSE-FORM LEFT-FORM  
 MIDDLE-FORM RIGHT-FORM EXIT-FORM)  
 Outputs:  
 Calls: WITH-COLOR-MOUSE  
 >map>control.lisp  
 Called by: SINGLE-POINT  
 >map>control.lisp  
 SELECT-POLYGON  
 >map>control.lisp  
 RUBBER-LINE  
 >map>control.lisp  
 ROTATABLE-RECTANGLE  
 >map>control.lisp  
 WITH-COLOR-MOUSE  
 >map>control.lisp  
 Description: None

#### 2.4.2.3.38 ROTATABLE-RECTANGLE

Definition 38

>map>control.lisp  
 Type: Function  
 Arguments: (WINDOW &OPTIONAL (DIRECTION NIL))  
 Outputs:

Calls: PIE  
>map>utilities.lisp  
SAFE-ATAN  
>map>utilities.lisp  
\*OVERLAY-ALU\*  
>map>color-map.lisp  
\*ERASE-OVERLAY-ALU\*  
>map>color-map.lisp  
WITH-COLOR-MOUSE  
>map>control.lisp  
DRAW-ROT-RECT  
>map>control.lisp  
Called by: BATTLE-POSITION  
>map>control.lisp  
AREA-CONTROL-MEASURE  
>map>control.lisp  
Description: None

#### 2.4.2.3.39 DRAW-ROT-RECT

Definition 39

>map>control.lisp  
Type: Function  
Arguments: (WIDTH HEIGHT CENTER-X CENTER-Y ROTATION WINDOW ALU  
DIRECTION ARROW-HEAD-LENGTH  
ARROW-BASE-WIDTH)  
Outputs:  
Calls: WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
Called by: ROTATABLE-RECTANGLE  
>map>control.lisp  
Description: None

#### 2.4.2.3.40 'RUBBER-LINE

Definition 40

>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None



**2.4.2.3.41 RUBBER-LINE**

Definition 41

>map>control.lisp  
Type: Function  
Arguments: (WINDOW &OPTIONAL (LIMIT NIL))  
Outputs:  
Calls: WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
\*OVERLAY-ALU\*  
>map>color-map.lisp  
\*ERASE-OVERLAY-ALU\*  
>map>color-map.lisp  
WITH-COLOR-MOUSE  
>map>control.lisp  
Called by: ARROW-CONTROL-MEASURE  
>map>control.lisp  
UNIT-BOUNDARY  
>map>control.lisp  
LINE-CONTROL-MEASURE  
>map>control.lisp  
MAKE-ZONE  
>saf>cm>zone.lisp  
MAKE-AREA  
>saf>cm>area.lisp  
MAKE-LINE  
>saf>cm>line.lisp  
MAKE-ROUTE  
>saf>cm>route.lisp  
(METHOD INTERVENE SIMNET-AGENT ATTACK)  
>saf>objects>intervention.lisp  
Description: None

**2.4.2.3.42 'SELECT-POLYGON**

Definition 42

>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.3.43 SELECT-POLYGON**

Definition 43

>map>control.lisp  
Type: Function  
Arguments: (WINDOW)  
Outputs:  
Calls: WITH-INTEGGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
\*OVERLAY-ALU\*  
>map>color-map.lisp  
\*ERASE-OVERLAY-ALU\*  
>map>color-map.lisp  
WITH-COLOR-MOUSE  
>map>control.lisp  
Called by: None  
Description: None

**2.4.2.3.44 'SINGLE-POINT**

Definition 44

>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.3.45 SINGLE-POINT**

Definition 45

>map>control.lisp  
Type: Function  
Arguments: (WINDOW)  
Outputs:  
Calls: WITH-COLOR-MOUSE  
>map>control.lisp  
Called by: (METHOD MOVE-CONTROL-MEASURE ZONE)  
>saf>cm>zone.lisp  
(METHOD MOVE-CONTROL-MEASURE AREA)  
>saf>cm>area.lisp  
(METHOD INSERT-POINT-AFTER GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD MOVE-POINT GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD MOVE-CONTROL-MEASURE LINE)  
>saf>cm>line.lisp

```

(METHOD INSERT-POINT-BEFORE LINE)
>saf>cm>line.lisp
(METHOD INSERT-POINT-AFTER LINE)
>saf>cm>line.lisp
(METHOD MOVE-POINT LINE)
>saf>cm>line.lisp
MAKE-POINT
>saf>cm>point.lisp
(METHOD MOVE-POINT CM-POINT)
>saf>cm>point.lisp
(METHOD INSERT-POINT-BEFORE ROUTE)
>saf>cm>route.lisp
(METHOD INSERT-POINT-AFTER ROUTE)
>saf>cm>route.lisp
(METHOD MOVE-POINT ROUTE)
>saf>cm>route.lisp
MOUSE-ON-BRIDGE-APPROACH-POINT
>saf>cm>road-routes.lisp
(METHOD INTERVENE SIMNET-AGENT LAND)
>saf>objects>intervention.lisp
(METHOD INTERVENE SIMNET-AGENT GO-TO-LOCATION)
>saf>objects>intervention.lisp
(METHOD INTERVENE SIMNET-AGENT FOLLOW-VEHICLE)
>saf>objects>intervention.lisp
FACE-DIRECTION
>saf>objects>simnet-agent.lisp
(METHOD SPECIFY-RULES-OF-ENGAGEMENT GUNNER)
>saf>objects>gunner.lisp
BOMB-BUTTON
>saf>network>commands.lisp

```

Description: None

#### 2.4.2.3.46 'DRAW-UNIT-SYMBOL

Definition 46

```

>map>control.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

#### 2.4.2.3.47 DRAW-UNIT-SYMBOL

Definition 47

```

>map>control.lisp
Type: Function
Arguments: (POINT UNIT-TYPE ROTATION WINDOW ALU)
Outputs:

```

**Calls:** PIE  
 >map>utilities.lisp  
 WITH-INTEGER-CONVERSION-MODE  
 >map>utilities.lisp  
 WITH-MAP-GRAPHICS  
 >map>utilities.lisp  
 WITH-FAST-MAP-GRAPHICS  
 >map>utilities.lisp  
 \*ERASE-OVERLAY-ALU\*  
 >map>color-map.lisp  
 VEC-NORMALIZE  
 >map>vectors.lisp  
 VEC-ROTATE  
 >map>vectors.lisp  
 VEC-ADD  
 >map>vectors.lisp  
 VEC-SUB  
 >map>vectors.lisp  
 VEC-SCALE  
 >map>vectors.lisp  
**Called by:** (METHOD DRAW UNIT-BOUNDARY AFTER)  
 >map>control.lisp  
 (METHOD DRAW BATTLE-POSITION AFTER)  
 >map>control.lisp  
**Description:** None

#### 2.4.2.3.48 DRAW-1-SCALLOPED-LINE

Definition 48

>map>control.lisp  
**Type:** Function  
**Arguments:** (POINTS WINDOW ALU)  
**Outputs:**  
**Calls:** PIE  
 >map>utilities.lisp  
 WITH-INTEGER-CONVERSION-MODE  
 >map>utilities.lisp  
 WITH-MAP-GRAPHICS  
 >map>utilities.lisp  
 WITH-FAST-MAP-GRAPHICS  
 >map>utilities.lisp  
 DISTANCE  
 >map>utilities.lisp  
 SAFE-ATAN  
 >map>utilities.lisp  
 VEC-NORMALIZE  
 >map>vectors.lisp  
 VEC-ADD  
 >map>vectors.lisp  
 VEC-SUB  
 >map>vectors.lisp  
 VEC-SCALE  
 >map>vectors.lisp

Called by: (METHOD DRAW LINE-CONTROL-MEASURE)  
>map>control.lisp  
Description: None

#### 2.4.2.3.49 DRAW-2-SCALLOPED-LINES

Definition 49

>map>control.lisp  
Type: Function  
Arguments: (POINTS WINDOW ALU)  
Outputs:  
Calls: PIE  
>map>utilities.lisp  
WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
DISTANCE  
>map>utilities.lisp  
SAFE-ATAN  
>map>utilities.lisp  
VEC-NORMALIZE  
>map>vectors.lisp  
VEC-ROTATE  
>map>vectors.lisp  
VEC-ADD  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
VEC-SCALE  
>map>vectors.lisp  
Called by: (METHOD DRAW LINE-CONTROL-MEASURE)  
>map>control.lisp  
Description: None

#### 2.4.2.3.50 FIND-CENTER-POINT

Definition 50

>map>control.lisp  
Type: Function  
Arguments: (POINTS)  
Outputs:  
Calls: DISTANCE  
>map>utilities.lisp  
SAFE-ATAN  
>map>utilities.lisp  
VEC-SUB  
>map>vectors.lisp  
VEC-SCALE  
>map>vectors.lisp

Called by: (METHOD DRAW ARROW-CONTROL-MEASURE)  
>map>control.lisp  
(METHOD DRAW UNIT-BOUNDARY AFTER)  
>map>control.lisp  
Description: None

#### 2.4.2.3.51 DRAW-ARROW

Definition 51

>map>control.lisp  
Type: Function  
Arguments: (POINTS WINDOW ALU DASHED CROSSED)  
Outputs:  
Calls: PIE  
>map>utilities.lisp  
WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
VEC-ROTATE  
>map>vectors.lisp  
VEC-ADD  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
VEC-SCALE  
>map>vectors.lisp  
Called by: (METHOD DRAW ARROW-CONTROL-MEASURE)  
>map>control.lisp  
Description: None

#### 2.4.2.3.52 'CONTROL-MEASURES-MENU

Definition 52

>map>control.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.3.53 CONTROL-MEASURES-MENU

Definition 53

>map>control.lisp  
Type: Function  
Arguments: (WINDOW)  
Outputs:

Calls: \*CONTROL-MEASURE-MENU-ITEMS\*

>map>control.lisp

Called by: None

Description: None

#### 2.4.2.3.54 'EDIT-CONTROL-MEASURES

Definition 54

>map>control.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.3.55 EDIT-CONTROL-MEASURES

Definition 55

>map>control.lisp

Type: Function

Arguments: (WINDOW)

Outputs:

Calls: CONTROL-MEASURE

>map>control.lisp

Called by: \*CONTROL-MEASURE-MENU-ITEMS\*

>map>control.lisp

Description: None

#### 2.4.2.3.56 'DRAW-ALL-CONTROL-MEASURES

Definition 56

>map>control.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.3.57 DRAW-ALL-CONTROL-MEASURES

Definition 57

>map>control.lisp

Type: Function

Arguments: (WINDOW)

Output:

Calls: \*CONTROL-MEASURES\*  
       >map>control.lisp  
       CONTROL-MEASURE  
       >map>control.lisp  
 Called by: (METHOD EDIT CONTROL-MEASURE)  
       >map>control.lisp  
       DRAW-MAP  
       >map>draw-terrain.lisp  
 Description: None

#### 2.4.2.4 CSU map>draw-wide-curve.lisp

This unit contains two routines that draw lines with more than single pixel widths. The first routine draws lines with a single width. The second routine takes a list of three widths, and tapers each end of the wide line based on the first and third widths.

The method :map-draw-wide-curve, for the graphics-mixin object, is a slightly modified copy of the code for the Symbolics function draw-wide-curve, in the tv package. The key step in this method is finding the intersection points of lines parallel to the given segments, at offsets of half the desired width on either side. This is done by the macro compute-points, defined by a macrolet statement inside the method. This macro computes the intersection points by solving two linear equations in two unknowns. In the special case where the two segments have nearly identical slopes, the determinant of the linear system is near zero, making the results inaccurate. In this case, the intersection point is found simply by taking an offset from the point where the two given segments meet.

##### 2.4.2.4.1 (METHOD MAP-DRAW-WIDE-CURVE GRAPHICS-MIXIN) Definition 1

      >map>draw-wide-curve.lisp  
 Type: Method  
 Arguments: (PX PY CURVE-WIDTH &OPTIONAL END (ALU DRAW))  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

##### 2.4.2.4.2 (METHOD MAP-DRAW-TAPERED-WIDE-CURVE GRAPHICS-MIXIN)

Definition 2

      >map>draw-wide-curve.lisp  
 Type: Method  
 Arguments: (PX PY CURVE-WIDTH-LIST &OPTIONAL END (ALU DRAW))  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None



### 2.4.2.5 CSU map>grids.lisp

This unit contains the routines that draw the UTM grids on the color map display. The UTM location of the lower left and upper right corners of the terrain database are stored in the terrain quadtree, and these routines use these values to determine the UTM grids to draw for the current zoom level and map pan. The UTM (Universal Transverse Mercator) coordinate system is documented in the Defense Mapping Agency document DMA TM 8358.1, entitled "Datums, Ellipsoids, Grids, and Grid Reference Systems".

#### 2.4.2.5.1 (METHOD GRID-INC UTM-GRID-MIXIN)

Definition 1

>map>grids.lisp

Type: Method

Arguments: (&OPTIONAL (GRID-LINES-ON-SCREEN 2))

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.5.2 (METHOD LEFT-X-GRID UTM-GRID-MIXIN)

Definition 2

>map>grids.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.5.3 (METHOD RIGHT-X-GRID UTM-GRID-MIXIN)

Definition 3

>map>grids.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.5.4 (METHOD SW-GRID-WORLDS UTM-GRID-MIXIN)

Definition 4

>map>grids.lisp

Type: Method

Arguments: (LEFT BOTTOM GRID-INC)

Outputs:

Calls: None  
Called by: None  
Description: None

#### 2.4.2.5.5 'DRAW-GRIDS Definition 5

>map>grids.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.5.6 (METHOD DRAW-GRIDS UTM-GRID-MIXIN) Definition 6

>map>grids.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: \*QUAD-TREE\*  
>map>terrain-vars.lisp  
WITH-INTEGGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
\*OVERLAY-ALU\*  
>map>color-map.lisp  
\*SOIL-ALU\*  
>map>color-map.lisp  
Called by: None  
Description: None

#### 2.4.2.6 CSU map>intersection.lisp

This unit contains routines for determining line segment, point and polygon intersections. These include functions for counting and listing intersection points of lines and polygons, determining if a point or segment is inside a polygon, finding the bounding rectangle of a list of points, and determining if a point is on a given line or line segment.

#### 2.4.2.6.1 COUNT-INTERSECTIONS Definition 1

>map>intersection.lisp  
Type: Function  
Arguments: (POLYGON X1 Y1 X2 Y2)

Outputs:  
Calls: None  
Called by: POINT-INSIDE-POLYGON-P  
>map>intersection.lisp  
Description: None

**2.4.2.6.2 'POINT-INSIDE-POLYGON-P**  
Definition 2

>map>intersection.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.6.3 POINT-INSIDE-POLYGON-P**  
Definition 3

>map>intersection.lisp  
Type: Function  
Arguments: (POLYGON X Y)  
Outputs:  
Calls: COUNT-INTERSECTIONS  
>map>intersection.lisp  
Called by: SEGMENT-INSIDE-POLYGON-P  
>map>intersection.lisp  
Description: None

**2.4.2.6.4 'SEGMENT-INSIDE-POLYGON-P**  
Definition 4

>map>intersection.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.6.5 SEGMENT-INSIDE-POLYGON-P**  
Definition 5

>map>intersection.lisp  
Type: Function  
Arguments: (POLYGON X1 Y1 X2 Y2)  
Outputs:  
Calls: POINT-INSIDE-POLYGON-P  
>map>intersection.lisp

Called by: LAKES-THRU  
>saf>cm>water-check.lisp  
CHECK-LAKE-INTERSECTIONS  
>saf>cm>water-check.lisp  
Description: None

**2.4.2.6.6 'SEGMENT-INTERSECTS-POLYGON-P**  
Definition 6

>map>intersection.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.6.7 SEGMENT-INTERSECTS-POLYGON-P**  
Definition 7

>map>intersection.lisp  
Type: Function  
Arguments: (POLYGON X1 Y1 X2 Y2)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.6.8 BOUNDING-RECTANGLE**  
Definition 8

>map>intersection.lisp  
Type: Function  
Arguments: (POINTS)  
Outputs:  
Calls: None  
Called by: POSSIBLE-INTERSECTION  
>map>intersection.lisp  
Description: None

**2.4.2.6.9 'POSSIBLE-INTERSECTION**  
Definition 9

>map>intersection.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.6.10 POSSIBLE-INTERSECTION**

## Definition 10

>map>intersection.lisp  
Type: Function  
Arguments: (POINTS X1 Y1 X2 Y2)  
Outputs:  
Calls: CLIP  
>map>clip.lisp  
BOUNDING-RECTANGLE  
>map>intersection.lisp  
Called by: LAKES-THRU  
>saf>cm>water-check.lisp  
WATER-SEGMENTS-THRU  
>saf>cm>water-check.lisp  
SEGMENT-THRU-RIVER  
>saf>cm>water-check.lisp  
Description: None

**2.4.2.6.11 'POINT-SEGMENT-INTERSECTION**

## Definition 11

>map>intersection.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.6.12 POINT-SEGMENT-INTERSECTION**

## Definition 12

>map>intersection.lisp  
Type: Function  
Arguments: (POINTS P1 P2)  
Outputs:  
Calls: DISTANCE  
>map>utilities.lisp  
POINT-LINE-INTERSECTION  
>map>intersection.lisp  
Called by: None  
Description: None

**2.4.2.6.13 'POINT-LINE-INTERSECTION**

## Definition 13

>map>intersection.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:

Calls: None  
Called by: None  
Description: None

#### 2.4.2.6.14 POINT-LINE-INTERSECTION

Definition 14

>map>intersection.lisp  
Type: Function  
Arguments: (X1 Y1 X2 Y2 P1 P2)  
Outputs:  
Calls: PIE  
>map>utilities.lisp  
VEC-NORMALIZE  
>map>vectors.lisp  
VEC-ROTATE  
>map>vectors.lisp  
VEC-ADD  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
Called by: POINT-SEGMENT-INTERSECTION  
>map>intersection.lisp  
CALCULATE-POINT-LINE-INTERSECTION  
>saf>cm>road-routes.lisp  
Description: None

#### 2.4.2.7 CSU map>legend.lisp

This unit contains the routines for drawing the legend information on the map display. A separate window type is defined for the legend. Functions in this CSU define the legend-window flavor, and methods for drawing boxes and lines, the legend scale-line, legend buildings, a legend bridge, and a legend contour-line.

##### 2.4.2.7.1 'LEGEND-WINDOW

Definition 1

>map>legend.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.7.2 LEGEND-WINDOW**

Definition 2

&gt;map&gt;legend.lisp

Type: Flavor

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.7.3 (METHOD INIT LEGEND-WINDOW AFTER)**

Definition 3

&gt;map&gt;legend.lisp

Type: Method

Arguments: (IGNORE)

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.7.4 (METHOD ERASE LEGEND-WINDOW)**

Definition 4

&gt;map&gt;legend.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.7.5 (METHOD SET-LEGEND-POSITIONS LEGEND-WINDOW)**

Definition 5

&gt;map&gt;legend.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.7.6 'DRAW-LEGEND**

Definition 6

&gt;map&gt;legend.lisp

Type: EXPORT

Arguments: ()

Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.7.7 (METHOD DRAW-LEGEND LEGEND-WINDOW)

Definition 7

```

    >map>legend.lisp
  Type: Method
  Arguments: (&OPTIONAL (CONTOURS NIL))
  Outputs:
  Calls: *QUAD-TREE*
    >map>terrain-vars.lisp
    *CURRENT-ZOOM-LEVEL*
    >map>zoom-levels.lisp
    SCALE-STRING
    >map>zoom-levels.lisp
    MAJOR-CONTOUR-LINE-INTERVAL
    >map>zoom-levels.lisp
    MINOR-CONTOUR-LINE-INTERVAL
    >map>zoom-levels.lisp
    LEGEND-SIZE
    >map>zoom-levels.lisp
    LEGEND-LENGTH
    >map>zoom-levels.lisp
    *CURRENT-ZOOM-LEVEL*
    >map>zoom-levels.lisp
    *TREE-ALU*
    >map>color-map.lisp
    *SOIL-ROAD-ALU*
    >map>color-map.lisp
    *SOIL-WATER-ALU*
    >map>color-map.lisp
    *SOIL-MUCK-ALU*
    >map>color-map.lisp
    *LOW-CONTOUR-ALU*
    >map>color-map.lisp
    *HIGH-CONTOUR-ALU*
    >map>color-map.lisp
    *LEGEND-TEXT-ALU*
    >map>color-map.lisp
    DRAW-LEGEND-BOX-AND-LINE
    >map>legend.lisp
    DRAW-LEGEND-SCALE-LINE
    >map>legend.lisp
    DRAW-LEGEND-BUILDINGS
    >map>legend.lisp
    DRAW-LEGEND-BRIDGE
    >map>legend.lisp
    DRAW-LEGEND-CONTOUR-LINE
    >map>legend.lisp
  
```



Called by: None  
Description: None

#### 2.4.2.7.8 DRAW-LEGEND-BOX-AND-LINE

Definition 8

>map>legend.lisp  
Type: Function  
Arguments: (STRING X1 Y X2 STREAM LINE-ALU)  
Outputs:  
Calls: \*LEGEND-TEXT-ALU\*  
>map>color-map.lisp  
Called by: (METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
Description: None

#### 2.4.2.7.9 DRAW-LEGEND-SCALE-LINE

Definition 9

>map>legend.lisp  
Type: Function  
Arguments: (LENGTH X Y STREAM ALU)  
Outputs:  
Calls: None  
Called by: (METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
Description: None

#### 2.4.2.7.10 DRAW-LEGEND-BUILDINGS

Definition 10

>map>legend.lisp  
Type: Function  
Arguments: (X Y STREAM)  
Outputs:  
Calls: \*OBJECT-ALU\*  
>map>color-map.lisp  
Called by: (METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
Description: None

#### 2.4.2.7.11 DRAW-LEGEND-BRIDGE

Definition 11

>map>legend.lisp  
Type: Function  
Arguments: (X Y STREAM ALU)  
Outputs:  
Calls: None

Called by: (METHOD DRAW-LEGEND LEGEND-WINDOW)  
 >map>legend.lisp  
 Description: None

#### 2.4.2.7.12 DRAW-LEGEND-CONTOUR-LINE Definition 12

>map>legend.lisp  
 Type: Function  
 Arguments: (X Y STREAM ALU)  
 Outputs:  
 Calls: None  
 Called by: (METHOD DRAW-LEGEND LEGEND-WINDOW)  
 >map>legend.lisp  
 Description: None

#### 2.4.2.8 CSU map>quadtree-search.lisp

This unit contains the routines that perform the searches in the terrain database quadtree structure. Given an area of interest, the feature indices of the quadtree nodes that are partially or totally within that area are found. The overall quadtree approach is described in the BBN technical report *Terrain Reasoning in the Simnet Semi-Automated Forces System*.

#### 2.4.2.8.1 'QUADS-TO-DRAW Definition 1

>map>quadtree-search.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.8.2 QUADS-TO-DRAW Definition 2

>map>quadtree-search.lisp  
 Type: Function  
 Arguments: (QUAD-TREE STREAM)  
 Outputs:  
 Calls: GET-QUAD-NODES  
 >map>quadtree-search.lisp  
 Called by: DRAW-MAP  
 >map>draw-terrain.lisp  
 DRAW-MAP  
 >saf>sys>update-process.lisp  
 Description: None

**2.4.2.8.3 'GET-QUAD-NODES**

Definition 3

>map>quadtree-search.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.8.4 GET-QUAD-NODES**

Definition 4

>map>quadtree-search.lisp  
Type: Function  
Arguments: (QUAD-NODES X Y SQUARE-SIZE X-MIN Y-MIN X-MAX Y-MAX)  
Outputs:  
Calls: GET-QUAD-NODES  
>map>quadtree-search.lisp  
GET-THIS-NODE  
>map>quadtree-search.lisp  
Called by: HEIGHT-AT-POINT  
>map>draw-terrain.lisp  
GET-QUAD-NODES  
>map>quadtree-search.lisp  
QUADS-TO-DRAW  
>map>quadtree-search.lisp  
GET-QUADS-PASSED-THRU  
>saf>cm>water-check.lisp  
GET-NEIGHBOR-QUAD-ROADS  
>saf>cm>road-routes.lisp  
Description: None

**2.4.2.8.5 GET-THIS-NODE**

Definition 5

>map>quadtree-search.lisp  
Type: Function  
Arguments: (QUAD-NODE X Y SQUARE-SIZE X-MIN Y-MIN X-MAX Y-MAX)  
Outputs:  
Calls: CLIP  
>map>clip.lisp  
Called by: GET-QUAD-NODES  
>map>quadtree-search.lisp  
Description: None

### 2.4.2.9 CSU map>scalable-window.lisp

This unit defines the map display window, and contains routines that manipulate that window, such as pan and zoom. It also contains routines that convert from terrain database world coordinates to color screen (pixel) coordinates. Notice the whopper :draw-triangle, which is needed to correct a limitation of the Symbolics draw-triangle function. The whopper checks to see if any two points of the triangle are equal, and, if so, draws the appropriate line segment, a "degenerate" triangle.

#### 2.4.2.9.1 'SCALABLE-WINDOW

Definition 1

```
>map>scalable-window.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.4.2.9.2 SCALABLE-WINDOW

Definition 2

```
>map>scalable-window.lisp
Type: Flavor
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.4.2.9.3 (METHOD INIT SCALABLE-WINDOW AFTER)

Definition 3

```
>map>scalable-window.lisp
Type: Method
Arguments: (&REST IGNORE)
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.4.2.9.4 (METHOD UPDATE SCALABLE-WINDOW)

Definition 4

```
>map>scalable-window.lisp
Type: Method
Arguments: ()
Outputs:
```

Calls: \*QUAD-TREE\*  
>map>terrain-vars.lisp  
\*ZOOM-LEVELS\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*ZOOM-LEVELS\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
MAKE-COLOR-ALUS  
>map>color-map.lisp  
Called by: None  
Description: None

#### 2.4.2.9.5 (METHOD CLEAR-COORDS SCALABLE-WINDOW) Definition 5

>map>scalable-window.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.9.6 (METHOD NEW-SCALE-INTERNAL SCALABLE-WINDOW) Definition 6

>map>scalable-window.lisp  
Type: Method  
Arguments: (XTRANS YTRANS &OPTIONAL (SCALE 1))  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.9.7 (METHOD NEW-SCALE SCALABLE-WINDOW) Definition 7

>map>scalable-window.lisp  
Type: Method  
Arguments: (XTRANS YTRANS &OPTIONAL (SCALE 1))  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.9.8 (METHOD NEW-SCALE SCALABLE-WINDOW BEFORE)**

Definition 8

&gt;map&gt;scalable-window.lisp

Type: Method

Arguments: (XTRANS YTRANS &amp;OPTIONAL (SCALE 1))

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.9.9 (METHOD NEW-SCALE SCALABLE-WINDOW AFTER)**

Definition 9

&gt;map&gt;scalable-window.lisp

Type: Method

Arguments: (XTRANS YTRANS &amp;OPTIONAL (SCALE 1))

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.9.10 (METHOD DRAW-REGION SCALABLE-WINDOW)**

Definition 10

&gt;map&gt;scalable-window.lisp

Type: Method

Arguments: ()

Outputs:

Calls: \*QUAD-TREE\*

&gt;map&gt;terrain-vars.lisp

WITH-INTEGGER-CONVERSION-MODE

&gt;map&gt;utilities.lisp

WITH-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

WITH-FAST-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

Called by: None

Description: None

**2.4.2.9.11 'WINDOW-SCALE**

Definition 11

&gt;map&gt;scalable-window.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.9.12 (METHOD WINDOW-SCALE SCALABLE-WINDOW)**

Definition 12

```
>map>scalable-window.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.4.2.9.13 (METHOD SOUTH-WEST-CORNER SCALABLE-WINDOW)**

Definition 13

```
>map>scalable-window.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.4.2.9.14 (METHOD SCALED-HEIGHT SCALABLE-WINDOW)**

Definition 14

```
>map>scalable-window.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.4.2.9.15 (METHOD SCALED-WIDTH SCALABLE-WINDOW)**

Definition 15

```
>map>scalable-window.lisp
Type: Method
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

**2.4.2.9.16 'WORLD-EDGES**

Definition 16

```
>map>scalable-window.lisp
Type: EXPORT
Arguments: ()
```

Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.9.17 (METHOD WORLD-EDGES SCALABLE-WINDOW)

Definition 17

>map>scalable-window.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.9.18 'CURRENT-CENTER

Definition 18

>map>scalable-window.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.9.19 (METHOD CURRENT-CENTER SCALABLE-WINDOW)

Definition 19

>map>scalable-window.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.9.20 'PAN-TO-NEW-POINT

Definition 20

>map>scalable-window.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None



**2.4.2.9.21 (METHOD PAN-TO-NEW-POINT SCALABLE-WINDOW)**

Definition 21

>map>scalable-window.lisp  
Type: Method  
Arguments: (X Y &OPTIONAL (NEW-SCALE NIL))  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.9.22 'RESCALE**

Definition 22

>map>scalable-window.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.9.23 (METHOD RESCALE SCALABLE-WINDOW)**

Definition 23

>map>scalable-window.lisp  
Type: Method  
Arguments: (NEW-SCALE &OPTIONAL LEGEND-WINDOW)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.9.24 'RESCALE-FROM-MENU**

Definition 24

>map>scalable-window.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.9.25 (METHOD RESCALE-FROM-MENU SCALABLE-WINDOW)**

Definition 25

>map>scalable-window.lisp  
Type: Method  
Arguments: (&OPTIONAL LEGEND-WINDOW)

Outputs:

Calls: \*ZOOM-LEVELS\*  
 >map>zoom-levels.lisp  
 \*CURRENT-ZOOM-LEVEL\*  
 >map>zoom-levels.lisp  
 SCALE-STRING  
 >map>zoom-levels.lisp  
 CURRENT-SCALE  
 >map>zoom-levels.lisp  
 CURRENT-ANCHOR-X  
 >map>zoom-levels.lisp  
 CURRENT-ANCHOR-Y  
 >map>zoom-levels.lisp  
 \*ZOOM-LEVELS\*  
 >map>zoom-levels.lisp  
 \*CURRENT-ZOOM-LEVEL\*  
 >map>zoom-levels.lisp

Called by: None

Description: None

#### 2.4.2.9.26 'ZOOM-TO

Definition 26

>map>scalable-window.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.9.27 (METHOD ZOOM-TO SCALABLE-WINDOW)

Definition 27

>map>scalable-window.lisp  
 Type: Method  
 Arguments: (ZOOM-LEVEL &OPTIONAL LEGEND-WINDOW)  
 Outputs:  
 Calls: \*CURRENT-ZOOM-LEVEL\*  
 >map>zoom-levels.lisp  
 CURRENT-SCALE  
 >map>zoom-levels.lisp  
 CURRENT-ANCHOR-X  
 >map>zoom-levels.lisp  
 CURRENT-ANCHOR-Y  
 >map>zoom-levels.lisp  
 \*CURRENT-ZOOM-LEVEL\*  
 >map>zoom-levels.lisp  
 Called by: None  
 Description: None

**2.4.2.9.28 'ZOOM-IN**

Definition 28

>map>scalable-window.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.9.29 (METHOD ZOOM-IN SCALABLE-WINDOW)**

Definition 29

>map>scalable-window.lisp  
Type: Method  
Arguments: (X Y &OPTIONAL LEGEND-WINDOW)  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
CURRENT-SCALE  
>map>zoom-levels.lisp  
CURRENT-ANCHOR-X  
>map>zoom-levels.lisp  
CURRENT-ANCHOR-Y  
>map>zoom-levels.lisp  
NEXT-ZOOM-IN  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: None  
Description: None

**2.4.2.9.30 'ZOOM-OUT**

Definition 30

>map>scalable-window.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.9.31 (METHOD ZOOM-OUT SCALABLE-WINDOW)**

Definition 31

>map>scalable-window.lisp  
Type: Method  
Arguments: (X Y &OPTIONAL LEGEND-WINDOW)  
Outputs:

Calls: \*CURRENT-ZOOM-LEVEL\*

>map>zoom-levels.lisp

CURRENT-SCALE

>map>zoom-levels.lisp

CURRENT-ANCHOR-X

>map>zoom-levels.lisp

CURRENT-ANCHOR-Y

>map>zoom-levels.lisp

NEXT-ZOOM-OUT

>map>zoom-levels.lisp

\*CURRENT-ZOOM-LEVEL\*

>map>zoom-levels.lisp

Called by: None

Description: None

#### 2.4.2.9.32 'ZOOM-IN-AROUND-CENTER

Definition 32

>map>scalable-window.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.9.33 (METHOD ZOOM-IN-AROUND-CENTER SCALABLE-WINDOW)

Definition 33

>map>scalable-window.lisp

Type: Method

Arguments: (&OPTIONAL LEGEND-WINDOW)

Outputs:

Calls: \*CURRENT-ZOOM-LEVEL\*

>map>zoom-levels.lisp

CURRENT-SCALE

>map>zoom-levels.lisp

CURRENT-ANCHOR-X

>map>zoom-levels.lisp

CURRENT-ANCHOR-Y

>map>zoom-levels.lisp

NEXT-ZOOM-IN

>map>zoom-levels.lisp

\*CURRENT-ZOOM-LEVEL\*

>map>zoom-levels.lisp

Called by: None

Description: None

**2.4.2.9.34 'ZOOM-OUT-AROUND-CENTER**

Definition 34

>map>scalable-window.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.9.35 (METHOD ZOOM-OUT-AROUND-CENTER SCALABLE-WINDOW)**

Definition 35

>map>scalable-window.lisp  
Type: Method  
Arguments: (&OPTIONAL LEGEND-WINDOW)  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
CURRENT-SCALE  
>map>zoom-levels.lisp  
CURRENT-ANCHOR-X  
>map>zoom-levels.lisp  
CURRENT-ANCHOR-Y  
>map>zoom-levels.lisp  
NEXT-ZOOM-OUT  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: None  
Description: None

**2.4.2.9.36 'ON-TERRAIN-P**

Definition 36

>map>scalable-window.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.9.37 (METHOD ON-TERRAIN-P SCALABLE-WINDOW)**

Definition 37

>map>scalable-window.lisp  
Type: Method  
Arguments: (X Y)

Outputs:

Calls: \*QUAD-TREE\*

>map>terrain-vars.lisp

Called by: None

Description: None

#### 2.4.2.9.38 'ON-SCREEN-P

Definition 38

>map>scalable-window.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.9.39 (METHOD ON-SCREEN-P SCALABLE-WINDOW)

Definition 39

>map>scalable-window.lisp

Type: Method

Arguments: (X Y)

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.9.40 (DRAW-TRIANGLE SCALABLE-WINDOW)

Definition 40

>map>scalable-window.lisp

Type: DEFWHOPPER

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.9.41 (METHOD MOUSE-TO-WORLD SCALABLE-WINDOW)

Definition 41

>map>scalable-window.lisp

Type: Method

Arguments: (MOUSE-X MOUSE-Y)

Outputs:

Calls: WITH-INTEGER-CONVERSION-MODE  
 >map>utilities.lisp  
 WITH-MAP-GRAPHICS  
 >map>utilities.lisp  
 SCREEN-TO-WORLD  
 >map>utilities.lisp

Called by: None

Description: None

#### 2.4.2.9.42 (METHOD WORLD-TO-MOUSE SCALABLE-WINDOW)

Definition 42

>map>scalable-window.lisp  
 Type: Method  
 Arguments: (WORLD-X WORLD-Y)  
 Outputs:  
 Calls: WITH-INTEGER-CONVERSION-MODE  
 >map>utilities.lisp  
 WITH-MAP-GRAPHICS  
 >map>utilities.lisp  
 WORLD-TO-SCREEN  
 >map>utilities.lisp

Called by: None

Description: None

#### 2.4.2.10 CSU map>terrain-vars.lisp

This unit contains all of the definitions of the terrain database structures, as well as global variables that hold these structures for a loaded terrain database. The quadtree structure is the top level terrain structure, which points to the quadtree node structures. These nodes point to feature structures. The indices in the feature structures point to individual terrain features in the various feature arrays.

##### 2.4.2.10.1 '(\*ROAD-SEGMENT-ARRAY\* \*ROAD-INTERSECTION-ARRAY\* \*RAIL-SEGMENT-ARRAY\* \*BRIDGE-ARRAY\*

Definition 1

\*TREES-ARRAY\* \*CONTOUR-ARRAY\* \*OBJECT-ARRAY\* \*CANOPY-ARRAY\*  
 \*CANOPY-TRIANGLES\*  
 \*WATER-SEGMENT-ARRAY\* \*WATER-INTERSECTION-ARRAY\* \*WATER-AREA-ARRAY\* \*WATER-AREA-TRIANGLES\*)  
 >map>terrain-vars.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.10.2 \*ROAD-SEGMENT-ARRAY\***

Definition 2

```

>map>terrain-vars.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: DRAW-ALL-ROADS
>map>draw-terrain.lisp
DRAW-ROADS
>map>draw-terrain.lisp
FIND-NEAREST-BRIDGE
>saf>cm>road-routes.lisp
GET-BRIDGE-POINTS
>saf>cm>road-routes.lisp
DRAW-EXPANDED-ROUTE-CORE
>saf>cm>road-routes.lisp
ROAD-SEGMENTS-FROM-INTERSECTIONS
>saf>cm>road-routes.lisp
FIND-ROAD-DIRECTION
>saf>cm>road-routes.lisp
CALCULATE-ROUTE-DISTANCE
>saf>cm>road-routes.lisp
ROUTE-INTERSECTION
>saf>cm>road-routes.lisp
FIND-NEAREST-ROAD-SEGMENT
>saf>cm>road-routes.lisp
EXPAND-ROUTE-INTO-POINTS
>saf>cm>route-finder.lisp
EXPAND-FIRST-ROUTE
>saf>cm>route-finder.lisp
Description: None

```

**2.4.2.10.3 \*ROAD-INTERSECTION-ARRAY\***

Definition 3

```

>map>terrain-vars.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: GET-BRIDGE-POINTS
>saf>cm>road-routes.lisp
ROAD-SEGMENTS-FROM-INTERSECTIONS
>saf>cm>road-routes.lisp
CALCULATE-ROUTE-DISTANCE
>saf>cm>road-routes.lisp
FIND-ROAD-INTERSECTIONS
>saf>cm>road-routes.lisp
FIND-NEAREST-INTERSECTION
>saf>cm>road-routes.lisp
GET-ROAD-POINT

```



```
>saf>cm>road-routes.lisp
EXPAND-ROUTE-INTO-POINTS
>saf>cm>route-finder.lisp
DISTANCE-BETWEEN-INTERSECTIONS
>saf>cm>route-finder.lisp
EXPAND-FIRST-ROUTE
>saf>cm>route-finder.lisp
FIND-ROUTE
>saf>cm>route-finder.lisp
```

Description: None

#### 2.4.2.10.4 \*TREES-ARRAY\*

Definition 4

```
>map>terrain-vars.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: DRAW-TREES
           >map>draw-terrain.lisp
Description: None
```

#### 2.4.2.10.5 \*CONTOUR-ARRAY\*

Definition 5

```
>map>terrain-vars.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: HEIGHT-AT-POINT
           >map>draw-terrain.lisp
           DRAW-ALL-CONTOURS
           >map>draw-terrain.lisp
           DRAW-CONTOURS
           >map>draw-terrain.lisp
Description: None
```

#### 2.4.2.10.6 \*OBJECT-ARRAY\*

Definition 6

```
>map>terrain-vars.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: DRAW-OBJECTS
           >map>draw-terrain.lisp
Description: None
```

**2.4.2.10.7 \*CANOPY-ARRAY\***

Definition 7

>map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.10.8 \*CANOPY-TRIANGLES\***

Definition 8

>map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-ALL-CANOPIES  
>map>draw-terrain.lisp  
DRAW-CANOPY-TRIANGLES  
>map>draw-terrain.lisp  
Description: None

**2.4.2.10.9 \*WATER-SEGMENT-ARRAY\***

Definition 9

>map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-ALL-RIVERS  
>map>draw-terrain.lisp  
DRAW-WATER  
>map>draw-terrain.lisp  
FIND-RIVER-BEND-POINTS  
>saf>cm>water-avoidance.lisp  
SKIRT-RIVER-BEND  
>saf>cm>water-avoidance.lisp  
CROSSING-LOCATION  
>saf>cm>water-avoidance.lisp  
FIND-RIVER-POINTS  
>saf>cm>water-avoidance.lisp  
SKIRT-RIVER  
>saf>cm>water-avoidance.lisp  
FIND-SEGMENT-CROSS-POINTS  
>saf>cm>water-avoidance.lisp  
INTERSECTION-DIRECTION  
>saf>cm>water-avoidance.lisp  
EXTEND-BRIDGE

```
>saf>cm>water-avoidance.lisp  
EXTEND-INTERSECTION  
>saf>cm>water-avoidance.lisp  
FOLLOW-WATER-SEGMENTS  
>saf>cm>water-avoidance.lisp  
WATER-SEGMENTS-THRU  
>saf>cm>water-check.lisp  
SEGMENT-THRU-RIVER  
>saf>cm>water-check.lisp
```

Description: None

#### 2.4.2.10.10 \*WATER-INTERSECTION-ARRAY\*

Definition 10

```
>map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INTERSECTION-DIRECTION  
>saf>cm>water-avoidance.lisp  
FIND-WATER-INTERSECTIONS  
>saf>cm>water-avoidance.lisp  
FOLLOW-WATER-SEGMENTS  
>saf>cm>water-avoidance.lisp
```

Description: None

#### 2.4.2.10.11 \*BRIDGE-ARRAY\*

Definition 11

```
>map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-BRIDGES  
>map>draw-terrain.lisp  
FOLLOW-WATER-SEGMENTS  
>saf>cm>water-avoidance.lisp
```

Description: None

#### 2.4.2.10.12 \*RAIL-SEGMENT-ARRAY\*

Definition 12

```
>map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None
```

Called by: DRAW-ALL-RAILS  
>map>draw-terrain.lisp  
DRAW-RAILS  
>map>draw-terrain.lisp

Description: None

#### 2.4.2.10.13 \*WATER-AREA-ARRAY\*

Definition 13

>map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: SKIRT-LAKE  
>saf>cm>water-avoidance.lisp  
LAKES-THRU  
>saf>cm>water-check.lisp  
SEGMENT-THRU-LAKE  
>saf>cm>water-check.lisp

Description: None

#### 2.4.2.10.14 \*WATER-AREA-TRIANGLES\*

Definition 14

>map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: DRAW-ALL-WATER-AREAS  
>map>draw-terrain.lisp  
DRAW-WATER-TRIANGLES  
>map>draw-terrain.lisp  
LAKES-THRU  
>saf>cm>water-check.lisp  
SEGMENT-THRU-LAKE  
>saf>cm>water-check.lisp

Description: None

#### 2.4.2.10.15 \*X-ORIGIN\*

Definition 15

>map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.10.16            \*X-MAXIMUM\***  
Definition 16

          >map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.10.17            \*Y-ORIGIN\***  
Definition 17

          >map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.10.18            \*Y-MAXIMUM\***  
Definition 18

          >map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.10.19            \*COLOR-MAP\***  
Definition 19

          >map>terrain-vars.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.10.20            \*\*QUAD-TREE\***  
Definition 20

          >map>terrain-vars.lisp  
Type: EXPORT  
Arguments: ()

Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.10.21      \*QUAD-TREE\*

Definition 21

```

>map>terrain-vars.lisp
Type: Variable
Arguments: ()
Outputs:
Calls: None
Called by: HEIGHT-AT-POINT
>map>draw-terrain.lisp
DRAW-MAP
>map>draw-terrain.lisp
(METHOD DRAW-GRIDS UTM-GRID-MIXIN)
>map>grids.lisp
(METHOD UPDATE UTM-GRID-MIXIN AFTER)
>map>utm-grid-mixin.lisp
(METHOD DRAW-LEGEND LEGEND-WINDOW)
>map>legend.lisp
(METHOD ON-TERRAIN-P SCALABLE-WINDOW)
>map>scalable-window.lisp
(METHOD DRAW-REGION SCALABLE-WINDOW)
>map>scalable-window.lisp
(METHOD UPDATE SCALABLE-WINDOW)
>map>scalable-window.lisp
(METHOD TOP-LEVEL SAF)
>saf>ui>frame.lisp
(METHOD DISPLAY-FWA-PANE BMI)
>saf>bmi>bmi-frame.lisp
DRAW-MAP
>saf>sys>update-process.lisp
UPDATE-TOP-LEVEL-AUX
>saf>sys>update-process.lisp
MAKE-AIRPORTS
>saf>bmi>airport.lisp
GET-QUADS-IN-REGION
>saf>cm>water-avoidance.lisp
FOLLOW-WATER-SEGMENTS
>saf>cm>water-avoidance.lisp
WATER-THRU
>saf>cm>water-check.lisp
ALL-WIDE-SEGMENTS-THRU-WATER
>saf>cm>water-check.lisp
SEGMENT-THRU-WATER
>saf>cm>water-check.lisp
ANY-WIDE-SEGMENT-THRU-WATER
>saf>cm>water-check.lisp
FIND-NEAREST-BRIDGE
>saf>cm>road-routes.lisp

```

```

GET-NEIGHBOR-QUAD-ROADS
>saf>cm>road-routes.lisp
FIND-NEAREST-ROAD-SEGMENT
>saf>cm>road-routes.lisp
FIND-NEAREST-INTERSECTION
>saf>cm>road-routes.lisp

```

Description: None

#### 2.4.2.10.22 '(QUAD-TREE-DB-NAME QUAD-TREE-VERSION

Definition 22

```

QUAD-TREE-EXTRACTION-DATE
QUAD-TREE-COMMENTS
QUAD-TREE-MAP-SHEETS
QUAD-TREE-NODES
QUAD-TREE-SIZE
QUAD-TREE-RESOLUTION
QUAD-TREE-X
QUAD-TREE-Y
QUAD-TREE-MAX-X
QUAD-TREE-MAX-Y
QUAD-TREE-UTM-SW-CORNER
QUAD-TREE-UTM-NE-CORNER
QUAD-TREE-MIN-Z
QUAD-TREE-MAX-Z
QUAD-TREE-ZOOM-LEVELS
QUAD-TREE-COLOR-MAP)

```

```
>map>terrain-vars.lisp
```

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.10.23 QUAD-TREE

Definition 23

```
>map>terrain-vars.lisp
```

Type: DEFSTRUCT

Arguments: ()

Outputs:

Calls: None

Called by: QUAD-TREE-P

```
>map>terrain-vars.lisp
```

```
COPY-QUAD-TREE
```

```
>map>terrain-vars.lisp
```

```
MAKE-QUAD-TREE
```

```
>map>terrain-vars.lisp
```

Description: None

**2.4.2.10.24 QUAD-TREE-DEFAULT**

Definition 24

>map>terrain-vars.lisp  
Type: DEFSTRUCT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: QUAD-TREE-DEFAULT-P  
>map>terrain-vars.lisp  
COPY-QUAD-TREE-DEFAULT  
>map>terrain-vars.lisp  
MAKE-QUAD-TREE-DEFAULT  
>map>terrain-vars.lisp  
Description: None

**2.4.2.10.25 '(QUAD-FEATURES QUAD-NW-NODE QUAD-NE-NODE QUAD-SE-NODE QUAD-SW-NODE)**

Definition 25

>map>terrain-vars.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None .  
Description: None

**2.4.2.10.26 QUAD-NODE**

Definition 26

>map>terrain-vars.lisp  
Type: DEFSTRUCT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: QUAD-NODE-P  
>map>terrain-vars.lisp  
COPY-QUAD-NODE  
>map>terrain-vars.lisp  
MAKE-QUAD-NODE  
>map>terrain-vars.lisp  
Description: None



**2.4.2.10.27** '(AREA-ROAD-SEGMENTS AREA-ROAD-INTERSECTIONS

Definition 27

AREA-WATER-SEGMENTS  
AREA-WATER-INTERSECTIONS  
AREA-BRIDGES  
AREA-RAIL-SEGMENTS  
AREA-OBJECTS  
AREA-TREES  
AREA-CONTOUR-LINES  
AREA-CANOPIES  
AREA-CANOPY-TRIANGLES  
AREA-WATER  
AREA-WATER-TRIANGLES)

&gt;map&gt;terrain-vars.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.10.28** FEATURE-NODE

Definition 28

&gt;map&gt;terrain-vars.lisp

Type: DEFSTRUCT

Arguments: ()

Outputs:

Calls: None

Called by: FEATURE-NODE-P

&gt;map&gt;terrain-vars.lisp

COPY-FEATURE-NODE

&gt;map&gt;terrain-vars.lisp

MAKE-FEATURE-NODE

&gt;map&gt;terrain-vars.lisp

Description: None

**2.4.2.10.29** \*FEATURE-LIST\*

Definition 29

&gt;map&gt;terrain-vars.lisp

Type: Variable

Arguments: ()

Outputs:

Calls: BRIDGE

&gt;map&gt;terrain-vars.lisp

Called by: None

Description: None

**2.4.2.10.30** '(SEGMENT-POINTS SEGMENT-WIDTH SEGMENT-HEIGHT SEGMENT-ELEVATION)

Definition 30

>map>terrain-vars.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.10.31** SEGMENT

Definition 31

>map>terrain-vars.lisp  
Type: DEFSTRUCT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: SEGMENT-P  
>map>terrain-vars.lisp  
COPY-SEGMENT  
>map>terrain-vars.lisp  
MAKE-SEGMENT  
>map>terrain-vars.lisp  
Description: None

**2.4.2.10.32** SEGMENT-HEIGHT

Definition 32

>map>terrain-vars.lisp  
Type: Subst  
Arguments: (SEGMENT)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.10.33** SEGMENT-ELEVATION

Definition 33

>map>terrain-vars.lisp  
Type: Subst  
Arguments: (SEGMENT)  
Outputs:  
Calls: None

Called by: HEIGHT-AT-POINT

>map>draw-terrain.lisp  
DRAW-ALL-CONTOURS  
>map>draw-terrain.lisp  
DRAW-CONTOURS  
>map>draw-terrain.lisp

Description: None

#### 2.4.2.10.34 '(NET-POINTS NET-WIDTH

Definition 34

NET-DISTANCE  
NET-FORDABLE  
NET-BRIDGE  
INTERSECTION-POS-X  
INTERSECTION-POS-Y  
INTERSECTION-PAIRS  
INTERSECTION-BRIDGE)  
>map>terrain-vars.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.10.35 NETWORK-SEGMENT

Definition 35

>map>terrain-vars.lisp

Type: DEFSTRUCT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.10.36 NETWORK-INTERSECTION

Definition 36

>map>terrain-vars.lisp

Type: DEFSTRUCT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

### 2.4.2.10.37 '(BRIDGE-POINTS BRIDGE-NODE BRIDGE-WIDTH) Definition 37

```

>map>terrain-vars.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

### 2.4.2.10.38 BRIDGE Definition 38

```

>map>terrain-vars.lisp
Type: DEFSTRUCT
Arguments: ()
Outputs:
Calls: None
Called by: BRIDGE-P
>map>terrain-vars.lisp
COPY-BRIDGE
>map>terrain-vars.lisp
MAKE-BRIDGE
>map>terrain-vars.lisp
*FEATURE-LIST*
>map>terrain-vars.lisp
Description: None

```

### 2.4.2.11 CSU map>utilities.lisp

This unit contains macro definitions used by the terrain drawing routines. These macros are used to transform the terrain database world coordinates to color screen (pixel) coordinates. The macros *with-map-graphics*, *with-fast-map-graphics*, and *with-ultra-fast-map-graphics* encapsulate commonly used drawing code, for convenience and modularity.

This unit also contains general graphics utility routines, such as the Euclidean *distance* function, and a defsubst *near* that determines if two numbers are within a given tolerance. Also included is a function *safe-atan* that returns  $\arctan(y/x)$ , giving the correct value of  $\pi/2$  or  $-\pi/2$  when  $x$  is zero and  $y$  is nonzero. (The Common Lisp *atan* function is required by the language standard to handle these cases properly; *safe-atan*, originally from earlier Zeta-Lisp code, does not rely on this.)

#### 2.4.2.11.1 PIE Definition 1

```

>map>utilities.lisp
Type: Constant
Arguments: ()
Outputs:
Calls: None

```

Called by: ROTATABLE-RECTANGLE

```

>map>control.lisp
DRAW-ARROW
>map>control.lisp
DRAW-2-SCALLOPED-LINES
>map>control.lisp
DRAW-1-SCALLOPED-LINE
>map>control.lisp
DRAW-UNIT-SYMBOL
>map>control.lisp
POINT-LINE-INTERSECTION
>map>intersection.lisp
DRAW-BRIDGE-SYMBOL
>map>vectors.lisp
SAFE-ATAN
>map>utilities.lisp
OFFSET-POINT
>saf>cm>water-avoidance.lisp
EXTEND-SEGMENT
>saf>cm>water-avoidance.lisp
ALL-WIDE-SEGMENTS-THRU-WATER
>saf>cm>water-check.lisp
ANY-WIDE-SEGMENT-THRU-WATER
>saf>cm>water-check.lisp

```

Description: None

#### 2.4.2.11.2 GRAPHICS-TRANSFORM

Definition 2

```

>map>utilities.lisp
Type: DEFSTRUCT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

#### 2.4.2.11.3 WITH-INTEGER-CONVERSION-MODE

Definition 3

```

>map>utilities.lisp
Type: Macro
Arguments: ((WINDOW) &BODY BODY)
Outputs:
Calls: WITH-INTEGER-CONVERSION-MODE
>map>utilities.lisp
Called by: SELECT-POLYGON
>map>control.lisp
RUBBER-LINE
>map>control.lisp
DRAW-ARROW
>map>control.lisp

```

```
DRAW-2-SCALLOPED-LINES
>map>control.lisp
DRAW-1-SCALLOPED-LINE
>map>control.lisp
DRAW-UNIT-SYMBOL
>map>control.lisp
DRAW-ROT-RECT
>map>control.lisp
(METHOD DRAW ARROW-CONTROL-MEASURE)
>map>control.lisp
(METHOD DRAW LINE-CONTROL-MEASURE)
>map>control.lisp
(METHOD DRAW AREA-CONTROL-MEASURE)
>map>control.lisp
DRAW-ALL-CONTOURS
>map>draw-terrain.lisp
DRAW-ALL-CANOPIES
>map>draw-terrain.lisp
DRAW-ALL-RIVERS
>map>draw-terrain.lisp
DRAW-WATER-OR-LAND-TRIANGLES
>map>draw-terrain.lisp
DRAW-ALL-ROADS
>map>draw-terrain.lisp
DRAW-ALL-RAILS
>map>draw-terrain.lisp
DRAW-RAILS
>map>draw-terrain.lisp
DRAW-WATER-OR-LAND-TRIANGLES-MAYBE
>map>draw-terrain.lisp
DRAW-WATER
>map>draw-terrain.lisp
DRAW-CANOPY-TRIANGLES
>map>draw-terrain.lisp
DRAW-OBJECTS
>map>draw-terrain.lisp
DRAW-CONTOURS
>map>draw-terrain.lisp
DRAW-TREES
>map>draw-terrain.lisp
DRAW-ROADS
>map>draw-terrain.lisp
(METHOD DRAW-GRIDS UTM-GRID-MIXIN)
>map>grids.lisp
(METHOD WORLD-TO-MOUSE SCALABLE-WINDOW)
>map>scalable-window.lisp
(METHOD MOUSE-TO-WORLD SCALABLE-WINDOW)
>map>scalable-window.lisp
(METHOD DRAW-REGION SCALABLE-WINDOW)
>map>scalable-window.lisp
DRAW-BRIDGE-SYMBOL
>map>vectors.lisp
DRAW-VEHICLE
>saf>simnet-objects>new-draw-vehicles.lisp
```

```
(METHOD COM-ZOOM-OUT-INTERNAL PVD)
No Source File Record
(METHOD COM-PAN-INTERNAL PVD)
No Source File Record
(METHOD COM-ZOOM-IN-INTERNAL PVD)
No Source File Record
GET-LOCATION-AND-BEARING
>saf>sandbox>utilities.lisp
(METHOD INSERT-POINT-AFTER GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD DELETE-POINT GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD MOVE-POINT GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD PAINT GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD INSERT-POINT-BEFORE LINE)
>saf>cm>line.lisp
(METHOD INSERT-POINT-AFTER LINE)
>saf>cm>line.lisp
(METHOD DELETE-POINT LINE)
>saf>cm>line.lisp
(METHOD MOVE-POINT LINE)
>saf>cm>line.lisp
(METHOD PAINT LINE)
>saf>cm>line.lisp
(METHOD ERASE CM-POINT)
>saf>cm>point.lisp
(METHOD DRAW CM-POINT)
>saf>cm>point.lisp
MAKE-ROUTE
>saf>cm>route.lisp
(METHOD INSERT-POINT-BEFORE ROUTE)
>saf>cm>route.lisp
(METHOD INSERT-POINT-AFTER ROUTE)
>saf>cm>route.lisp
(METHOD DELETE-POINT ROUTE)
>saf>cm>route.lisp
(METHOD MOVE-POINT ROUTE)
>saf>cm>route.lisp
(METHOD PAINT ROUTE)
>saf>cm>route.lisp
GET-BRIDGE-ROUTE
>saf>cm>road-routes.lisp
DRAW-EXPANDED-ROUTE
>saf>cm>road-routes.lisp
GET-ROAD-ROUTE
>saf>cm>road-routes.lisp
(METHOD DRAW-AS-FIRST-POINT CONTROL-MEASURE-POINT)
>saf>cm>control-measure-point.lisp
(METHOD PAINT CONTROL-MEASURE-POINT)
>saf>cm>control-measure-point.lisp
(METHOD ERASE-NAME CONTROL-MEASURE)
>saf>cm>control-measure.lisp
```

```

(METHOD DRAW-NAME CONTROL-MEASURE)
>saf>cm>control-measure.lisp
DRAW-UNIT
>saf>simnet-objects>draw-units.lisp
DRAW-ARTY
>saf>simnet-objects>draw-effects.lisp
DRAW-IMPACT
>saf>simnet-objects>draw-effects.lisp
(METHOD DRAW-MISSILE-IMAGE MISSILE-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-TURRET-IMAGE RD-TURRET-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-TURRET-IMAGE SQ-TURRET-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-IMAGE FIGHTER-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-IMAGE HELO-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
DRAW-FILLED-BOX
>saf>simnet-objects>draw-vehicles.lisp
DRAW-BOX
>saf>simnet-objects>draw-vehicles.lisp
(METHOD HIGHLIGHT SIMNET-AGENT)
>saf>objects>simnet-agent.lisp
DRAW-STEALTH
>saf>sys>utilities.lisp
WITH-ULTRA-FAST-GRAPHICS
>map>utilities.lisp
WITH-MAP-GRAPHICS
>map>utilities.lisp
WITH-INTEGER-CONVERSION-MODE
>map>utilities.lisp

```

Description: None

#### 2.4.2.11.4 'WITH-MAP-GRAPHICS

Definition 4

```

>map>utilities.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

#### 2.4.2.11.5 WITH-MAP-GRAPHICS

Definition 5

```

>map>utilities.lisp
Type: Macro
Arguments: ((WINDOW) &BODY BODY)
Outputs:

```



Calls: WITH-INTEGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
Called by: SELECT-POLYGON  
>map>control.lisp  
RUBBER-LINE  
>map>control.lisp  
DRAW-ARROW  
>map>control.lisp  
DRAW-2-SCALLOPED-LINES  
>map>control.lisp  
DRAW-1-SCALLOPED-LINE  
>map>control.lisp  
DRAW-UNIT-SYMBOL  
>map>control.lisp  
DRAW-ROT-RECT  
>map>control.lisp  
(METHOD DRAW ARROW-CONTROL-MEASURE)  
>map>control.lisp  
(METHOD DRAW LINE-CONTROL-MEASURE)  
>map>control.lisp  
(METHOD DRAW AREA-CONTROL-MEASURE)  
>map>control.lisp  
DRAW-RAILS  
>map>draw-terrain.lisp  
DRAW-ROADS  
>map>draw-terrain.lisp  
(METHOD DRAW-GRIDS UTM-GRID-MIXIN)  
>map>grids.lisp  
(METHOD WORLD-TO-MOUSE SCALABLE-WINDOW)  
>map>scalable-window.lisp  
(METHOD MOUSE-TO-WORLD SCALABLE-WINDOW)  
>map>scalable-window.lisp  
(METHOD DRAW-REGION SCALABLE-WINDOW)  
>map>scalable-window.lisp  
DRAW-BRIDGE-SYMBOL  
>map>vectors.lisp  
DRAW-VEHICLE  
>saf>simnet-objects>new-draw-vehicles.lisp  
(METHOD COM-ZOOM-OUT-INTERNAL PVD)  
No Source File Record  
(METHOD COM-PAN-INTERNAL PVD)  
No Source File Record  
(METHOD COM-ZOOM-IN-INTERNAL PVD)  
No Source File Record  
GET-LOCATION-AND-BEARING  
>saf>sandbox>utilities.lisp  
(METHOD INSERT-POINT-AFTER GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD DELETE-POINT GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD MOVE-POINT GENERIC-AREA)  
>saf>cm>generic-area.lisp

```
(METHOD PAINT GENERIC-AREA)
>saf>cm>generic-area.lisp
(METHOD INSERT-POINT-BEFORE LINE)
>saf>cm>line.lisp
(METHOD INSERT-POINT-AFTER LINE)
>saf>cm>line.lisp
(METHOD DELETE-POINT LINE)
>saf>cm>line.lisp
(METHOD MOVE-POINT LINE)
>saf>cm>line.lisp
(METHOD PAINT LINE)
>saf>cm>line.lisp
(METHOD ERASE CM-POINT)
>saf>cm>point.lisp
(METHOD DRAW CM-POINT)
>saf>cm>point.lisp
MAKE-ROUTE
>saf>cm>route.lisp
(METHOD INSERT-POINT-BEFORE ROUTE)
>saf>cm>route.lisp
(METHOD INSERT-POINT-AFTER ROUTE)
>saf>cm>route.lisp
(METHOD DELETE-POINT ROUTE)
>saf>cm>route.lisp
(METHOD MOVE-POINT ROUTE)
>saf>cm>route.lisp
(METHOD PAINT ROUTE)
>saf>cm>route.lisp
GET-BRIDGE-ROUTE
>saf>cm>road-routes.lisp
DRAW-EXPANDED-ROUTE
>saf>cm>road-routes.lisp
GET-ROAD-ROUTE
>saf>cm>road-routes.lisp
(METHOD DRAW-AS-FIRST-POINT CONTROL-MEASURE-POINT)
>saf>cm>control-measure-point.lisp
(METHOD PAINT CONTROL-MEASURE-POINT)
>saf>cm>control-measure-point.lisp
(METHOD ERASE-NAME CONTROL-MEASURE)
>saf>cm>control-measure.lisp
(METHOD DRAW-NAME CONTROL-MEASURE)
>saf>cm>control-measure.lisp
DRAW-UNIT
>saf>simnet-objects>draw-units.lisp
DRAW-ARTY
>saf>simnet-objects>draw-effects.lisp
DRAW-IMPACT
>saf>simnet-objects>draw-effects.lisp
(METHOD DRAW-MISSILE-IMAGE MISSILE-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-TURRET-IMAGE RD-TURRET-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-TURRET-IMAGE SQ-TURRET-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
```

```
(METHOD DRAW-IMAGE FIGHTER-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-IMAGE HELO-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
DRAW-FILLED-BOX
>saf>simnet-objects>draw-vehicles.lisp
DRAW-BOX
>saf>simnet-objects>draw-vehicles.lisp
(METHOD HIGHLIGHT SIMNET-AGENT)
>saf>objects>simnet-agent.lisp
DRAW-STEALTH
>saf>sys>utilities.lisp
WORLD-TO-SCREEN
>map>utilities.lisp
SCREEN-TO-WORLD
>map>utilities.lisp
WITH-FAST-MAP-Graphics
>map>utilities.lisp
WITH-MAP-Graphics
>map>utilities.lisp
WITH-CORRECT-MAP-Graphics
>saf>simnet-objects>draw-vehicles.lisp
```

Description: None

#### 2.4.2.11.6 'WITH-FAST-MAP-Graphics

Definition 6

```
>map>utilities.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None
```

#### 2.4.2.11.7 WITH-FAST-MAP-Graphics

Definition 7

```
>map>utilities.lisp
Type: Macro
Arguments: ((WINDOW) &BODY BODY)
Outputs:
Calls: WITH-MAP-Graphics
>map>utilities.lisp
WITH-FAST-MAP-Graphics
>map>utilities.lisp
Called by: SELECT-POLYGON
>map>control.lisp
RUBBER-LINE
>map>control.lisp
DRAW-ARROW
>map>control.lisp
```

DRAW-2-SCALLOPED-LINES  
>map>control.lisp  
DRAW-1-SCALLOPED-LINE  
>map>control.lisp  
DRAW-UNIT-SYMBOL  
>map>control.lisp  
DRAW-ROT-RECT  
>map>control.lisp  
(METHOD DRAW ARROW-CONTROL-MEASURE)  
>map>control.lisp  
(METHOD DRAW LINE-CONTROL-MEASURE)  
>map>control.lisp  
(METHOD DRAW AREA-CONTROL-MEASURE)  
>map>control.lisp  
DRAW-RAILS  
>map>draw-terrain.lisp  
DRAW-ROADS  
>map>draw-terrain.lisp  
(METHOD DRAW-GRIDS UTM-GRID-MIXIN)  
>map>grids.lisp  
(METHOD DRAW-REGION SCALABLE-WINDOW)  
>map>scalable-window.lisp  
DRAW-BRIDGE-SYMBOL  
>map>vectors.lisp  
(METHOD INSERT-POINT-AFTER GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD DELETE-POINT GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD MOVE-POINT GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD PAINT GENERIC-AREA)  
>saf>cm>generic-area.lisp  
(METHOD INSERT-POINT-BEFORE LINE)  
>saf>cm>line.lisp  
(METHOD INSERT-POINT-AFTER LINE)  
>saf>cm>line.lisp  
(METHOD DELETE-POINT LINE)  
>saf>cm>line.lisp  
(METHOD MOVE-POINT LINE)  
>saf>cm>line.lisp  
(METHOD PAINT LINE)  
>saf>cm>line.lisp  
(METHOD ERASE CM-POINT)  
>saf>cm>point.lisp  
MAKE-ROUTE  
>saf>cm>route.lisp  
(METHOD INSERT-POINT-BEFORE ROUTE)  
>saf>cm>route.lisp  
(METHOD INSERT-POINT-AFTER ROUTE)  
>saf>cm>route.lisp  
(METHOD DELETE-POINT ROUTE)  
>saf>cm>route.lisp  
(METHOD MOVE-POINT ROUTE)  
>saf>cm>route.lisp

```

(METHOD PAINT ROUTE)
>saf>cm>route.lisp
DRAW-EXPANDED-ROUTE
>saf>cm>road-routes.lisp
(METHOD DRAW-AS-FIRST-POINT CONTROL-MEASURE-POINT)
>saf>cm>control-measure-point.lisp
(METHOD ERASE-NAME CONTROL-MEASURE)
>saf>cm>control-measure.lisp
DRAW-ARTY
>saf>simnet-objects>draw-effects.lisp
DRAW-IMPACT
>saf>simnet-objects>draw-effects.lisp
(METHOD DRAW-MISSILE-IMAGE MISSILE-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-TURRET-IMAGE RD-TURRET-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-TURRET-IMAGE SQ-TURRET-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-IMAGE FIGHTER-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-IMAGE HELO-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
DRAW-FILLED-BOX
>saf>simnet-objects>draw-vehicles.lisp
DRAW-BOX
>saf>simnet-objects>draw-vehicles.lisp
(METHOD HIGHLIGHT SIMNET-AGENT)
>saf>objects>simnet-agent.lisp
DRAW-STEALTH
>saf>sys>utilities.lisp
WITH-FAST-MAP-GRAPHICS
>map>utilities.lisp
WITH-CORRECT-MAP-GRAPHICS
>saf>simnet-objects>draw-vehicles.lisp

```

Description: None

#### 2.4.2.11.8 'SCREEN-TO-WORLD

Definition 8

```

>map>utilities.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

**2.4.2.11.9 SCREEN-TO-WORLD**

Definition 9

&gt;map&gt;utilities.lisp

Type: Macro

Arguments: (WINDOW &amp;REST POINTS)

Outputs:

Calls: WITH-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

SCREEN-TO-WORLD

&gt;map&gt;utilities.lisp

Called by: (METHOD MOUSE-TO-WORLD SCALABLE-WINDOW)

&gt;map&gt;scalable-window.lisp

(METHOD COM-ZOOM-OUT-INTERNAL PVD)

No Source File Record

(METHOD COM-PAN-INTERNAL PVD)

No Source File Record

(METHOD COM-ZOOM-IN-INTERNAL PVD)

No Source File Record

GET-LOCATION-AND-BEARING

&gt;saf&gt;sandbox&gt;utilities.lisp

GET-BRIDGE-ROUTE

&gt;saf&gt;cm&gt;road-routes.lisp

GET-ROAD-ROUTE

&gt;saf&gt;cm&gt;road-routes.lisp

SCREEN-TO-WORLD

&gt;map&gt;utilities.lisp

Description: None

**2.4.2.11.10 'WORLD-TO-SCREEN**

Definition 10

&gt;map&gt;utilities.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.4.2.11.11 WORLD-TO-SCREEN**

Definition 11

&gt;map&gt;utilities.lisp

Type: Macro

Arguments: (WINDOW &amp;REST POINTS)

Outputs:

Calls: WITH-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

WORLD-TO-SCREEN

&gt;map&gt;utilities.lisp

Called by: (METHOD WORLD-TO-MOUSE SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 DRAW-VEHICLE  
 >saf>simnet-objects>new-draw-vehicles.lisp  
 WORLD-TO-SCREEN  
 >map>utilities.lisp

Description: None

#### 2.4.2.11.12 WITH-ULTRA-FAST-GRAPHICS Definition 12

>map>utilities.lisp  
 Type: Macro  
 Arguments: ((STREAM) &BODY BODY)  
 Outputs:  
 Calls: WITH-INTEGGER-CONVERSION-MODE  
 >map>utilities.lisp  
 WITH-ULTRA-FAST-GRAPHICS  
 >map>utilities.lisp  
 Called by: DRAW-ALL-CONTOURS  
 >map>draw-terrain.lisp  
 DRAW-ALL-CANOPIES  
 >map>draw-terrain.lisp  
 DRAW-ALL-RIVERS  
 >map>draw-terrain.lisp  
 DRAW-WATER-OR-LAND-TRIANGLES  
 >map>draw-terrain.lisp  
 DRAW-ALL-ROADS  
 >map>draw-terrain.lisp  
 DRAW-ALL-RAILS  
 >map>draw-terrain.lisp  
 DRAW-RAILS  
 >map>draw-terrain.lisp  
 DRAW-WATER-OR-LAND-TRIANGLES-MAYBE  
 >map>draw-terrain.lisp  
 DRAW-WATER  
 >map>draw-terrain.lisp  
 DRAW-CANOPY-TRIANGLES  
 >map>draw-terrain.lisp  
 DRAW-OBJECTS  
 >map>draw-terrain.lisp  
 DRAW-CONTOURS  
 >map>draw-terrain.lisp  
 DRAW-TREES  
 >map>draw-terrain.lisp  
 DRAW-ROADS  
 >map>draw-terrain.lisp  
 WITH-ULTRA-FAST-GRAPHICS  
 >map>utilities.lisp

Description: None

**2.4.2.11.13 FAST-WORLD-TO-SCREEN**

Definition 13

&gt;map&gt;utilities.lisp

Type: Macro

Arguments: (&amp;REST POINTS)

Outputs:

Calls: TRANSFORM-POINT

&gt;map&gt;utilities.lisp

Called by: DRAW-ALL-CONTOURS

&gt;map&gt;draw-terrain.lisp

DRAW-ALL-CANOPIES

&gt;map&gt;draw-terrain.lisp

DRAW-ALL-RIVERS

&gt;map&gt;draw-terrain.lisp

DRAW-WATER-OR-LAND-TRIANGLES

&gt;map&gt;draw-terrain.lisp

DRAW-ALL-ROADS

&gt;map&gt;draw-terrain.lisp

DRAW-ALL-RAILS

&gt;map&gt;draw-terrain.lisp

DRAW-RAILS

&gt;map&gt;draw-terrain.lisp

DRAW-WATER-OR-LAND-TRIANGLES-MAYBE

&gt;map&gt;draw-terrain.lisp

DRAW-WATER

&gt;map&gt;draw-terrain.lisp

DRAW-CANOPY-TRIANGLES

&gt;map&gt;draw-terrain.lisp

DRAW-OBJECTS

&gt;map&gt;draw-terrain.lisp

DRAW-CONTOURS

&gt;map&gt;draw-terrain.lisp

DRAW-TREES

&gt;map&gt;draw-terrain.lisp

DRAW-ROADS

&gt;map&gt;draw-terrain.lisp

Description: None

**2.4.2.11.14 TRANSFORM-POINT**

Definition 14

&gt;map&gt;utilities.lisp

Type: Function

Arguments: (X Y TRANSFORM)

Outputs:

Calls: None

Called by: DRAW-ALL-CONTOURS

&gt;map&gt;draw-terrain.lisp

DRAW-ALL-CANOPIES

&gt;map&gt;draw-terrain.lisp

DRAW-ALL-RIVERS

&gt;map&gt;draw-terrain.lisp



DRAW-WATER-OR-LAND-TRIANGLES  
 >map>draw-terrain.lisp  
 DRAW-ALL-ROADS  
 >map>draw-terrain.lisp  
 DRAW-ALL-RAILS  
 >map>draw-terrain.lisp  
 DRAW-RAILS  
 >map>draw-terrain.lisp  
 DRAW-WATER-OR-LAND-TRIANGLES-MAYBE  
 >map>draw-terrain.lisp  
 DRAW-WATER  
 >map>draw-terrain.lisp  
 DRAW-CANOPY-TRIANGLES  
 >map>draw-terrain.lisp  
 DRAW-OBJECTS  
 >map>draw-terrain.lisp  
 DRAW-CONTOURS  
 >map>draw-terrain.lisp  
 DRAW-TREES  
 >map>draw-terrain.lisp  
 DRAW-ROADS  
 >map>draw-terrain.lisp  
 FAST-WORLD-TO-SCREEN  
 >map>utilities.lisp

Description: None

#### 2.4.2.11.15 'DISTANCE

Definition 15

>map>utilities.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.11.16 DISTANCE

Definition 16

>map>utilities.lisp  
 Type: Function  
 Arguments: (X1 Y1 X2 Y2)  
 Outputs:  
 Calls: None  
 Called by: FIND-CENTER-POINT  
 >map>control.lisp  
 DRAW-2-SCALLOPED-LINES  
 >map>control.lisp  
 DRAW-1-SCALLOPED-LINE  
 >map>control.lisp  
 HEIGHT-AT-POINT

```

>map>draw-terrain.lisp
POINT-SEGMENT-INTERSECTION
>map>intersection.lisp
FIND-FORMATION-INFO
>saf>sandbox>sandbox.lisp
SORT-CMS
>saf>cm>overlay.lisp
(METHOD CM-INTERSECTION CM-POINT)
>saf>cm>point.lisp
DISTANCE-AROUND-PATH
>saf>cm>water-avoidance.lisp
PRUNE-TO-POINT
>saf>cm>water-avoidance.lisp
FIND-CLOSER-CROSSING
>saf>cm>water-avoidance.lisp
FIND-SEGMENT-CROSS-POINTS
>saf>cm>water-avoidance.lisp
WATER-THRU
>saf>cm>water-check.lisp
FIND-NEAREST-BRIDGE
>saf>cm>road-routes.lisp
GET-BRIDGE-POINTS
>saf>cm>road-routes.lisp
ROUTE-INTERSECTION
>saf>cm>road-routes.lisp
PARALLEL-DISTANCE
>saf>cm>road-routes.lisp
FIND-NEAREST-ROAD-SEGMENT
>saf>cm>road-routes.lisp
FIND-NEAREST-INTERSECTION
>saf>cm>road-routes.lisp
DISTANCE-BETWEEN-INTERSECTIONS
>saf>cm>route-finder.lisp

```

Description: None

#### 2.4.2.11.17 'NEAR

Definition 17

```

>map>utilities.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

#### 2.4.2.11.18 NEAR

Definition 18

```

>map>utilities.lisp
Type: Subst
Arguments: (POINT1 POINT2 THRESHOLD)
Outputs:

```

Calls: None  
 Called by: CALCULATE-POINT-LINE-INTERSECTION  
           >saf>cm>road-routes.lisp  
           EXPAND-ROUTE-INTO-POINTS  
           >saf>cm>route-finder.lisp  
 Description: None

#### 2.4.2.11.19 SAFE-ATAN

Definition 19

          >map>utilities.lisp  
 Type: Function  
 Arguments: (DELTA-Y DELTA-X)  
 Outputs:  
 Calls: PIE  
           >map>utilities.lisp  
 Called by: ROTATABLE-RECTANGLE  
           >map>control.lisp  
           FIND-CENTER-POINT  
           >map>control.lisp  
           DRAW-2-SCALLOPED-LINES  
           >map>control.lisp  
           DRAW-1-SCALLOPED-LINE  
           >map>control.lisp  
 Description: None

#### 2.4.2.12 CSU map>utm-grid-mixin.lisp

This unit contains the routines to convert between terrain database world coordinates and UTM coordinates. First, the letters A through Z are coerced into a vector, by the function `fill-alphabet-array`, to form `*alphabet-array*`. Functions `char-to-coord` and `coord-to-char` use this array to convert between UTM letter prefixes and numerical coordinates for the origin-corner of a lettered UTM square. The flavor `utm-grid-mixin` is then defined, fixing the UTM coordinates of the origin of the PVD map. This flavor has methods `world-to-utm` and `utm-to-world` that convert to and from UTM coordinates.

The unique feature of the UTM system is its ability to use shorter strings to convey less precision, but *without* using a delimiter to separate the x value and the y value. The separation is achieved by dividing the total number of digits, which must be even, by 2. In `utm-to-world`, the form *(case length (2 ...) (4 ...) (6 ...) (8 ...) (10 ...))* does this, inferring the number of digits of precision in use, and separating the x and y values into the variables `x-num` and `y-num`. In `world-to-utm`, the precision is passed explicitly in the variable `size`, with a default value of 3. After the x and y offsets are calculated from map coordinates using `coord-to-char`, they are printed to a string using `format` and then trimmed to the specified size with `substring` calls. Finally, the digit strings are appended with the UTM letters `x-char` and `y-char`, returned by `coord-to-char`, to get the completed UTM string.

For more information on the UTM system, see the Defense Mapping Agency document DMA TM 8358.1, entitled "Datums, Ellipsoids, Grids, and Grid Reference Systems".

**2.4.2.12.1 \*ALPHABET-ARRAY\***

## Definition 1

>map>utm-grid-mixin.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: COORD-TO-CHAR  
>map>utm-grid-mixin.lisp  
CHAR-TO-COORD  
>map>utm-grid-mixin.lisp  
FILL-ALPHABET-ARRAY  
>map>utm-grid-mixin.lisp  
Description: None

**2.4.2.12.2 FILL-ALPHABET-ARRAY**

## Definition 2

>map>utm-grid-mixin.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*ALPHABET-ARRAY\*  
>map>utm-grid-mixin.lisp  
Called by: None  
Description: None

**2.4.2.12.3 NIL**

## Definition 3

>map>utm-grid-mixin.lisp  
Type: FILL-ALPHABET-ARRAY  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.12.4 CHAR-TO-COORD**

## Definition 4

>map>utm-grid-mixin.lisp  
Type: Function  
Arguments: (CHAR)  
Outputs:  
Calls: \*ALPHABET-ARRAY\*  
>map>utm-grid-mixin.lisp

Called by: (METHOD UTM-TO-WORLD UTM-GRID-MIXIN)  
>map>utm-grid-mixin.lisp  
(METHOD WORLD-TO-UTM UTM-GRID-MIXIN)  
>map>utm-grid-mixin.lisp  
Description: None

#### 2.4.2.12.5 COORD-TO-CHAR

Definition 5

>map>utm-grid-mixin.lisp  
Type: Function  
Arguments: (COORD)  
Outputs:  
Calls: \*ALPHABET-ARRAY\*  
>map>utm-grid-mixin.lisp  
Called by: (METHOD WORLD-TO-UTM UTM-GRID-MIXIN)  
>map>utm-grid-mixin.lisp  
Description: None

#### 2.4.2.12.6 UTM-OFFSET

Definition 6

>map>utm-grid-mixin.lisp  
Type: Function  
Arguments: (DIGIT-STRING)  
Outputs:  
Calls: None  
Called by: (METHOD UTM-TO-WORLD UTM-GRID-MIXIN)  
>map>utm-grid-mixin.lisp  
Description: None

#### 2.4.2.12.7 'UTM-GRID-MIXIN

Definition 7

>map>utm-grid-mixin.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.12.8 UTM-GRID-MIXIN

Definition 8

>map>utm-grid-mixin.lisp  
Type: Flavor  
Arguments: ()  
Outputs:

Calls: None  
Called by: None  
Description: None

#### 2.4.2.12.9 (METHOD UPDATE UTM-GRID-MIXIN AFTER)

Definition 9

>map>utm-grid-mixin.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: \*QUAD-TREE\*  
>map>terrain-vars.lisp  
Called by: None  
Description: None

#### 2.4.2.12.10 'SET-ORIGIN-UTM-COORDINATES

Definition 10

>map>utm-grid-mixin.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.12.11 (METHOD SET-ORIGIN-UTM-COORDINATES UTM-GRID-MIXIN)

Definition 11

>map>utm-grid-mixin.lisp  
Type: Method  
Arguments: (UTM-STRING)  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.12.12 'WORLD-TO-UTM

Definition 12

>map>utm-grid-mixin.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.12.13 (METHOD WORLD-TO-UTM UTM-GRID-MIXIN)**

Definition 13

>map>utm-grid-mixin.lisp  
Type: Method  
Arguments: (WORLD-X WORLD-Y &KEY (SIZE 3))  
Outputs:  
Calls: CHAR-TO-COORD  
>map>utm-grid-mixin.lisp  
COORD-TO-CHAR  
>map>utm-grid-mixin.lisp  
Called by: None  
Description: None

**2.4.2.12.14 'UTM-TO-WORLD**

Definition 14

>map>utm-grid-mixin.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.12.15 (METHOD UTM-TO-WORLD UTM-GRID-MIXIN)**

Definition 15

>map>utm-grid-mixin.lisp  
Type: Method  
Arguments: (UTM-STRING)  
Outputs:  
Calls: CHAR-TO-COORD  
>map>utm-grid-mixin.lisp  
UTM-OFFSET  
>map>utm-grid-mixin.lisp  
Called by: None  
Description: None

**2.4.2.12.16 UTM-GRID-MIXIN**

Definition 16

>map>utm-grid-mixin.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: MAP-WINDOW  
>saf>ui>frame-utils.lisp  
Description: None

**2.4.2.13 CSU map>vectors.lisp**

This unit contains general purpose two-dimensional vector routines, as well as the routine that draws the bridge symbol. These include functions for normalizing, rotating, adding, subtracting, and scaling two-dimensional vectors. Another function finds the direction angle of a vector using the arctangent.

**2.4.2.13.1 'VEC-NORMALIZE**

Definition 1

```

>map>vectors.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

**2.4.2.13.2 VEC-NORMALIZE**

Definition 2

```

>map>vectors.lisp
Type: Function
Arguments: (VECTOR)
Outputs:
Calls: None
Called by: DRAW-2-SCALLOPED-LINES
          >map>control.lisp
          DRAW-1-SCALLOPED-LINE
          >map>control.lisp
          DRAW-UNIT-SYMBOL
          >map>control.lisp
          (METHOD DRAW LINE-CONTROL-MEASURE)
          >map>control.lisp
          POINT-LINE-INTERSECTION
          >map>intersection.lisp
          DRAW-BRIDGE-SYMBOL
          >map>vectors.lisp
          FIND-INTER-POINT
          >map>vectors.lisp
          VEC-ANGLE
          >map>vectors.lisp
          OFFSET-POINT
          >saf>cm>water-avoidance.lisp
          NORMALIZE-AND-ROTATE
          >saf>cm>water-avoidance.lisp
          EXTEND-SEGMENT
          >saf>cm>water-avoidance.lisp

```



EXTEND-BRIDGE  
 >saf>cm>water-avoidance.lisp  
 EXTEND-INTERSECTION  
 >saf>cm>water-avoidance.lisp  
 ALL-WIDE-SEGMENTS-THRU-WATER  
 >saf>cm>water-check.lisp  
 ANY-WIDE-SEGMENT-THRU-WATER  
 >saf>cm>water-check.lisp

Description: None

#### 2.4.2.13.3 'VEC-ROTATE

Definition 3

>map>vectors.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.13.4 VEC-ROTATE

Definition 4

>map>vectors.lisp  
 Type: Function  
 Arguments: (VECTOR ANGLE)  
 Outputs:  
 Calls: None  
 Called by: DRAW-ARROW  
 >map>control.lisp  
 DRAW-2-SCALLOPED-LINES  
 >map>control.lisp  
 DRAW-UNIT-SYMBOL  
 >map>control.lisp  
 (METHOD DRAW BATTLE-POSITION AFTER)  
 >map>control.lisp  
 POINT-LINE-INTERSECTION  
 >map>intersection.lisp  
 DRAW-BRIDGE-SYMBOL  
 >map>vectors.lisp  
 OFFSET-POINT  
 >saf>cm>water-avoidance.lisp  
 FIND-FIRST-VECTOR  
 >saf>cm>water-avoidance.lisp  
 NORMALIZE-AND-ROTATE  
 >saf>cm>water-avoidance.lisp  
 EXTEND-SEGMENT  
 >saf>cm>water-avoidance.lisp

EXTEND-INTERSECTION  
 >saf>cm>water-avoidance.lisp  
 ALL-WIDE-SEGMENTS-THRU-WATER  
 >saf>cm>water-check.lisp  
 ANY-WIDE-SEGMENT-THRU-WATER  
 >saf>cm>water-check.lisp  
 (METHOD INTERVENE SIMNET-AGENT FOLLOW-VEHICLE)  
 >saf>objects>intervention.lisp

Description: None

#### 2.4.2.13.5 'VEC-ADD

Definition 5

>map>vectors.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None  
 2.4.2.13.6 VEC-ADD  
 Definition 6

>map>vectors.lisp  
 Type: Function  
 Arguments: (VECTOR-A VECTOR-B)  
 Outputs:  
 Calls: None  
 Called by: DRAW-ARROW  
 >map>control.lisp  
 DRAW-2-SCALLOPED-LINES  
 >map>control.lisp  
 DRAW-1-SCALLOPED-LINE  
 >map>control.lisp  
 DRAW-UNIT-SYMBOL  
 >map>control.lisp  
 (METHOD DRAW LINE-CONTROL-MEASURE)  
 >map>control.lisp  
 (METHOD DRAW BATTLE-POSITION AFTER)  
 >map>control.lisp  
 POINT-LINE-INTERSECTION  
 >map>intersection.lisp  
 DRAW-BRIDGE-SYMBOL  
 >map>vectors.lisp  
 FIND-INTER-POINT  
 >map>vectors.lisp  
 OFFSET-POINT  
 >saf>cm>water-avoidance.lisp  
 OFFSET-POINTS  
 >saf>cm>water-avoidance.lisp

EXTEND-SEGMENT  
 >saf>cm>water-avoidance.lisp  
 EXTEND-BRIDGE  
 >saf>cm>water-avoidance.lisp  
 EXTEND-INTERSECTION  
 >saf>cm>water-avoidance.lisp  
 ALL-WIDE-SEGMENTS-THRU-WATER  
 >saf>cm>water-check.lisp  
 ANY-WIDE-SEGMENT-THRU-WATER  
 >saf>cm>water-check.lisp

Description: None

#### 2.4.2.13.7 'VEC-SUB

Definition 7

>map>vectors.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.13.8 VEC-SUB

Definition 8

>map>vectors.lisp  
 Type: Function  
 Arguments: (VECTOR-A VECTOR-B)  
 Outputs:  
 Calls: None  
 Called by: DRAW-ARROW  
 >map>control.lisp  
 FIND-CENTER-POINT  
 >map>control.lisp  
 DRAW-2-SCALLOPED-LINES  
 >map>control.lisp  
 DRAW-1-SCALLOPED-LINE  
 >map>control.lisp  
 DRAW-UNIT-SYMBOL  
 >map>control.lisp  
 (METHOD DRAW LINE-CONTROL-MEASURE)  
 >map>control.lisp  
 (METHOD DRAW BATTLE-POSITION AFTER)  
 >map>control.lisp  
 POINT-LINE-INTERSECTION  
 >map>intersection.lisp  
 DRAW-BRIDGE-SYMBOL  
 >map>vectors.lisp  
 FIND-INTER-POINT  
 >map>vectors.lisp  
 (METHOD CM-INTERSECTION CM-POINT)  
 >saf>cm>point.lisp  
 FIND-DIRECTION-AT-CROSSING  
 >saf>cm>water-avoidance.lisp

OFFSET-POINT  
 >saf>cm>water-avoidance.lisp  
 OFFSET-POINTS  
 >saf>cm>water-avoidance.lisp  
 FIND-SEGMENT-CROSS-POINTS  
 >saf>cm>water-avoidance.lisp  
 NORMALIZE-AND-ROTATE  
 >saf>cm>water-avoidance.lisp  
 EXTEND-SEGMENT  
 >saf>cm>water-avoidance.lisp  
 EXTEND-BRIDGE  
 >saf>cm>water-avoidance.lisp  
 EXTEND-INTERSECTION  
 >saf>cm>water-avoidance.lisp  
 ALL-WIDE-SEGMENTS-THRU-WATER  
 >saf>cm>water-check.lisp  
 ANY-WIDE-SEGMENT-THRU-WATER  
 >saf>cm>water-check.lisp  
 (METHOD INTERVENE SIMNET-AGENT FOLLOW-VEHICLE)  
 >saf>objects>intervention.lisp

Description: None

#### 2.4.2.13.9 'VEC-SCALE

Definition 9

>map>vectors.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.13.10 VEC-SCALE

Definition 10

>map>vectors.lisp  
 Type: Function  
 Arguments: (VECTOR SCALE)  
 Outputs:  
 Calls: None  
 Called by: DRAW-ARROW  
 >map>control.lisp  
 FIND-CENTER-POINT  
 >map>control.lisp  
 DRAW-2-SCALLOPED-LINES  
 >map>control.lisp  
 DRAW-1-SCALLOPED-LINE  
 >map>control.lisp  
 DRAW-UNIT-SYMBOL  
 >map>control.lisp  
 (METHOD DRAW LINE-CONTROL-MEASURE)

```

>map>control.lisp
DRAW-BRIDGE-SYMBOL
>map>vectors.lisp
FIND-INTER-POINT
>map>vectors.lisp
OFFSET-POINT
>saf>cm>water-avoidance.lisp
FIND-SEGMENT-CROSS-POINTS
>saf>cm>water-avoidance.lisp
EXTEND-SEGMENT
>saf>cm>water-avoidance.lisp
EXTEND-BRIDGE
>saf>cm>water-avoidance.lisp
EXTEND-INTERSECTION
>saf>cm>water-avoidance.lisp
ALL-WIDE-SEGMENTS-THRU-WATER
>saf>cm>water-check.lisp
ANY-WIDE-SEGMENT-THRU-WATER
>saf>cm>water-check.lisp

```

Description: None

#### 2.4.2.13.11 'VEC-ANGLE

Definition 11

```

>map>vectors.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

#### 2.4.2.13.12 VEC-ANGLE

Definition 12

```

>map>vectors.lisp
Type: Function
Arguments: (VECTOR &OPTIONAL (NORMALIZE NIL))
Outputs:
Calls: VEC-NORMALIZE
>map>vectors.lisp
Called by: OFFSET-POINT
>saf>cm>water-avoidance.lisp
Description: None

```

#### 2.4.2.13.13 'FIND-INTER-POINT

Definition 13

```

>map>vectors.lisp
Type: EXPORT
Arguments: ()
Outputs:

```

Calls: None  
Called by: None  
Description: None

#### 2.4.2.13.14 FIND-INTER-POINT

Definition 14

>map>vectors.lisp  
Type: Function  
Arguments: (LENGTH X1 Y1 X2 Y2)  
Outputs:  
Calls: VEC-NORMALIZE  
>map>vectors.lisp  
VEC-ADD  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
VEC-SCALE  
>map>vectors.lisp  
Called by: GET-BRIDGE-POINTS  
>saf>cm>road-routes.lisp  
Description: None

#### 2.4.2.13.15 DRAW-BRIDGE-SYMBOL

Definition 15

>map>vectors.lisp  
Type: Function  
Arguments: (POINTS ALU STREAM)  
Outputs:  
Calls: PIE  
>map>utilities.lisp  
WITH-INTEGGER-CONVERSION-MODE  
>map>utilities.lisp  
WITH-MAP-GRAPHICS  
>map>utilities.lisp  
WITH-FAST-MAP-GRAPHICS  
>map>utilities.lisp  
VEC-NORMALIZE  
>map>vectors.lisp  
VEC-ROTATE  
>map>vectors.lisp  
VEC-ADD  
>map>vectors.lisp  
VEC-SUB  
>map>vectors.lisp  
VEC-SCALE  
>map>vectors.lisp  
Called by: DRAW-BRIDGES  
>map>draw-terrain.lisp  
Description: None

**2.4.2.14 CSU map>zoom-levels.lisp**

This unit defines the zoom level structure, which is used to store information about what is drawn at each zoom level. This information is stored with each quadtree structure. Routines are provided to access this zoom information. The zoom-level structure, defined by a defstruct, contains a number of slots that determine how the PVD is to be drawn at each zoom level. In general, as the level of magnification goes up, choices are made that increase the realism of the picture. At low magnification, simpler drawing methods are used for speed.

The first slot, meters-per-pixel, defines the actual numerical scale. The scale-string is a descriptive string used on the menu and legend, such as "1:50,000". The major and minor contour-line-intervals determine the altitude increments at which contour lines are drawn. The contour-point-interval determines how finely contour lines are drawn: 1 means draw every point, 2 means draw every other point, etc. The boolean variables draw-treelines and draw-treelines-as-spline determine if treelines are drawn, and whether splines are used instead of straight line segments.

Other boolean variables determine whether roads and railroads are drawn as narrow lines or with their width shown, and whether the zoom-level is included in the user's menu. Finally, anchor coordinates locate the center of the screen, and legend size and length determine the size of the legend box.

**2.4.2.14.1 ZOOM-LEVEL**

Definition 1

```

>map>zoom-levels.lisp
Type: DEFSTRUCT
Arguments: ()
Outputs:
Calls: None
Called by: MAKE-FT-KNOX-ZOOM-LEVELS
           >map>zoom-levels.lisp
           ZOOM-LEVEL-P
           >map>zoom-levels.lisp
           COPY-ZOOM-LEVEL
           >map>zoom-levels.lisp
           MAKE-ZOOM-LEVEL
           >map>zoom-levels.lisp
Description: None

```

**2.4.2.14.2 '\*ZOOM-LEVELS\***

Definition 2

```

>map>zoom-levels.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

**2.4.2.14.3 \*ZOOM-LEVELS\***

Definition 3

```

>map>zoom-levels.lisp
Type: Variable
Arguments:  ()
Outputs:
Calls: None
Called by:  DRAW-MAP
           >map>draw-terrain.lisp
           (METHOD RESCALE-FROM-MENU SCALABLE-WINDOW)
           >map>scalable-window.lisp
           (METHOD UPDATE SCALABLE-WINDOW)
           >map>scalable-window.lisp
           NEXT-ZOOM-IN
           >map>zoom-levels.lisp
           NEXT-ZOOM-OUT
           >map>zoom-levels.lisp
           (METHOD ADJUST-VIEWPORT SCENARIO)
           >saf>sys>new-storage.lisp
           RESCALE-PVD-FROM-MENU
           >saf>ui>commands.lisp
           (METHOD TOP-LEVEL SAF)
           >saf>ui>frame.lisp
           SET-UP-PVD-SCALE
           >saf>ui>frame.lisp
           DRAW-ANOTHER-TERRAIN-QUAD
           >saf>sys>update-process.lisp
           DRAW-MAP
           >saf>sys>update-process.lisp
           PROCESS-USER-COMMAND
           >saf>sys>update-process.lisp
Description: None

```

**2.4.2.14.4 \*CURRENT-ZOOM-LEVEL\***

Definition 4

```

>map>zoom-levels.lisp
Type: EXPORT
Arguments:  ()
Outputs:
Calls: None
Called by:  None
Description: None

```

**2.4.2.14.5 \*CURRENT-ZOOM-LEVEL\***

Definition 5

```

>map>zoom-levels.lisp
Type: Variable
Arguments:  ()
Outputs:

```



Calls: None

Called by: (METHOD ZOOM-OUT SCALABLE-WINDOW)

```
>map>scalable-window.lisp
(METHOD ZOOM-IN SCALABLE-WINDOW)
>map>scalable-window.lisp
DRAW-ALL-CONTOURS
>map>draw-terrain.lisp
DRAW-RAILS
>map>draw-terrain.lisp
DRAW-WATER
>map>draw-terrain.lisp
MOVE-DOWN-CONTOUR-LIST
>map>draw-terrain.lisp
DRAW-CONTOURS
>map>draw-terrain.lisp
DRAW-TREES
>map>draw-terrain.lisp
DRAW-ROADS
>map>draw-terrain.lisp
DRAW-TERRAIN
>map>draw-terrain.lisp
DRAW-MAP
>map>draw-terrain.lisp
(METHOD DRAW-LEGEND LEGEND-WINDOW)
>map>legend.lisp
(METHOD ZOOM-OUT-AROUND-CENTER SCALABLE-WINDOW)
>map>scalable-window.lisp
(METHOD ZOOM-IN-AROUND-CENTER SCALABLE-WINDOW)
>map>scalable-window.lisp
(METHOD ZOOM-TO SCALABLE-WINDOW)
>map>scalable-window.lisp
(METHOD RESCALE-FROM-MENU SCALABLE-WINDOW)
>map>scalable-window.lisp
(METHOD UPDATE SCALABLE-WINDOW)
>map>scalable-window.lisp
MAKE-HUNTERLGT-ZOOM-LEVELS
>map>zoom-levels.lisp
NEXT-ZOOM-IN
>map>zoom-levels.lisp
NEXT-ZOOM-OUT
>map>zoom-levels.lisp
LEGEND-LENGTH
>map>zoom-levels.lisp
LEGEND-SIZE
>map>zoom-levels.lisp
CURRENT-ANCHOR-Y
>map>zoom-levels.lisp
CURRENT-ANCHOR-X
>map>zoom-levels.lisp
CURRENT-SCALE
>map>zoom-levels.lisp
DRAW-RAILS-WITH-WIDTH
>map>zoom-levels.lisp
DRAW-WATER-WITH-WIDTH
```

```

>map>zoom-levels.lisp
DRAW-ROADS-WITH-WIDTH
>map>zoom-levels.lisp
DRAW-TREELINE-AS-SPLINE
>map>zoom-levels.lisp
DRAW-TREELINES
>map>zoom-levels.lisp
CONTOUR-POINT-INTERVAL
>map>zoom-levels.lisp
MINOR-CONTOUR-LINE-INTERVAL
>map>zoom-levels.lisp
MAJOR-CONTOUR-LINE-INTERVAL
>map>zoom-levels.lisp
SCALE-STRING
>map>zoom-levels.lisp
(METHOD COM-REFRESH-INTERNAL PVD)
No Source File Record
(METHOD COM-RESCALE-INTERNAL PVD)
No Source File Record
RESCALE-PVD-FROM-MENU
>saf>ui>commands.lisp
(METHOD TOP-LEVEL SAF)
>saf>ui>frame.lisp
SET-UP-PVD-SCALE
>saf>ui>frame.lisp
DRAW-ANOTHER-TERRAIN-QUAD
>saf>sys>update-process.lisp
DRAW-MAP
>saf>sys>update-process.lisp
PROCESS-NEW-MAP-OPTIONS
>saf>sys>update-process.lisp

```

Description: None

#### 2.4.2.14.6 'SCALE-STRING

Definition 6

```

>map>zoom-levels.lisp
Type: EXPORT
Arguments: ()
Outputs:
Calls: None
Called by: None
Description: None

```

#### 2.4.2.14.7 SCALE-STRING

Definition 7

```

>map>zoom-levels.lisp
Type: Subst
Arguments: ()
Outputs:

```

Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp

Called by: (METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
(METHOD RESCALE-FROM-MENU SCALABLE-WINDOW)  
>map>scalable-window.lisp  
RESCALE-PVD-FROM-MENU  
>saf>ui>commands.lisp

Description: None

#### 2.4.2.14.8 'MAJOR-CONTOUR-LINE-INTERVAL

Definition 8

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.2.14.9 MAJOR-CONTOUR-LINE-INTERVAL

Definition 9

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: DRAW-ALL-CONTOURS  
>map>draw-terrain.lisp  
DRAW-CONTOURS  
>map>draw-terrain.lisp  
(METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
Description: None

#### 2.4.2.14.10 'MINOR-CONTOUR-LINE-INTERVAL

Definition 10

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:

Calls: None  
Called by: None  
Description: None

**2.4.2.14.11 MINOR-CONTOUR-LINE-INTERVAL**  
Definition 11

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: DRAW-ALL-CONTOURS  
>map>draw-terrain.lisp  
DRAW-CONTOURS  
>map>draw-terrain.lisp  
(METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
Description: None

**2.4.2.14.12 'CONTOUR-POINT-INTERVAL**  
Definition 12

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.14.13 CONTOUR-POINT-INTERVAL**  
Definition 13

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: MOVE-DOWN-CONTOUR-LIST  
>map>draw-terrain.lisp  
Description: None

**2.4.2.14.14 'DRAW-TREELINES**

Definition 14

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.14.15 DRAW-TREELINES**

Definition 15

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: DRAW-TERRAIN  
>map>draw-terrain.lisp  
Description: None

**2.4.2.14.16 'DRAW-TREELINE-AS-SPLINE**

Definition 16

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.14.17 DRAW-TREELINE-AS-SPLINE**

Definition 17

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: DRAW-TREES  
>map>draw-terrain.lisp  
Description: None

**2.4.2.14.18 'DRAW-ROADS-WITH-WIDTH**  
Definition 18

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.14.19 DRAW-ROADS-WITH-WIDTH**  
Definition 19

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: DRAW-ROADS  
>map>draw-terrain.lisp  
Description: None

**2.4.2.14.20 'DRAW-WATER-WITH-WIDTH**  
Definition 20

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.14.21 DRAW-WATER-WITH-WIDTH**  
Definition 21

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: DRAW-WATER  
>map>draw-terrain.lisp  
Description: None

**2.4.2.14.22**            **'DRAW-RAILS-WITH-WIDTH**  
Definition 22

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.14.23**            **DRAW-RAILS-WITH-WIDTH**  
Definition 23

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: DRAW-RAILS  
>map>draw-terrain.lisp  
Description: None

**2.4.2.14.24**            **'CURRENT-SCALE**  
Definition 24

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.14.25**            **CURRENT-SCALE**  
Definition 25

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp

Called by: (METHOD ZOOM-OUT SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD ZOOM-IN SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 DRAW-MAP  
 >map>draw-terrain.lisp  
 (METHOD ZOOM-OUT-AROUND-CENTER SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD ZOOM-IN-AROUND-CENTER SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD ZOOM-TO SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD RESCALE-FROM-MENU SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD COM-REFRESH-INTERNAL PVD)  
 No Source File Record  
 (METHOD COM-RESCALE-INTERNAL PVD)  
 No Source File Record  
 SET-UP-PVD-SCALE  
 >saf>ui>frame.lisp  
 PROCESS-NEW-MAP-OPTIONS  
 >saf>sys>update-process.lisp  
 Description: None

#### 2.4.2.14.26 'CURRENT-ANCHOR-X Definition 26

>map>zoom-levels.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.2.14.27 CURRENT-ANCHOR-X Definition 27

>map>zoom-levels.lisp  
 Type: Subst  
 Arguments: ()  
 Outputs:  
 Calls: \*CURRENT-ZOOM-LEVEL\*  
 >map>zoom-levels.lisp  
 \*CURRENT-ZOOM-LEVEL\*  
 >map>zoom-levels.lisp  
 Called by: (METHOD ZOOM-OUT SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD ZOOM-IN SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD ZOOM-OUT-AROUND-CENTER SCALABLE-WINDOW)  
 >map>scalable-window.lisp



(METHOD ZOOM-IN-AROUND-CENTER SCALABLE-WINDOW)

>map>scalable-window.lisp

(METHOD ZOOM-TO SCALABLE-WINDOW)

>map>scalable-window.lisp

(METHOD RESCALE-FROM-MENU SCALABLE-WINDOW)

>map>scalable-window.lisp

(METHOD COM-RESCALE-INTERNAL PVD)

No Source File Record

SET-UP-PVD-SCALE

>saf>ui>frame.lisp

Description: None

#### 2.4.2.14.28 'CURRENT-ANCHOR-Y

Definition 28

>map>zoom-levels.lisp

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.2.14.29 CURRENT-ANCHOR-Y

Definition 29

>map>zoom-levels.lisp

Type: Subst

Arguments: ()

Outputs:

Calls: \*CURRENT-ZOOM-LEVEL\*

>map>zoom-levels.lisp

\*CURRENT-ZOOM-LEVEL\*

>map>zoom-levels.lisp

Called by: (METHOD ZOOM-OUT SCALABLE-WINDOW)

>map>scalable-window.lisp

(METHOD ZOOM-IN SCALABLE-WINDOW)

>map>scalable-window.lisp

(METHOD ZOOM-OUT-AROUND-CENTER SCALABLE-WINDOW)

>map>scalable-window.lisp

(METHOD ZOOM-IN-AROUND-CENTER SCALABLE-WINDOW)

>map>scalable-window.lisp

(METHOD ZOOM-TO SCALABLE-WINDOW)

>map>scalable-window.lisp

(METHOD RESCALE-FROM-MENU SCALABLE-WINDOW)

>map>scalable-window.lisp

(METHOD COM-RESCALE-INTERNAL PVD)

No Source File Record

SET-UP-PVD-SCALE

>saf>ui>frame.lisp

Description: None

**2.4.2.14.30 'LEGEND-SIZE**

Definition 30

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.14.31 LEGEND-SIZE**

Definition 31

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: (METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
Description: None

**2.4.2.14.32 'LEGEND-LENGTH**

Definition 32

>map>zoom-levels.lisp  
Type: EXPORT  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.2.14.33 LEGEND-LENGTH**

Definition 33

>map>zoom-levels.lisp  
Type: Subst  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: (METHOD DRAW-LEGEND LEGEND-WINDOW)  
>map>legend.lisp  
Description: None

**2.4.2.14.34**            **'NEXT-ZOOM-OUT**  
 Definition 34

          >map>zoom-levels.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None  
**2.4.2.14.35**            **NEXT-ZOOM-OUT**  
 Definition 35

          >map>zoom-levels.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: \*ZOOM-LEVELS\*  
       >map>zoom-levels.lisp  
       \*CURRENT-ZOOM-LEVEL\*  
       >map>zoom-levels.lisp  
       \*ZOOM-LEVELS\*  
       >map>zoom-levels.lisp  
       \*CURRENT-ZOOM-LEVEL\*  
       >map>zoom-levels.lisp  
 Called by: (METHOD ZOOM-OUT SCALABLE-WINDOW)  
           >map>scalable-window.lisp  
           (METHOD ZOOM-OUT-AROUND-CENTER SCALABLE-WINDOW)  
           >map>scalable-window.lisp  
 Description: None

**2.4.2.14.36**            **'NEXT-ZOOM-IN**  
 Definition 36

          >map>zoom-levels.lisp  
 Type: EXPORT  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

**2.4.2.14.37**            **NEXT-ZOOM-IN**  
 Definition 37

          >map>zoom-levels.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:

Calls: \*ZOOM-LEVELS\*  
       >map>zoom-levels.lisp  
       \*CURRENT-ZOOM-LEVEL\*  
       >map>zoom-levels.lisp  
       \*ZOOM-LEVELS\*  
       >map>zoom-levels.lisp  
       \*CURRENT-ZOOM-LEVEL\*  
       >map>zoom-levels.lisp  
 Called by: (METHOD ZOOM-IN SCALABLE-WINDOW)  
           >map>scalable-window.lisp  
           (METHOD ZOOM-IN-AROUND-CENTER SCALABLE-WINDOW)  
           >map>scalable-window.lisp  
 Description: None  
**2.4.2.14.38 MAKE-FT-KNOX-ZOOM-LEVELS**  
 Definition 38

      >map>zoom-levels.lisp  
 Type: Function  
 Arguments: ()  
 Outputs:  
 Calls: ZOOM-LEVEL  
       >map>zoom-levels.lisp  
 Called by: MAKE-HUNTERLGT-ZOOM-LEVELS  
           >map>zoom-levels.lisp  
 Description: None

**2.4.2.14.39 \*ZOOM-LEVELS\***  
 Definition 39

      >map>zoom-levels.lisp  
 Type: SETQ  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: DRAW-MAP  
           >map>draw-terrain.lisp  
           (METHOD RESCALE-FROM-MENU SCALABLE-WINDOW)  
           >map>scalable-window.lisp  
           (METHOD UPDATE SCALABLE-WINDOW)  
           >map>scalable-window.lisp  
           NEXT-ZOOM-IN  
           >map>zoom-levels.lisp  
           NEXT-ZOOM-OUT  
           >map>zoom-levels.lisp  
           (METHOD ADJUST-VIEWPORT SCENARIO)  
           >saf>sys>new-storage.lisp  
           RESCALE-PVD-FROM-MENU  
           >saf>ui>commands.lisp  
           (METHOD TOP-LEVEL SAF)  
           >saf>ui>frame.lisp

SET-UP-PVD-SCALE  
 >saf>ui>frame.lisp  
 DRAW-ANOTHER-TERRAIN-QUAD  
 >saf>sys>update-process.lisp  
 DRAW-MAP  
 >saf>sys>update-process.lisp  
 PROCESS-USER-COMMAND  
 >saf>sys>update-process.lisp

Description: None

#### 2.4.2.14.40 \*CURRENT-ZOOM-LEVEL\*

Definition 40

>map>zoom-levels.lisp  
 Type: SETQ  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: (METHOD ZOOM-OUT SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD ZOOM-IN SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 DRAW-ALL-CONTOURS  
 >map>draw-terrain.lisp  
 DRAW-RAILS  
 >map>draw-terrain.lisp  
 DRAW-WATER  
 >map>draw-terrain.lisp  
 MOVE-DOWN-CONTOUR-LIST  
 >map>draw-terrain.lisp  
 DRAW-CONTOURS  
 >map>draw-terrain.lisp  
 DRAW-TREES  
 >map>draw-terrain.lisp  
 DRAW-ROADS  
 >map>draw-terrain.lisp  
 DRAW-TERRAIN  
 >map>draw-terrain.lisp  
 DRAW-MAP  
 >map>draw-terrain.lisp  
 (METHOD DRAW-LEGEND LEGEND-WINDOW)  
 >map>legend.lisp  
 (METHOD ZOOM-OUT-AROUND-CENTER SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD ZOOM-IN-AROUND-CENTER SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD ZOOM-TO SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD RESCALE-FROM-MENU SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 (METHOD UPDATE SCALABLE-WINDOW)  
 >map>scalable-window.lisp  
 MAKE-HUNTERLGT-ZOOM-LEVELS

```
>map>zoom-levels.lisp
NEXT-ZOOM-IN
>map>zoom-levels.lisp
NEXT-ZOOM-OUT
>map>zoom-levels.lisp
LEGEND-LENGTH
>map>zoom-levels.lisp
LEGEND-SIZE
>map>zoom-levels.lisp
CURRENT-ANCHOR-Y
>map>zoom-levels.lisp
CURRENT-ANCHOR-X
>map>zoom-levels.lisp
CURRENT-SCALE
>map>zoom-levels.lisp
DRAW-RAILS-WITH-WIDTH
>map>zoom-levels.lisp
DRAW-WATER-WITH-WIDTH
>map>zoom-levels.lisp
DRAW-ROADS-WITH-WIDTH
>map>zoom-levels.lisp
DRAW-TREELINE-AS-SPLINE
>map>zoom-levels.lisp
DRAW-TREELINES
>map>zoom-levels.lisp
CONTOUR-POINT-INTERVAL
>map>zoom-levels.lisp
MINOR-CONTOUR-LINE-INTERVAL
>map>zoom-levels.lisp
MAJOR-CONTOUR-LINE-INTERVAL
>map>zoom-levels.lisp
SCALE-STRING
>map>zoom-levels.lisp
(METHOD COM-REFRESH-INTERNAL PVD)
No Source File Record
(METHOD COM-RESCALE-INTERNAL PVD)
No Source File Record
RESCALE-PVD-FROM-MENU
>saf>ui>commands.lisp
(METHOD TOP-LEVEL SAF)
>saf>ui>frame.lisp
SET-UP-PVD-SCALE
>saf>ui>frame.lisp
DRAW-ANOTHER-TERRAIN-QUAD
>saf>sys>update-process.lisp
DRAW-MAP
>saf>sys>update-process.lisp
PROCESS-NEW-MAP-OPTIONS
>saf>sys>update-process.lisp
Description:  None
```

**2.4.2.14.41 MAKE-HUNTERLGT-ZOOM-LEVELS**  
Definition 41

>map>zoom-levels.lisp  
Type: Function  
Arguments: ()  
Outputs:  
Calls: \*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
MAKE-FT-KNOX-ZOOM-LEVELS  
>map>zoom-levels.lisp  
\*CURRENT-ZOOM-LEVEL\*  
>map>zoom-levels.lisp  
Called by: None  
Description: None

**2.4.2.15 CSU map>draw-terrain.lisp**

This unit contains the routines for drawing all of the terrain features on the Symbolics color screen. At the highest map scale, the drawing routines just use the terrain feature arrays. At all other map scales, the terrain quadtree and line clipping routines are used to determine which features to draw. A global variable holds the types of terrain features to draw, which can be changed by the user, via menus. These features include roads, rivers, lakes and oceans, railroads, bridges, trees, buildings, contour-lines, the grid, and control-measures.

### 2.4.3 Vehicle and Effects Display CSC

This CSC contains the code to draw the vehicle icons and the fire effects on the map display. It contains the following CSUs:

```
color-window>color-alus.lisp csu
fonts>bluefor-icons.bfd csu
fonts>opfor-icons.bfd csu
simnet-objects>draw-vehicles.lisp csu
simnet-objects>new-draw-vehicles.lisp csu
simnet-objects>draw-effects csu
simnet-objects>draw-units csu
```

#### 2.4.3.1 CSU color-window>color-alus.lisp

This unit contains the routines that generate the color alus and colormap entries for the Symbolics color system to display vehicle icons and weapon effects.

The vehicles are displayed in the fourth and fifth bits of the 8 bit color system pixels. The weapon effects are displayed in the sixth and seventh bits, and the eighth bit is used for drawing the control measures. The first, second and third bits, numbered 0, 1, and 2, are used for terrain colors.

The SAF color display uses a bit-plane scheme that effectively allows certain kinds of features to mask others. This saves time when erasing vehicles and effects, because the background terrain features are still there, and simply reappear; they don't have to be redrawn. The scheme is based on an 8-bit representation of an integer index into the Symbolics color-map of 256 colors. The 8 bits are divided into 4 fields, as follows:

0	1	2		3	4		5	6		7
terrain				vehicles			effects			overlays

Bits 0, 1, and 2 are used to represent terrain colors, bits 3 and 4 are used to represent vehicles, bits 5 and 6 are used to represent effects, and bit 7 is used to represent overlays.

The code that sets up the color map uses these bit-fields to implement the following *masking rule*: Each bit-field represents, in effect, an image-plane. The planes are stacked on top of each other, with the overlay plane closest to the viewer. The bit values in each bit field specify the color that is present on that image-plane, at a particular pixel. Zero (all bits 0) represents a transparent "color". Thus the color the user actually sees at the pixel will be the color that is coded in the *closest non-zero* bit-field. Zero bit-fields in front of this one are transparent; bit-fields behind it are masked by its opaque color. If all the bit-fields are zero, then all four layers are transparent, and the user sees a fixed background color that effectively forms a final background image-plane behind the four original image-planes.

This masking effect is not provided by the Symbolics software, but is implemented by the SAF code, using the color map, an array of 256 colors. For each 8-bit pattern, a color-map entry with the corresponding integer index is created, using the masking rule to determine the color. This could be accomplished by scanning each bit-pattern to find the closest (highest numbered) nonzero bit-field, but instead, the colors for the layers are simply loaded into the color-map starting with the lowest field, and working toward the highest (closest) field, as follows.



First, the background color, which happens to be green, is loaded into the 0 entry in the color-map. Next, for each of the 7 nonzero combinations in the first bitfield (terrain), all color-map indices which match that combination in that field are assigned the corresponding terrain color. Then, for each of the 3 nonzero combinations in the second bitfield (vehicles), all color-map indices which match that combination in that field are assigned the corresponding vehicle color, reassigning the colors of some indices that had previously been given terrain colors. This process, which occurs in the function `make-alu-and-set-colormap` (in the CSU map `>color-map.lisp`) is continued for each bit-field. Because colors for higher-numbered fields (closer to the viewer) are loaded in *after* lower-numbered fields, the colormap obeys the masking rule. In fact, if a given nonzero 8-bit pattern has some number K of nonzero bit fields, its color will be set K times, once for each bit-field, and the final value will be the *last* one set, by the *closest* nonzero bit-field.

The code does this in the function `setup-color-alus`. Terrain colors have previously been loaded elsewhere, so the vehicles, effects, and overlay colors are loaded, in that order. Notice also that alus for the transparent color in each field are created, using the function `make-an-alu`; these are used to *erase* imagery from the corresponding image-plane. The masking rule insures that when the visible image plane is erased at a given pixel, the next nonzero image plane beneath it becomes visible, so features that were masked do not have to be redrawn.

The arguments to `make-an-alu-and-set-colormap` are (in order) the size of the bit-field, the starting position of the bit-field, the specific bit-pattern in that field, and the red, green and blue numbers (adding up to 1.0) for the color. The function `make-an-alu` is similar, but without the color numbers.

#### 2.4.3.1.1 `(*ERASE-VEHICLES-ALU* *DEFENSE-ALU* *OFFENSE-ALU* *TRIM-ALU* *ERASE-EFFECTS-ALU*`

Definition 1

```
*WHITE-EFFECTS-ALU* *YELLOW-EFFECTS-ALU* *BOMB-EFFECTS-ALU*
*ERASE-OVERLAY-ALU*
*OVERLAY-ALU*)
```

```
>saf>color-window>color-alus.lisp
```

Type: EXPORT

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.3.1.2 `SETUP-COLOR-ALUS`

Definition 2

```
>saf>color-window>color-alus.lisp
```

Type: Function

Arguments: ()

Outputs:

Calls: \*OVERLAY-ALU\*  
       >map>color-map.lisp  
       \*ERASE-OVERLAY-ALU\*  
       >map>color-map.lisp  
       MAKE-AN-ALU  
       >map>color-map.lisp  
       MAKE-ALU-AND-SET-COLOR-MAP  
       >map>color-map.lisp  
       \*OFFENSE-ALU\*  
       >saf>sys>vars.lisp  
       \*DEFENSE-ALU\*  
       >saf>sys>vars.lisp  
       \*ERASE-EFFECTS-ALU\*  
       >saf>sys>vars.lisp  
       \*ERASE-VEHICLES-ALU\*  
       >saf>sys>vars.lisp  
       \*BOMB-EFFECTS-ALU\*  
       >saf>sys>vars.lisp  
       \*TRIM-ALU\*  
       >saf>sys>vars.lisp  
       \*WHITE-EFFECTS-ALU\*  
       >saf>sys>vars.lisp  
       \*YELLOW-EFFECTS-ALU\*  
       >saf>sys>vars.lisp  
 Called by: (INITIALIZATION \*TERRAIN-INITIALIZATION-LIST\* Init Window)  
           No Source File Record  
           (METHOD TOP-LEVEL SAF)  
           >saf>ui>frame.lisp  
 Description: None

#### 2.4.3.1.3 Init Window

Definition 3

      >saf>color-window>color-alus.lisp  
 Type: ADD-INITIALIZATION  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

#### 2.4.3.2 CSU fonts>bluefor-icons.bfd

Font containing US unit and vehicle symbols. These symbols, and those in fonts>opfor-  
 icons.bfd and fonts>military-icons.bfd, are adapted from standard military symbols. Many  
 of these are described in Chapter 2 of the Department of the Army Field Manual *Operational  
 Terms and Symbols*, (Field Manual FM 101-5-1.)

To view a font on the Symbolics workstation, type

```
show font [font-name]
```

at the Lisp command line. For a list of available fonts, press the Help key after typing "show font".

#### 2.4.3.3 CSU fonts>opfor-icons.bfd

Font containing Soviet unit and vehicle symbols.

#### 2.4.3.4 CSU simnet-objects>draw-vehicles.lisp

This unit contains code for drawing vehicles on the PVD. It uses graphics primitives to create images for helicopters, fixed-wing aircraft, tank hulls and turrets, mechs (mechanized infantry vehicles), mortars and trucks. This code is used when the user selects "No" under the "Paint Vehicles as Military Icons" option on the Appearance submenu of the robo-cop-control menu. (See the discussion of robo-cop-control in CSU ui>parameter-menus.lisp, section 2.7.10) When the user selects "Yes", the default, vehicles are painted using the new vehicle drawing code in in simnet-objects>new-draw-vehicles.lisp.

The code in this unit draws with realistic position and orientation, unlike the coarser (but faster) font-based approach in new-draw-vehicles.lisp. The realism feature is useful for debugging code that controls vehicle motion, because it displays more precise position and turret rotation, making it easier to see how the vehicles are actually responding.

##### 2.4.3.4.1 ERASE-VEHICLE-ALU

Definition 1

```
>saf>simnet-objects>draw-vehicles.lisp
Type: Subst
Arguments: ()
Outputs:
Calls: *ERASE-VEHICLES-ALU*
       >saf>sys>vars.lisp
Called by: (METHOD DRAW-MISSILE-IMAGE MISSILE-IMAGE)
           >saf>simnet-objects>draw-vehicles.lisp
           (METHOD DRAW-TURRET-IMAGE RD-TURRET-IMAGE)
           >saf>simnet-objects>draw-vehicles.lisp
           (METHOD DRAW-TURRET-IMAGE SQ-TURRET-IMAGE)
           >saf>simnet-objects>draw-vehicles.lisp
           (METHOD DRAW-IMAGE FIGHTER-IMAGE)
           >saf>simnet-objects>draw-vehicles.lisp
           (METHOD DRAW-IMAGE HELO-IMAGE)
           >saf>simnet-objects>draw-vehicles.lisp
```

**DRAW-FILLED-BOX**

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

**DRAW-BOX**

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

**WITH-CORRECT-MAP-GRAPHICS**

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Description: None

**2.4.3.4.2 WITH-CORRECT-MAP-GRAPHICS**

## Definition 2

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Type: Macro

Arguments: ((STREAM ALU) &amp;BODY BODY)

Outputs:

Calls: WITH-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

**WITH-FAST-MAP-GRAPHICS**

&gt;map&gt;utilities.lisp

**ERASE-VEHICLE-ALU**

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

**WITH-CORRECT-MAP-GRAPHICS**

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Called by: (METHOD DRAW-MISSILE-IMAGE MISSILE-IMAGE)

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

(METHOD DRAW-TURRET-IMAGE RD-TURRET-IMAGE)

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

(METHOD DRAW-TURRET-IMAGE SQ-TURRET-IMAGE)

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

(METHOD DRAW-IMAGE FIGHTER-IMAGE)

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

(METHOD DRAW-IMAGE HELO-IMAGE)

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

**DRAW-FILLED-BOX**

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

**DRAW-BOX**

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

**WITH-CORRECT-MAP-GRAPHICS**

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Description: None

**2.4.3.4.3 \*MIN-IMAGE-SCALE\***

## Definition 3

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Type: DEFINE-APPEARANCE-OPTION

Arguments: ()

Outputs:

Calls: None

Called by: DRAW-ARTY  
 >saf>simnet-objects>draw-effects.lisp  
 AMMO-TYPE-RADIUS  
 >saf>simnet-objects>draw-effects.lisp  
 (METHOD UPDATE-SCALE GROUND-VEHICLE-IMAGE)  
 >saf>simnet-objects>draw-vehicles.lisp  
 (METHOD UPDATE-SCALE FIGHTER-IMAGE)  
 >saf>simnet-objects>draw-vehicles.lisp  
 (METHOD UPDATE-SCALE HELO-IMAGE)  
 >saf>simnet-objects>draw-vehicles.lisp

Description: None

#### 2.4.3.4.4 DRAW-BOX

Definition 4

>saf>simnet-objects>draw-vehicles.lisp  
 Type: Function  
 Arguments: (RASTER ALU X Y FRONT BACK WIDTH WX/HY WY/HY)  
 Outputs:  
 Calls: WITH-INTEGER-CONVERSION-MODE  
 >map>utilities.lisp  
 WITH-MAP-GRAPHICS  
 >map>utilities.lisp  
 WITH-FAST-MAP-GRAPHICS  
 >map>utilities.lisp  
 \*ERASE-VEHICLES-ALU\*  
 >saf>sys>vars.lisp  
 \*!  
 >saf>sys>macros.lisp  
 ERASE-VEHICLE-ALU  
 >saf>simnet-objects>draw-vehicles.lisp  
 WITH-CORRECT-MAP-GRAPHICS  
 >saf>simnet-objects>draw-vehicles.lisp

Called by: (METHOD DRAW-COMPARTMENT-IMAGE B-COMPARTMENT-IMAGE)  
 >saf>simnet-objects>draw-vehicles.lisp  
 (METHOD DRAW-COMPARTMENT-IMAGE A-COMPARTMENT-IMAGE)  
 >saf>simnet-objects>draw-vehicles.lisp  
 (METHOD DRAW-TURRET-IMAGE SQ-TURRET-IMAGE)  
 >saf>simnet-objects>draw-vehicles.lisp  
 (METHOD DRAW-HULL-IMAGE HULL-IMAGE)  
 >saf>simnet-objects>draw-vehicles.lisp

Description: None

#### 2.4.3.4.5 DRAW-FILLED-BOX

Definition 5

>saf>simnet-objects>draw-vehicles.lisp  
 Type: Function  
 Arguments: (RASTER ALU X Y FRONT BACK WIDTH WX/HY WY/HY)  
 Outputs:

**Calls:** WITH-INTEGER-CONVERSION-MODE

```
>map>utilities.lisp
WITH-MAP-GRAPHICS
>map>utilities.lisp
WITH-FAST-MAP-GRAPHICS
>map>utilities.lisp
*ERASE-VEHICLES-ALU*
>saf>sys>vars.lisp
*!
>saf>sys>macros.lisp
ERASE-VEHICLE-ALU
>saf>simnet-objects>draw-vehicles.lisp
WITH-CORRECT-MAP-GRAPHICS
>saf>simnet-objects>draw-vehicles.lisp
```

**Called by:** (METHOD DRAW-COMPARTMENT-IMAGE B-COMPARTMENT-IMAGE)

```
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-COMPARTMENT-IMAGE A-COMPARTMENT-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-TURRET-IMAGE SQ-TURRET-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW-HULL-IMAGE HULL-IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
```

**Description:** None

#### 2.4.3.4.6 '(DRAW-IMAGE ERASE-IMAGE)

Definition 6

```
>saf>simnet-objects>draw-vehicles.lisp
```

**Type:** EXPORT

**Arguments:** ()

**Outputs:**

**Calls:** None

**Called by:** None

**Description:** None

#### 2.4.3.4.7 DRAW-IMAGE

Definition 7

```
>saf>simnet-objects>draw-vehicles.lisp
```

**Type:** DEFGENERIC

**Arguments:** ()

**Outputs:**

**Calls:** None

**Called by:** DRAW-SANDBOX-OBJECT

```
>saf>sandbox>sandbox-object.lisp
(METHOD ERASE-IMAGE IMAGE)
>saf>simnet-objects>draw-vehicles.lisp
(METHOD DRAW VEHICLE)
>saf>objects>vehicle.lisp
```

**Description:** None

**2.4.3.4.8 ERASE-IMAGE**

Definition 8

>saf>simnet-objects>draw-vehicles.lisp  
Type: DEFGENERIC  
Arguments: ()  
Outputs:  
Calls: None  
Called by: ERASE-SANDBOX-OBJECT  
>saf>sandbox>sandbox-object.lisp  
(METHOD ERASE VEHICLE)  
>saf>objects>vehicle.lisp  
Description: None

**2.4.3.4.9 UPDATE-SCALE**

Definition 9

>saf>simnet-objects>draw-vehicles.lisp  
Type: DEFGENERIC  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INIT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
(METHOD DRAW-IMAGE IMAGE BEFORE)  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.10 IMAGE**

Definition 10

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.11 (METHOD UPDATE-SCALE IMAGE)**

Definition 11

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (NEW-M/PIXEL &KEY (FORCE NIL))  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.12 (METHOD ERASE-IMAGE IMAGE)**

## Definition 12

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER STATUS X Y Z BEARING BEARING2 ALU)  
Outputs:  
Calls: DRAW-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.13 (METHOD DRAW-IMAGE IMAGE BEFORE)**

## Definition 13

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER STATUS X Y Z BEARING BEARING2 ALU ALU2  
&OPTIONAL (DRAW P T))  
Outputs:  
Calls: UPDATE-SCALE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.14 (METHOD DRAW-IMAGE IMAGE)**

## Definition 14

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER STATUS X Y Z BEARING BEARING2 ALU ALU2  
&OPTIONAL (DRAW P T))  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.15 IMAGE**

## Definition 15

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: SMOKE-CLOUD-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
MISSILE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
FAADS-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp



B-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HOWITZER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
MORTAR-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
FUEL-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
A-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
COMMAND-POST-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
SUPPLY-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
AMMO-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
RD-TURRET-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
MECH-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
SQ-TURRET-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
TANK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
UNKNOWN-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
FIGHTER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
REMOTE-FIGHTER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
LOCAL-FIGHTER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HELO-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
REMOTE-HELO-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
LOCAL-HELO-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp

Description: None

#### 2.4.3.4.16 HELO-IMAGE

Definition 16

>saf>simnet-objects>draw-vehicles.lisp

Type: Flavor

Arguments: ()

Outputs:

Calls: IMAGE  
       >saf>simnet-objects>draw-vehicles.lisp  
 Called by: None  
 Description: None

#### 2.4.3.4.17 (METHOD UPDATE-SCALE HELO-IMAGE) Definition 17

      >saf>simnet-objects>draw-vehicles.lisp  
 Type: Method  
 Arguments: (NEW-M/PIXEL &KEY (FORCE NIL))  
 Outputs:  
 Calls: \*MIN-IMAGE-SCALE\*  
       >saf>simnet-objects>draw-vehicles.lisp  
 Called by: None  
 Description: None

#### 2.4.3.4.18 (METHOD DRAW-IMAGE HELO-IMAGE) Definition 18

      >saf>simnet-objects>draw-vehicles.lisp  
 Type: Method  
 Arguments: (RASTER STATUS X Y Z BEARING BEARING2 ALU ALU2  
 &OPTIONAL (DRAWP T))  
 Outputs:  
 Calls: WITH-INTEGER-CONVERSION-MODE  
       >map>utilities.lisp  
       WITH-MAP-GRAPHICS  
       >map>utilities.lisp  
       WITH-FAST-MAP-GRAPHICS  
       >map>utilities.lisp  
       VEH-IMMOBILE  
       >saf>sys>constants.lisp  
       VEH-DESTROYED  
       >saf>sys>constants.lisp  
       VEH-OUT-OF-GAS  
       >saf>sys>constants.lisp  
       VEH-OUT-OF-AMMO  
       >saf>sys>constants.lisp  
       \*ERASE-VEHICLES-ALU\*  
       >saf>sys>vars.lisp  
       \*!  
       >saf>sys>macros.lisp  
       IS-STATUS  
       >saf>simnet-objects>macros.lisp  
       ERASE-VEHICLE-ALU  
       >saf>simnet-objects>draw-vehicles.lisp  
       WITH-CORRECT-MAP-GRAPHICS  
       >saf>simnet-objects>draw-vehicles.lisp  
 Called by: None  
 Description: None

**2.4.3.4.19 LOCAL-HELO-IMAGE**

Definition 19

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HELO-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.20 REMOTE-HELO-IMAGE**

Definition 20

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HELO-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.21 HELO-IMAGE**

Definition 21

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: REMOTE-HELO-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
LOCAL-HELO-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.22 FIGHTER-IMAGE**

Definition 22

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:

Calls: IMAGE  
       >saf>simnet-objects>draw-vehicles.lisp  
 Called by: None  
 Description: None

#### 2.4.3.4.23 (METHOD UPDATE-SCALE FIGHTER-IMAGE)

Definition 23

      >saf>simnet-objects>draw-vehicles.lisp  
 Type: Method  
 Arguments: (NEW-M/PIXEL &KEY (FORCE NIL))  
 Outputs:  
 Calls: \*MIN-IMAGE-SCALE\*  
       >saf>simnet-objects>draw-vehicles.lisp  
 Called by: None  
 Description: None

#### 2.4.3.4.24 (METHOD DRAW-IMAGE FIGHTER-IMAGE)

Definition 24

      >saf>simnet-objects>draw-vehicles.lisp  
 Type: Method  
 Arguments: (RASTER STATUS X Y Z BEARING BEARING2 ALU ALU2  
 &OPTIONAL (DRAWPT))  
 Outputs:  
 Calls: WITH-INTEGER-CONVERSION-MODE  
       >map>utilities.lisp  
       WITH-MAP-GRAPHICS  
       >map>utilities.lisp  
       WITH-FAST-MAP-GRAPHICS  
       >map>utilities.lisp  
       VEH-DESTROYED  
       >saf>sys>constants.lisp  
       VEH-OUT-OF-GAS  
       >saf>sys>constants.lisp  
       VEH-OUT-OF-AMMO  
       >saf>sys>constants.lisp  
       \*ERASE-VEHICLES-ALU\*  
       >saf>sys>vars.lisp  
       \*!  
       >saf>sys>macros.lisp  
       IS-STATUS  
       >saf>simnet-objects>macros.lisp  
       ERASE-VEHICLE-ALU  
       >saf>simnet-objects>draw-vehicles.lisp  
       WITH-CORRECT-MAP-GRAPHICS  
       >saf>simnet-objects>draw-vehicles.lisp  
 Called by: None  
 Description: None

**2.4.3.4.25 LOCAL-FIGHTER-IMAGE**

Definition 25

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
FIGHTER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.26 REMOTE-FIGHTER-IMAGE**

Definition 26

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
FIGHTER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.27 FIGHTER-IMAGE**

Definition 27

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: REMOTE-FIGHTER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
LOCAL-FIGHTER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.28 GROUND-VEHICLE-IMAGE**

Definition 28

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp

Called by: None  
Description: None

#### 2.4.3.4.29 (METHOD UPDATE-HULL-SCALE GROUND-VEHICLE-IMAGE)

Definition 29

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.3.4.30 (METHOD UPDATE-TURRET-SCALE GROUND-VEHICLE-IMAGE)

Definition 30

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.3.4.31 (METHOD UPDATE-COMPARTMENT-SCALE GROUND-VEHICLE-IMAGE)

Definition 31

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

#### 2.4.3.4.32 (METHOD UPDATE-MISSILE-SCALE GROUND-VEHICLE-IMAGE)

Definition 32

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.33 (METHOD UPDATE-SCALE GROUND-VEHICLE-IMAGE)**

Definition 33

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (NEW-M/PIXEL &KEY (FORCE NIL))  
Outputs:  
Calls: \*MIN-IMAGE-SCALE\*  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.34 (METHOD DRAW-HULL-IMAGE GROUND-VEHICLE-IMAGE)**

Definition 34

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER STATUS X Y WX/HY WY/HY ALU)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.35 (METHOD DRAW-TURRET-IMAGE GROUND-VEHICLE-IMAGE)**

Definition 35

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER STATUS X Y WX/HY WY/HY WX/TY WY/TY ALU)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.36 (METHOD DRAW-COMPARTMENT-IMAGE GROUND-VEHICLE-IMAGE)**

Definition 36

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER STATUS X Y WX/HY WY/HY ALU)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.37 (METHOD DRAW-MISSILE-IMAGE GROUND-VEHICLE-IMAGE)**

Definition 37

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER X Y WX/HY WY/HY WX/TY WY/TY ALU)  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.38 (METHOD DRAW-IMAGE GROUND-VEHICLE-IMAGE)**

Definition 38

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER STATUS X Y Z BEARING BEARING2 ALU ALU2  
&OPTIONAL (DRAWP T))  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.39 GROUND-VEHICLE-IMAGE**

Definition 39

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: FAADS-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
UNKNOWN-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
COMMAND-POST-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HOWITZER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
MORTAR-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
SUPPLY-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
FUEL-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp



**AMMO-TRUCK-IMAGE**`>saf>simnet-objects>draw-vehicles.lisp`**MECH-IMAGE**`>saf>simnet-objects>draw-vehicles.lisp`**TANK-IMAGE**`>saf>simnet-objects>draw-vehicles.lisp`

Description: None

**2.4.3.4.40 HULL-IMAGE**

## Definition 40

`>saf>simnet-objects>draw-vehicles.lisp`

Type: Flavor

Arguments: ()

Outputs:

Calls: IMAGE

`>saf>simnet-objects>draw-vehicles.lisp`

Called by: None

Description: None

**2.4.3.4.41 (METHOD UPDATE-HULL-SCALE HULL-IMAGE)**

## Definition 41

`>saf>simnet-objects>draw-vehicles.lisp`

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

**2.4.3.4.42 (METHOD DRAW-HULL-IMAGE HULL-IMAGE)**

## Definition 42

`>saf>simnet-objects>draw-vehicles.lisp`

Type: Method

Arguments: (RASTER STATUS X Y WX/HY WY/HY ALU)

Outputs:

Calls: VEH-IMMOBILE

`>saf>sys>constants.lisp`

VEH-DESTROYED

`>saf>sys>constants.lisp`

IS-STATUS

`>saf>simnet-objects>macros.lisp`

DRAW-BOX

`>saf>simnet-objects>draw-vehicles.lisp`

DRAW-FILLED-BOX

`>saf>simnet-objects>draw-vehicles.lisp`

Called by: None

Description: None

**2.4.3.4.43 HULL-IMAGE**

Definition 43

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: FAADS-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
UNKNOWN-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
COMMAND-POST-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HOWITZER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
MORTAR-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
SUPPLY-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
FUEL-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
AMMO-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
MECH-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
TANK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.44 SQ-TURRET-IMAGE**

Definition 44

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.45 (METHOD UPDATE-TURRET-SCALE SQ-TURRET-IMAGE)**

Definition 45

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.46 (METHOD DRAW-TURRET-IMAGE SQ-TURRET-IMAGE)**

Definition 46

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Type: Method

Arguments: (RASTER STATUS X Y WX/HY WY/HY WX/TY WY/TY ALU)

Outputs:

Calls: WITH-INTEGGER-CONVERSION-MODE

&gt;map&gt;utilities.lisp

WITH-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

WITH-FAST-MAP-GRAPHICS

&gt;map&gt;utilities.lisp

VEH-CANTFIRE

&gt;saf&gt;sys&gt;constants.lisp

VEH-DESTROYED

&gt;saf&gt;sys&gt;constants.lisp

\*ERASE-VEHICLES-ALU\*

&gt;saf&gt;sys&gt;vars.lisp

\*!

&gt;saf&gt;sys&gt;macros.lisp

IS-STATUS

&gt;saf&gt;simnet-objects&gt;macros.lisp

ERASE-VEHICLE-ALU

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

WITH-CORRECT-MAP-GRAPHICS

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

DRAW-BOX

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

DRAW-FILLED-BOX

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Called by: None

Description: None

**2.4.3.4.47 SQ-TURRET-IMAGE**

Definition 47

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Type: COMPILE-FLAVOR-METHODS

Arguments: ()

Outputs:

Calls: None

Called by: TANK-IMAGE

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Description: None

**2.4.3.4.48 RD-TURRET-IMAGE**

Definition 48

&gt;saf&gt;simnet-objects&gt;draw-vehicles.lisp

Type: Flavor

Arguments: ()

Outputs:

Calls: IMAGE

>saf>simnet-objects>draw-vehicles.lisp

Called by: None

Description: None

#### 2.4.3.4.49 (METHOD UPDATE-TURRET-SCALE RD-TURRET-IMAGE)

Definition 49

>saf>simnet-objects>draw-vehicles.lisp

Type: Method

Arguments: ()

Outputs:

Calls: None

Called by: None

Description: None

#### 2.4.3.4.50 (METHOD DRAW-TURRET-IMAGE RD-TURRET-IMAGE)

Definition 50

>saf>simnet-objects>draw-vehicles.lisp

Type: Method

Arguments: (RASTER STATUS X Y WX/HY WY/HY WX/TY WY/TY ALU)

Outputs:

Calls: WITH-INTEGGER-CONVERSION-MODE

>map>utilities.lisp

WITH-MAP-GRAPHICS

>map>utilities.lisp

WITH-FAST-MAP-GRAPHICS

>map>utilities.lisp

VEH-CANTFIRE

>saf>sys>constants.lisp

VEH-DESTROYED

>saf>sys>constants.lisp

\*ERASE-VEHICLES-ALU\*

>saf>sys>vars.lisp

\*!

>saf>sys>macros.lisp

IS-STATUS

>saf>simnet-objects>macros.lisp

ERASE-VEHICLE-ALU

>saf>simnet-objects>draw-vehicles.lisp

WITH-CORRECT-MAP-GRAPHICS

>saf>simnet-objects>draw-vehicles.lisp

Called by: None

Description: None

**2.4.3.4.51 RD-TURRET-IMAGE**

Definition 51

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: FAADS-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
MECH-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.52 A-COMPARTMENT-IMAGE**

Definition 52

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.53 (METHOD UPDATE-COMPARTMENT-SCALE A-COMPARTMENT-IMAGE)**

Definition 53

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.54 (METHOD DRAW-COMPARTMENT-IMAGE A-COMPARTMENT-IMAGE)**

Definition 54

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER STATUS X Y WX/HY WY/HY ALU)  
Outputs:

Calls: VEH-DESTROYED  
>saf>sys>constants.lisp  
IS-STATUS  
>saf>simnet-objects>macros.lisp  
DRAW-BOX  
>saf>simnet-objects>draw-vehicles.lisp  
DRAW-FILLED-BOX  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

#### 2.4.3.4.55 A-COMPARTMENT-IMAGE

Definition 55

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: COMMAND-POST-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HOWITZER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
MORTAR-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
SUPPLY-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
FUEL-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
AMMO-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

#### 2.4.3.4.56 B-COMPARTMENT-IMAGE

Definition 56

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.57 (METHOD UPDATE-COMPARTMENT-SCALE B-COMPARTMENT-IMAGE)**

Definition 57

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: ()  
Outputs:  
Calls: None  
Called by: None  
Description: None

**2.4.3.4.58 (METHOD DRAW-COMPARTMENT-IMAGE B-COMPARTMENT-IMAGE)**

Definition 58

>saf>simnet-objects>draw-vehicles.lisp  
Type: Method  
Arguments: (RASTER STATUS X Y WX/HY WY/HY ALU)  
Outputs:  
Calls: VEH-DESTROYED  
>saf>sys>constants.lisp  
IS-STATUS  
>saf>simnet-objects>macros.lisp  
DRAW-BOX  
>saf>simnet-objects>draw-vehicles.lisp  
DRAW-FILLED-BOX  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.59 B-COMPARTMENT-IMAGE**

Definition 59

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: HOWITZER-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
MORTAR-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
FUEL-TRUCK-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.60 MISSILE-IMAGE**

Definition 60

>saf>simnet-objects>draw-vehicles.lisp  
 Type: Flavor  
 Arguments: ()  
 Outputs:  
 Calls: IMAGE  
 >saf>simnet-objects>draw-vehicles.lisp  
 Called by: None  
 Description: None

**2.4.3.4.61 (METHOD UPDATE-MISSILE-SCALE MISSILE-IMAGE)**

Definition 61

>saf>simnet-objects>draw-vehicles.lisp  
 Type: Method  
 Arguments: ()  
 Outputs:  
 Calls: None  
 Called by: None  
 Description: None

**2.4.3.4.62 (METHOD DRAW-MISSILE-IMAGE MISSILE-IMAGE)**

Definition 62

>saf>simnet-objects>draw-vehicles.lisp  
 Type: Method  
 Arguments: (RASTER HX HY WX/HY WY/HY WX/TY WY/TY ALU)  
 Outputs:  
 Calls: WITH-INTEGER-CONVERSION-MODE  
 >map>utilities.lisp  
 WITH-MAP-GRAPHICS  
 >map>utilities.lisp  
 WITH-FAST-MAP-GRAPHICS  
 >map>utilities.lisp  
 \*ERASE-VEHICLES-ALU\*  
 >saf>sys>vars.lisp  
 \*!  
 >saf>sys>macros.lisp  
 ERASE-VEHICLE-ALU  
 >saf>simnet-objects>draw-vehicles.lisp  
 WITH-CORRECT-MAP-GRAPHICS  
 >saf>simnet-objects>draw-vehicles.lisp  
 Called by: None  
 Description: None



**2.4.3.4.63 MISSILE-IMAGE**

Definition 63

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: FAADS-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.64 TANK-IMAGE**

Definition 64

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
SQ-TURRET-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.65 TANK-IMAGE**

Definition 65

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INIT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.66 MECH-IMAGE**

Definition 66

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:

Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
RD-TURRET-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

#### 2.4.3.4.67 MECH-IMAGE

Definition 67

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INIT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

#### 2.4.3.4.68 AMMO-TRUCK-IMAGE

Definition 68

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
A-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

#### 2.4.3.4.69 AMMO-TRUCK-IMAGE

Definition 69

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INIT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.70 FUEL-TRUCK-IMAGE**

Definition 70

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
A-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
B-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.71 FUEL-TRUCK-IMAGE**

Definition 71

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.72 SUPPLY-TRUCK-IMAGE**

Definition 72

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
A-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.73 SUPPLY-TRUCK-IMAGE**

Definition 73

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INIT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.74 MORTAR-IMAGE**

Definition 74

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
A-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
B-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

**2.4.3.4.75 MORTAR-IMAGE**

Definition 75

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INIT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

**2.4.3.4.76 HOWITZER-IMAGE**

Definition 76

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:

Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
A-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
B-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

#### 2.4.3.4.77 HOWITZER-IMAGE Definition 77

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

#### 2.4.3.4.78 COMMAND-POST-IMAGE Definition 78

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
A-COMPARTMENT-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

#### 2.4.3.4.79 COMMAND-POST-IMAGE Definition 79

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None

Called by: INT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

#### 2.4.3.4.80 UNKNOWN-VEHICLE-IMAGE

Definition 80

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
GROUND-VEHICLE-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
HULL-IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

#### 2.4.3.4.81 UNKNOWN-VEHICLE-IMAGE

Definition 81

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INT-IMAGES  
>saf>simnet-objects>draw-vehicles.lisp  
Description: None

#### 2.4.3.4.82 SMOKE-CLOUD-IMAGE

Definition 82

>saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
>saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

#### 2.4.3.4.83 SMOKE-CLOUD-IMAGE

Definition 83

>saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()

Outputs:  
Calls: None  
Called by: INIT-IMAGES  
          >saf>simnet-objects>draw-vehicles.lisp  
Description: None

#### 2.4.3.4.84 FAADS-IMAGE

##### Definition 84

          >saf>simnet-objects>draw-vehicles.lisp  
Type: Flavor  
Arguments: ()  
Outputs:  
Calls: IMAGE  
          >saf>simnet-objects>draw-vehicles.lisp  
          GROUND-VEHICLE-IMAGE  
          >saf>simnet-objects>draw-vehicles.lisp  
          HULL-IMAGE  
          >saf>simnet-objects>draw-vehicles.lisp  
          RD-TURRET-IMAGE  
          >saf>simnet-objects>draw-vehicles.lisp  
          MISSILE-IMAGE  
          >saf>simnet-objects>draw-vehicles.lisp  
Called by: None  
Description: None

#### 2.4.3.4.85 FAADS-IMAGE

##### Definition 85

          >saf>simnet-objects>draw-vehicles.lisp  
Type: COMPILE-FLAVOR-METHODS  
Arguments: ()  
Outputs:  
Calls: None  
Called by: INIT-IMAGES  
          >saf>simnet-objects>draw-vehicles.lisp  
Description: None

#### 2.4.3.4.86 \*IMAGE-ARRAY\*

##### Definition 86

          >saf>simnet-objects>draw-vehicles.lisp  
Type: Variable  
Arguments: ()  
Outputs:  
Calls: None  
Called by: IMAGE-FOR-VEHICLE  
          >saf>simnet-objects>draw-vehicles.lisp  
Description: None