

PROGRAMMER'S MANUAL

FOR THE

SCENARIST™

AUTOMATED SCENARIO GENERATION SYSTEM

PROJECT TITLE: RESEARCH IN ARTIFICIAL
INTELLIGENCE FOR NON-COMMUNICATIONS
ELECTRONIC WARFARE SYSTEMS

Volume I
Program Description

Contract No. DAAB07-89-C-P017

August 31, 1991

Prepared for:

US ARMY COMMUNICATIONS-ELECTRONICS COMMAND
Fort Monmouth, New Jersey

Prepared by:

VISTA RESEARCH CORPORATION
3826 Snead Drive
Sierra Vista, Arizona 85635
(602)378-2130 (602)790-0500

Note (added 31 January 2017): This document was originally produced using WordPerfect 4.2, which did not have a capability for constructing graphics or for imbedding graphics. The graphics (data flow diagrams) were prepared using another program (CADD), and are not included in this document.

Copyright (C) 1991 Vista Research Corporation

Contents

Foreword..... 3

1.0. Introduction..... 4

2.0. Summary of Software Documentation..... 4

3.0. Hierarchy Charts 5

4.0. Data Flow Diagrams 16

5.0. Input/Output Data Elements 18

6.0. Description of System Modules..... 18

Appendix A. Scenarist Test Deployment..... 22

Appendix B. Test Output Data Interface File 27

Foreword

This report was prepared by the staff of Vista Research Corporation, under Contract No. DAAB07-89-C-P017 to the US Army Communications-Electronics Command. Project staff included Dr. J. George Caldwell, Principal Investigator, Mr. William N. Goodhue, Ms. Sharon K. Hoting, Dr. William O. Rasmussen, Mr. Eric Weiss, and Mr. Fletcher Aleong. Government monitoring of the project was provided by Dr. Frank Elmer, Head of the Advanced Concepts Division of the Center for Electronic Warfare/Reconnaissance, Surveillance and Target Acquisition (EW/RSTA). The CECOM Project Manager was Dr. Frank Elmer.

1.0. Introduction

1.1. Purpose of Document

This document is the Software Programmer's Manual for the Scenarist Automated Scenario Generation System. The purpose of this document is to provide information needed by a programmer to understand the functioning of the software on the host PC computer. This information may be used to interpret, check out, troubleshoot, or modify the existing software.

This volume (Volume I, Program Description) presents a top-level program description of the Scenarist software. This top-level description includes a description of the major functional areas and a listing of the functions in each of those areas. It includes a list of the functions which each function calls and a list of the functions called by each function. It lists the data files used by the program. It includes data flow diagrams that show the data flows among the major Scenarist functions. It presents a brief summary description of each of the major functions.

The first version of this document (dated June 1, 1991) also contained (in Section 5.1) a description of the major program data structures. Since the publication of the June 1 version, the Final Report was produced. The Final Report included both summary and detailed descriptions of the program data structures. To avoid duplication, that material has been removed from Section 5.1. The Section number remains, to maintain a consistent document format. Duplication of the program description is avoided not only to save paper, but to avoid the time-consuming and unnecessary task of updating similar material in different documents.

Volume II (Program and Data File Listings) contains listings of the Scenarist programs, and of the data files used in the first two applications. Additional program description is presented in the Final Report and the Variable Glossary. The Final Report includes detailed descriptions of the major Scenarist functions and major program functions, files, and variables. The Variable Glossary presents definitions of all functions and variables of the Scenarist software.

2.0. Summary of Software Documentation

A list of the documentation for the Scenarist Automated Scenario Generation System is given below. In the early (design) stage of the project, a number of software description documents were produced in the format prescribed by DOD-STD-2167A (Defense System Software Development). Because of the large amount of effort required to maintain those documents, they were not updated after the software top-level design was completed. Since a "rapid prototyping" approach was used to the system development, many design developments and modifications occurred during the system development. For this reason, the 2167A documents became somewhat obsolete (mainly because they did not include any detailed-design information). All relevant material from those documents has been incorporated into either this Programmer's Manual, the Variable Glossary, or the Final Report.

1. Test Report for the Scenarist Automated Scenario Generation System, Vista Research Corporation, Tucson, Arizona, April 30, 1991.

2. User's Manual for the Scenarist Automated Scenario Generation System, Vista Research Corporation, Tucson, Arizona, August 31, 1991.

3. Programmer's Manual for the Scenarist Automated Scenario Generation System (including separately bound program and file listing), Vista Research Corporation, Tucson, Arizona, August 31, 1991.

4. Variable Glossary for the Scenarist Automated Scenario Generation System, Vista Research Corporation, Sierra Vista, Arizona, August 31, 1991.

5. Scenarist Automated Scenario Generation System: Final Report, Final Report for Project, "Research in Artificial Intelligence for Noncommunications Electronic Warfare Systems

3.0. Hierarchy Charts

The decomposition of the Scenarist Automated Scenario Generation System into functional areas provides a convenient mechanism for displaying and discussing information about the software. A tree-structured decomposition of the Scenarist showing functional areas and processes is presented in Section 3.1. of this manual.

Following sections will present information such as C function cross reference tables and data flow diagrams using this functional breakdown. This will provide the programmer with a consistent view of the structure of the Scenarist.

3.1. Summary of Functional Areas and Processes

Table 3.1. A tree-structured decomposition of the Scenarist Automated Scenario Generation System showing functional areas and processes.

```

*****
*   FUNCTIONAL AREAS           *   PROCESSES                       *
*****
* 0:  Scenarist Initialization *   Initialize screen windows, CLIPS setup *
*****
* 1:  Project File Selection   *   _project_selection() function           *
*****
* 2:  Set Initial Map Location *   _Change_Map_Location() function        *
*      Point                   *                                           *
*****
* 3:  Brief system description *   _About() function                       *
*****
* 4:  Processing Military      *   _Units() function                      *
*      Units                   *                                           *
*                               *   _Define_Unit(), _Copy_Unit(),          *
*                               *   _Delete_Unit(), _Reposition           *
*                               *   Unit_by_User(), _Reposition_         *
*                               *   Subunits_by_User(), _Reposition_     *
*                               *   Subunits_by_Rules(), _Reposition_FEBA(), *
*                               *   _Display_Unit(), _Output_Unit(),     *
*                               *   Return to previous Menu              *
*****
* 5:  Entry point for integrated *   _Rules() function                      *
*      CLIPS rules editor      *                                           *
*****
* 6:  Processing Maps          *   _Map() function                       *
*                               *                                           *
*                               *   _Draw_Terrain_Map(), _Draw_Elevation_ *
*                               *   Map(), _Draw_Road_Map(), _Add_Vector_Map_ *
*                               *   with_Labels(), _Add_Vector_Map_without_ *
*                               *   Labels(), _Draw_Vector_Map_without_ *
*                               *   Labels(), _Place_Unit_on_Map(),      *
*                               *   _Zoom_Map(), _Change_Map_Location(), *
*                               *   _Change_Map_Files(), _Print_Map(), *
*                               *   Return to Previous Menu              *
*****
* 7:  Entry point for large-   *   _Scenario_Generation function         *
*      scale scenario generation *                                           *
*****
* 8:  Display hardware         *   _Hardware_Setup() function            *
*      characteristics         *                                           *
*****
* 9:  Exit from the Scenarist  *   _Quit() function                      *
*****

```

3.2. Summary of C Functions by Source Code Files

The cross reference between Scenarist source code files and C functions, function prototypes, data structures, and data initializations is given in Table 3.2.

Table 3.2. The cross reference between Scenarist source code files and C functions, function prototypes, data structures, and data initializations.

```

*****
*   Scenarist Source                                     *
*   Code Files                                     Description                               *
*                                                                 *
*   s03adoca.doc   Documentation -- Description of Unit Data Structure *
*   s03binca.h    Principal Include File                               *
*   s03cincb.h    Header File for Map Drawing Programs -- Declarations *
*   s03dintb.h    Header File for Map Drawing Programs -- Initialization *
*   s03eclip.h    CLIPS Header File                                   *
*   s03fintf.h    Window, Menu, and Mouse Header File               *
*   s03gdemo.h    EGA, DEMO, and CLIPS Conditional Compilation Header File *
*   s03gmain.c    Main Program: main,usrfuncs,CLIPSval_0101,CLIPSval_0201 *
*   s03ipuu.c     Reposition Unit by User: _repositionunitbyuser,    *
*                 _repositionprogeny,_repositionfeba                 *
*   s03jpsu.c     Reposition Subunits by User:_repositionsubunitsbyuser *
*   s03kpsr.c     Reposition Subunits by Rule -- File 1 (General Rule *
*                 Repositioning Functions Plus Functions Specific    *
*                 to Problem 0101 -- Field Artillery and Air Defense *
*                 Radars):_repositionsubunitsbyrule,                 *
*                 _spiralsearch,_preprocessing0101,_action0101,     *
*                 _suitability0101,_terrain,                         *
*                 _elevation,_distancetofeba,_horizonangle,        *
*                 _clipssuitability0101                             *
*   s03kpsr2.c    Reposition Subunits by Rule -- File 2               *
*                 (Functions for Problem 0201 -- TRAILBLAZER):       *
*                 _preprocessing0201,_createaccessibilitymap,       *
*                 _accessibility,_lostotarget,_LOS                  *
*                 _road,_slopetorearcell,_lostootherunits,         *
*                 _lostoheadquarters,_dissttootherunits,           *
*                 _distancetofront,_inforwardarea,_action0201,     *
*                 _suitability0201,_clipssuitability0201           *
*   s03ldefu.c    Define Units Function: _defineunit                 *
*   s03mcopy.c    Copy, Delete, and Display Units Functions: _editunit, *
*                 _copyunit,                                         *
*                 _getunitbycode,_getsubunitbycode,_getunitbyidno, *
*                 _getunitbyindex,_getsubunitbyidno,_deleteunit,   *
*                 _displayunit,_setcoordsforzoommap,_outputunits   *
*   s03nmap.c     Map Drawing Functions: _drawmap,_label,_legend,   *
*                 _setcoordsforzoommapb                             *
*****

```

Table 3.2. (Continued) The cross reference between Scenarist source code files and C functions, function prototypes, data structures, and data initializations.

```

*****
*   Scenarist Source                                     *
*   Code Files                                     Description                               *
*                                                                 *
*   s03pdraw.c    Functions for Drawing Units on Map: _drawunit,    *
*                 _drawline,_transfstdtoreal                       *
*   s03qgftfl.c   Get Files Function: _getfilenames,_readmapheader, *
*                 _readmapdata,_resetmaplocpoint                   *
*   s03rsymb.c    Function for Drawing Symbols: _symbol              *
*   s03sio.c      Basic Input-Output Functions: _printscreen,_clearfoot, *
*                 LJ_Graphic,format,Grey_Scale,Print_Pause,PromptLine, *
*                 writString,writChar,put_out,status,pralel,stat,PrtInit,*
*                 printch,printst,PrtPutC,getMode,getxy,gotoxy,getPage *
*****

```

```

* s03twind.c      Windows Functions: setWindow,_windowa,_windowb,      *
*                _windowblack,_windowbenter,_windowbl,_windowblenter, *
*                _windowb2,_windowb3,_windowc,_windowcenter, *
*                _windowscreen,_windowscreenblue,_windowd,_windowe, *
*                _windoweenter *
* s03vmous.c      Mouse Functions: m_reset,m_show,m_hide,m_pos,m_moveto, *
*                m_pressed,m_released,m_xlimit,m_ylimit,m_text_cursor, *
*                m_motion,m_lightpen,m_move_ratio,m_conceal,m_speed, *
*                m_graphic_cursor,initialize_cursor,Set_Graphic_Cursor, *
*                _setup_button,_clickbutton,_click_accept,_click_cancel *
*                _call_button,_call_buttonb,_call_continueb, *
*                _call_continue,_Hardware_Setup,_setup_screen_windows *
*                _display_intro_screens, *
*                _display_screen_windows,_project_selection, *
*                _main_mouse_selection,_display_windowd,_About,_Units *
*                _Rules,_Map,_Scenario_Generation,_Quit *
* s03ygra2.c      Graphics Interface Functions -- Map- and Unit-Related: *
*                _map_menu_selection, *
*                _unit_menu_selection,_Draw_Terrain_Map, *
*                _Draw_Elevation_Map,_Draw_Vector_Map, *
*                _Place_Unit_on_Map,_Zoom_Map,_Change_Map_Location, *
*                _Change_Map_Files,_Reposition_Unit_by_User, *
*                _Reposition_Subunits_by_User, *
*                _Reposition_Subunits_by_Rules,_Print_Map,_Define_Unit, *
*                _Delete_Unit,_Reposition_FEBA,_Display_Unit, *
*                _Output_Units *
* s03xcomp.c      Map Compression Program: main, *
*                _readmapheadernongraphic,_writemapheader *
* s03zdemo.c      _run_demo_script,_call_continuec
*****

```

3.3. Summary of C Functions by Calling Function

The data shown in Table 3.3. can be used by the programmer to associate functional areas and functions with each of the primary Scenarist C functions. The data are organized alphabetically by function name (Column II). Each line entry for a function identifies a function that it calls.

Table 3.3. Alphabetical listing of each Scenarist C function (Column II) and the corresponding C function(s) (Column III) which it calls. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
I	II	III	IV
3	_About()	_call_continueb()	s03wgrap.c
3	_About()	_windowb()	s03wgrap.c
4	_accessibility()	none	s03kpsr2.c
4	_action0101()	none	s03kpsr.c
4	_action0201()	_elevation()	s03kpsr2.c
4	_Add_Vector_Map_with_Labels()	_drawmap()	s03ygra2.c
4	_Add_Vector_Map_without_Labels()	_drawmap()	s03ygra2.c
4	_Change_Map_Files()	_readmapheader()	s03ygra2.c
4	_Change_Map_Files()	_readmapheader()	s03ygra2.c
4	_Change_Map_Files()	_resetmaplocpoint()	s03ygra2.c
4	_Change_Map_Location()	_readmapdata()	s03ygra2.c
4	_Change_Map_Location()	_readmapheader()	s03ygra2.c
4	_Change_Map_Location()	_resetmaplocpoint()	s03ygra2.c
4	_clipssuitability0101()	_distancetofeba()	s03kpsr.c
4	_clipssuitability0101()	_horizonangle()	s03kpsr.c
4	_clipssuitability0101()	_terrain()	s03kpsr.c
4	_clipssuitability0101()	_transfstdtoreal()	s03kpsr.c
4	_clipssuitability0201()	none	s03kpsr2.c
4	_clipssuitability0101()	run()	s03kpsr.c
4	_Copy_Unit()	_copyunit()	s03ygra2.c

4	_Copy_Unit()	_editunit()	s03ygra2.c
4	_copyunit()	_getsubunitbycode()	s03mcopy.c
4	_copyunit()	_getunitbycode()	s03mcopy.c
4	_copyunit()	_transfstdtoreal()	s03mcopy.c
4	_copyunit()	_windowb1()	s03mcopy.c
4	_copyunit()	_windowb2()	s03mcopy.c
4	_copyunit()	_windowb3()	s03mcopy.c
4	_copyunit()	_windowc()	s03mcopy.c
4	_createaccessibilitymap()	_road()	s03kpsr2.c
4	_createaccessibilitymap()	_slopetorearcell()	s03kpsr2.c
4	_createaccessibilitymap()	_terrain()	s03kpsr2.c
4	_Define_Unit()	_defineunit()	s03ygra2.c
4	_Define_Unit()	_editunit()	s03ygra2.c
4	_defineunit()	_getsubunitbycode()	s03ldefu.c
4	_defineunit()	_getunitbycode()	s03ldefu.c
4	_defineunit()	_transfstdtoreal()	s03ldefu.c
4	_defineunit()	_windowb()	s03ldefu.c
4	_defineunit()	_windowbenter()	s03ldefu.c
4	_defineunit()	_windowc()	s03ldefu.c
4	_defineunit()	_windowe()	s03ldefu.c
4	_Delete_Unit()	_deleteunit()	s03ygra2.c

Table 3.3. (Continued) Alphabetical listing of each Scenarist C function (Column I) and the corresponding C function(s) (Column III) which it calls. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
<u>I</u>	<u>II</u>	<u>III</u>	<u>IV</u>
4	_deleteunit()	_getunitbycode()	s03mcopy.c
0	_display_intro_screens()	_call_continue()	s03wgrap.c
0	_display_intro_screens()	_windowscreenblue()	s03wgrap.c
0	_display_screen_windows()	_windowa()	s03wgrap.c
0	_display_screen_windows()	_windowb()	s03wgrap.c
0	_display_screen_windows()	_windowc()	s03wgrap.c
0	_display_screen_windows()	_windowd()	s03wgrap.c
0	_display_screen_windows()	_windowe()	s03wgrap.c
0	_display_screen_windows()	_windowscreen()	s03wgrap.c
4	_Display_Unit()	_displayunit()	s03ygra2.c
4	_Display_Unit()	_editunit()	s03ygra2.c
0	_display_windowd()	none	s03wgrap.c
4	_displayunit()	_drawmap()	s03mcopy.c
4	_displayunit()	_drawunit()	s03mcopy.c
4	_displayunit()	_getunitbycode()	s03mcopy.c
4	_displayunit()	_getunitbyidno()	s03mcopy.c
4	_displayunit()	_getunitbyindex()	s03mcopy.c
4	_displayunit()	_setcootdsforzoommap()	s03mcopy.c
4	_distancetofeba()	_transfstdtoreal()	s03kpsr.c
4	_disttootherunits()	_transfstdtoreal()	s03kpsr2.c
4	_Draw_Elevation_Map()	_drawmap()	s03ygra2.c
4	_Draw_Road_Map()	_drawmap()	s03ygra2.c
4	_Draw_Terrain_Map()	_drawmap()	s03ygra2.c
4	_Draw_Vector_Map_without_Labels()	_drawmap()	s03ygra2.c
4	_Draw_Vector_Map_with_Labels()	_drawmap()	s03ygra2.c
4	_drawline()	none	s03pdraw.c
6	_drawmap()	none	s03nmap.c
4	_drawunit()	_transfstdtoreal()	s03pdraw.c
4	_editunit()	none	s03mcopy.c
4	_elevation()	_transfstdtoreal()	s03kpsr.c
4	_mapcellsizestdcoords()	_transfstdtoreal()	s03kpsr.c
2	_getfilenames()	_readmapdata()	s03qgtfl.c
2	_getfilenames()	_readmapheader()	s03qgtfl.c
2	_getfilenames()	_resetmaplocpoint()	s03qgtfl.c
4	_getsubunitbycode()	none	s03mcopy.c
4	_getsubunitbyindex()	none	s03mcopy.c
4	_getunitbycode()	none	s03mcopy.c
4	_getunitbyidno()	none	s03mcopy.c

4	<code>_getunitbyindex()</code>	<code>none</code>	<code>s03mcopy.c</code>
8	<code>_Hardware_Setup()</code>	<code>_call_continueb()</code>	<code>s03vmous.c</code>
8	<code>_Hardware_Setup()</code>	<code>_windowb()</code>	<code>s03vmous.c</code>
4	<code>_horizonangle()</code>	<code>_elevation()</code>	<code>s03kpsr.c</code>
4	<code>_horizonangle()</code>	<code>_transfstdtoreal()</code>	<code>s03kpsr.c</code>
4	<code>_LOS()</code>	<code>_elevation()</code>	<code>s03kpsr2.c</code>

Table 3.3. (Continued) Alphabetical listing of each Scenarist C function (Column I) and the corresponding C function(s) (Column III) which it calls. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
<u>I</u>	<u>II</u>	<u>III</u>	<u>IV</u>
4	<code>_lostoheadquarters()</code>	<code>_getunitbycode()</code>	<code>s03kpsr2.c</code>
4	<code>_lostoheadquarters()</code>	<code>_LOS()</code>	<code>s03kpsr2.c</code>
4	<code>_losttootherunits()</code>	<code>_LOS()</code>	<code>s03kpsr2.c</code>
4	<code>_losttotarget()</code>	<code>_LOS()</code>	<code>s03kpsr2.c</code>
0	<code>_main_menu_selection()</code>	<code>_About()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_display_windowd()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_Hardware_Setup()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_Map()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_Quit()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_Rules()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_Scenario_Generation()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_Units()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_windowa()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_windowd()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_windowscreen()</code>	<code>s03wgrap.c</code>
6	<code>_Map()</code>	<code>_map_menu_selection()</code>	<code>s03wgrap.c</code>
6	<code>_Map()</code>	<code>_windowb()</code>	<code>s03wgrap.c</code>
6	<code>_map_menu_selection()</code>	<code>_Add_Vector_Map_without_Labels()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Add_Vector_Map_with_Labels()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Change_Map_Files()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Change_Map_Location()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_display_windowd()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Draw_Elevation_Map()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Draw_Road_Map()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Draw_Terrain_Map()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Draw_Vector_Map_without_Labels()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Place_Unit_on_Map()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Print_Map()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_windowa()</code>	<code>s03wgrap.c</code>
6	<code>_map_menu_selection()</code>	<code>_windowd()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_Zoom_Map()</code>	<code>s03ygra2.c</code>
4	<code>_Output_Units()</code>	<code>_outputunits()</code>	<code>s03ygra2.c</code>
4	<code>_outputunits()</code>	<code>_transfstdtoreal()</code>	<code>s03mcopy.c</code>
4	<code>_Place_Unit_on_Map()</code>	<code>_drawunit()</code>	<code>s03ygra2.c</code>
4	<code>_Place_Unit_on_Map()</code>	<code>_getunitbycode()</code>	<code>s03ygra2.c</code>
4	<code>_Place_Unit_on_Map()</code>	<code>_windowbenter()</code>	<code>s03ygra2.c</code>
4	<code>_Place_Unit_on_Map()</code>	<code>_windowcenter()</code>	<code>s03ygra2.c</code>
4	<code>_preprocessing0101()</code>	<code>none</code>	<code>s03kpsr.c</code>
4	<code>_preprocessing0201()</code>	<code>_createaccessibilitymap()</code>	<code>s03kpsr2.c</code>
4	<code>_Print_Map()</code>	<code>_printsreen()</code>	<code>s03ygra2.c</code>
1	<code>_project_selection()</code>	<code>_windowb()</code>	<code>s03wgrap.c</code>
1	<code>_project_selection()</code>	<code>_windowd()</code>	<code>s03wgrap.c</code>
9	<code>_Quit()</code>	<code>close_dribble()</code>	<code>s03wgrap.c</code>

Table 3.3. (Continued) Alphabetical listing of each Scenarist C function (Column I) and the corresponding C function(s) (Column III) which it calls. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
<u>I</u>	<u>II</u>	<u>III</u>	<u>IV</u>

6	_readmapdata()	_windowsscreenblue()	s03qgtf1.c
6	_readmapheader()	_call_continue()	s03qgtf1.c
6	_readmapheader()	_windowsscreenblue()	s03qgtf1.c
4	_Reposition_FEBA()	_repositionfeba()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_drawunit()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_getunitbycode()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_repositionprogeny()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_repositionsubunitsbyrule()	s03ygra2.c
4	_Reposition_Subunits_by_User()	_repositionsubunitsbyuser()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_windowbenter()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_windowc()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_windoweenter()	s03ygra2.c
4	_Reposition_Unit_by_User()	_repositionunitbyuser()	s03ygra2.c
4	_repositionfeba()	none	s03ipuu.c
4	_repositionprogeny()	_getsubunitbycode()	s03ipuu.c
4	_repositionprogeny()	_getunitbycode()	s03ipuu.c
4	_repositionprogeny()	_transfstdtoreal()	s03ipuu.c
4	_repositionsubunitsbyrule()	(*actionfunctionpointer())	s03kpsr.c
4	_repositionsubunitsbyrule()	(*preprocessingfnpointer())	s03kpsr.c
4	_repositionsubunitsbyrule()	(*suitabilityfunctionpointer())	s03kpsr.c
4	_repositionsubunitsbyuser()	_clearfoot()	s03jpsu.c
4	_repositionsubunitsbyrule()	_mapcellsizestdcoords()	s03kpsr.c
4	_repositionsubunitsbyuser()	_getunitbycode()	s03jpsu.c
4	_repositionsubunitsbyuser()	_repositionprogeny()	s03jpsu.c
4	_repositionsubunitsbyrule()	_spiralsearch()	s03kpsr.c
4	_repositionsubunitsbyrule()	_windowbenter()	s03kpsr.c
4	_repositionsubunitsbyuser()	_windowc()	s03jpsu.c
4	_repositionsubunitsbyrule()	_windoweenter()	s03kpsr.c
4	_repositionunitbyuser()	_getunitbycode()	s03ipuu.c
4	_repositionunitbyuser()	_repositionprogeny()	s03ipuu.c
4	_repositionunitbyuser()	_windowc()	s03ipuu.c
6	_resetmaplocpoint()	_call_continue()	s03qgtf1.c
4	_road()	_transfstdtoreal()	s03kpsr2.c
5	_Rules()	_call_continueb()	s03wgrap.c
5	_Rules()	_windowb()	s03wgrap.c
7	_Scenario_Generation()	_call_continueb()	s03wgrap.c
7	_Scenario_Generation()	_windowb()	s03wgrap.c
6	_setcoordsforzoommapb()	_getunitbycode()	s03nmap.c
6	_setcoordsforzoommapb()	_setcoordsforzoommap()	s03nmap.c
4	_setcoordsforzoommap()	none	s03mcopy.c
4	_slopetorearcell()	_elevation()	s03kpsr2.c
4	_slopetorearcell()	_transfstdtoreal()	s03kpsr2.c
4	_spiralsearch()	(*_suitabilityfunction())	s03kpsr.c

Table 3.3. (Continued) Alphabetical listing of each Scenarist C function (Column I) and the corresponding C function(s) (Column III) which it calls. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
<u>I</u>	<u>II</u>	<u>III</u>	<u>IV</u>
4	_suitability0101()	_transfstdtoreal()	s03kpsr.c
4	_suitability0201()	_accessibility()	s03kpsr2.c
4	_suitability0201()	_disttootherunits()	s03kpsr2.c
4	_suitability0201()	_lostoheadquarters()	s03kpsr2.c
4	_suitability0201()	_lostootherunits()	s03kpsr2.c
4	_suitability0201()	_lostotarget()	s03kpsr2.c
4	_suitability0201()	_terrain()	s03kpsr2.c
4	_suitability0201()	_transfstdtoreal()	s03kpsr2.c
4	_terrain()	_transfstdtoreal()	s03kpsr.c
4	_transfstdtoreal()	none	s03pdraw.c
4	_unit_menu_selection()	_Copy_Unit()	s03ygra2.c
4	_unit_menu_selection()	_Define_Unit()	s03ygra2.c
4	_unit_menu_selection()	_Delete_Unit()	s03ygra2.c
4	_unit_menu_selection()	_Display_Unit()	s03ygra2.c
4	_unit_menu_selection()	_display_windowd()	s03ygra2.c

4	_unit_menu_selection()	_Output_Units()	s03ygra2.c
4	_unit_menu_selection()	_Reposition_FEBA()	s03ygra2.c
4	_unit_menu_selection()	_Reposition_Subunits_by_User()	s03ygra2.c
4	_unit_menu_selection()	_Reposition_Subunits_by_Rules()	s03ygra2.c
4	_unit_menu_selection()	_Reposition_Unit_by_User()	s03ygra2.c
4	_unit_menu_selection()	_windowa()	s03ygra2.c
4	_unit_menu_selection()	_windowa()	s03ygra2.c
4	_unit_menu_selection()	_windowd()	s03ygra2.c
4	_unit_menu_selection()	_windowscreen()	s03ygra2.c
4	_Units()	_unit_menu_selection()	s03wgrap.c
0	_windowa()	none	s03twind.c
0	_windowb()	none	s03twind.c
0	_windowbl()	none	s03twind.c
0	_windowblenter()	none	s03twind.c
0	_windowb2()	none	s03twind.c
0	_windowb3()	none	s03twind.c
0	_windowbblack()	none	s03twind.c
0	_windowbenter()	none	s03twind.c
0	_windowc()	none	s03twind.c
0	_windowcenter()	none	s03twind.c
0	_windowd()	none	s03twind.c
0	_windowe()	none	s03twind.c
0	_windoweenter()	none	s03twind.c
0	_windowscreen()	none	s03twind.c
0	_windowscreenblue()	none	s03twind.c
4	_Zoom_Map()	_setcoordsforzoommapb()	s03ygra2.c
4	CLIPSval_0101(),CLIPSval_0202()	_windowbenter()	s03gmain.c
4	CLIPSval_0101(),CLIPSval_0202()	_windowc()	s03gmain.c

Table 3.3. (Continued) Alphabetical listing of each Scenarist C function (Column I) and the corresponding C function(s) (Column III) which it calls. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
<u>I</u>	<u>II</u>	<u>III</u>	<u>IV</u>
4	CLIPSval_0101(),CLIPSval_0201()	_windowe()	s03gmain.c
0	main()	_display_intro_screens()	s03gmain.c
0	main()	_display_screen_windows()	s03gmain.c
2	main()	_getfilenames()	s03gmain.c
0	main()	_main_menu_selection()	s03gmain.c
1	main()	_project_selection()	s03gmain.c
0	main()	_run_demo_script()	s03gmain.c
0	main()	_setup_screen_windows()	s03gmain.c
0	main()	close_dribble()	s03gmain.c
0	main()	init_clips()	s03gmain.c
0	main()	load_rules()	s03gmain.c
0	main()	reset_clips()	s03gmain.c
0	main()	run()	s03gmain.c
0	main()	set_conserve()	s03gmain.c
0	setWindow()	none	s03twind.c
4	usrfuncs()	define_function()	s03gmain.c

3.4. Summary of C Functions by Called Function

The data shown in Table 3.4. can be used by the programmer to associate functional areas and functions with each of the primary Scenarist C functions. The data are organized alphabetically by function name (Column III). Each line entry for a function identifies the function from which it was called.

Table 3.4. Alphabetical listing of Scenarist C functions (Column III) and the corresponding C function (Column II) from which it is called. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
I	II	III	IV
4	_spiralsearch()	(*_suitabilityfunction())	s03kpsr.c
4	_repositionsubunitsbyrule()	(*actionfunctionpointer())	s03kpsr.c
4	_repositionsubunitsbyrule()	(*preprocessingfnpointer())	s03kpsr.c
4	_repositionsubunitsbyrule()	(*suitabilityfunctionpointer())	s03kpsr.c
0	_main_menu_selection()	_About()	s03wgrap.c
4	_suitability0201()	_accessibility()	s03kpsr2.c
6	_map_menu_selection()	_Add_Vector_Map_without_Labels()	s03ygra2.c
6	_map_menu_selection()	_Add_Vector_Map_with_Labels()	s03ygra2.c
0	_display_intro_screens()	_call_continue()	s03wgrap.c
6	_readmapheader()	_call_continue()	s03qgtfl.c
6	_resetmaplocpoint()	_call_continue()	s03qgtfl.c
3	_About()	_call_continueb()	s03wgrap.c
5	_Rules()	_call_continueb()	s03wgrap.c
7	_Scenario_Generation()	_call_continueb()	s03wgrap.c
8	_Hardware_Setup()	_call_continueb()	s03vmous.c
6	_map_menu_selection()	_Change_Map_Files()	s03ygra2.c
6	_map_menu_selection()	_Change_Map_Location()	s03ygra2.c
4	_repositionsubunitsbyuser()	_clearfoot()	s03jpsu.c
4	_unit_menu_selection()	_Copy_Unit()	s03ygra2.c
4	_Copy_Unit()	_copyunit()	s03ygra2.c
4	_preprocessing0201()	_createaccessibilitymap()	s03kpsr2.c
4	_unit_menu_selection()	_Define_Unit()	s03ygra2.c
4	_Define_Unit()	_defineunit()	s03ygra2.c
4	_unit_menu_selection()	_Delete_Unit()	s03ygra2.c
4	_Delete_Unit()	_deleteunit()	s03ygra2.c
0	main()	_display_intro_screens()	s03gmain.c
0	main()	_display_screen_windows()	s03gmain.c
4	_unit_menu_selection()	_Display_Unit()	s03ygra2.c
0	_main_menu_selection()	_display_windowd()	s03wgrap.c
4	_unit_menu_selection()	_display_windowd()	s03ygra2.c
6	_map_menu_selection()	_display_windowd()	s03ygra2.c
4	_Display_Unit()	_displayunit()	s03ygra2.c
4	_clipssuitability0101()	_distancetofeba()	s03kpsr.c
4	_suitability0201()	_disttootherunits()	s03kpsr2.c
6	_map_menu_selection()	_Draw_Elevation_Map()	s03ygra2.c
6	_map_menu_selection()	_Draw_Road_Map()	s03ygra2.c
6	_map_menu_selection()	_Draw_Terrain_Map()	s03ygra2.c
6	_map_menu_selection()	_Draw_Vector_Map_without_Labels()	s03ygra2.c
4	_Add_Vector_Map_without_Labels()	_drawmap()	s03ygra2.c
4	_Add_Vector_Map_with_Labels()	_drawmap()	s03ygra2.c
4	_displayunit()	_drawmap()	s03mcopy.c
4	_Draw_Elevation_Map()	_drawmap()	s03ygra2.c
4	_Draw_Road_Map()	_drawmap()	s03ygra2.c

Table 3.4. (Continued) Alphabetical listing of Scenarist C functions (Column III) and the corresponding C function (Column II) from which it is called. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
I	II	III	IV
4	_Draw_Terrain_Map()	_drawmap()	s03ygra2.c
4	_Draw_Vector_Map_with_Labels()	_drawmap()	s03ygra2.c
4	_Draw_Vector_Map_without_Labels()	_drawmap()	s03ygra2.c
4	_displayunit()	_drawunit()	s03mcopy.c
4	_Place_Unit_on_Map()	_drawunit()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_drawunit()	s03ygra2.c
4	_Copy_Unit()	_editunit()	s03ygra2.c
4	_Define_Unit()	_editunit()	s03ygra2.c

4	_Display_Unit()	_editunit()	s03ygra2.c
4	_action0201()	_elevation()	s03kpsr2.c
4	_horizonangle()	_elevation()	s03kpsr.c
4	_LOS()	_elevation()	s03kpsr2.c
4	_slopetorearcell()	_elevation()	s03kpsr2.c
4	_repositionsubunitsbyrule()	_mapcellsizestdcoords()	s03kpsr.c
2	main()	_getfilenames()	s03gmain.c
4	_copyunit()	_getsubunitbycode()	s03mcopy.c
4	_defineunit()	_getsubunitbycode()	s031defu.c
4	_repositionprogeny()	_getsubunitbycode()	s03ipuu.c
4	_copyunit()	_getunitbycode()	s03mcopy.c
4	_defineunit()	_getunitbycode()	s031defu.c
4	_deleteunit()	_getunitbycode()	s03mcopy.c
4	_displayunit()	_getunitbycode()	s03mcopy.c
4	_lostoheadquarters()	_getunitbycode()	s03kpsr2.c
4	_Place_Unit_on_Map()	_getunitbycode()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_getunitbycode()	s03ygra2.c
4	_repositionprogeny()	_getunitbycode()	s03ipuu.c
4	_repositionsubunitsbyuser()	_getunitbycode()	s03jpsu.c
4	_repositionunitbyuser()	_getunitbycode()	s03ipuu.c
6	_setcoordsforzoommapb()	_getunitbycode()	s03nmap.c
4	_displayunit()	_getunitbyidno()	s03mcopy.c
4	_displayunit()	_getunitbyindex()	s03mcopy.c
0	_main_menu_selection()	_Hardware_Setup()	s03wgrap.c
4	_clipssuitability0101()	_horizonangle()	s03kpsr.c
4	_lostoheadquarters()	_LOS()	s03kpsr2.c
4	_lostootherunits()	_LOS()	s03kpsr2.c
4	_lostotarget()	_LOS()	s03kpsr2.c
4	_suitability0201()	_lostoheadquarters()	s03kpsr2.c
4	_suitability0201()	_lostootherunits()	s03kpsr2.c
4	_suitability0201()	_lostotarget()	s03kpsr2.c
0	main()	_main_menu_selection()	s03gmain.c
0	_main_menu_selection()	_Map()	s03wgrap.c
6	_Map()	_map_menu_selection()	s03wgrap.c
4	_unit_menu_selection()	_Output_Units()	s03ygra2.c
4	_Output_Units()	_outputunits()	s03ygra2.c

Table 3.4. (Continued) Alphabetical listing of Scenarist C functions (Column III) and the corresponding C function (Column II) from which it is called. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
<u>I</u>	<u>II</u>	<u>III</u>	<u>IV</u>
6	_map_menu_selection()	_Place_Unit_on_Map()	s03ygra2.c
6	_map_menu_selection()	_Print_Map()	s03ygra2.c
4	_Print_Map()	_printscreen()	s03ygra2.c
1	main()	_project_selection()	s03gmain.c 0
	_main_menu_selection()	_Quit()	s03wgrap.c
2	_getfilenames()	_readmapdata()	s03qgtf1.c
4	_Change_Map_Location()	_readmapdata()	s03ygra2.c
2	_getfilenames()	_readmapheader()	s03qgtf1.c
4	_Change_Map_Files()	_readmapheader()	s03ygra2.c
4	_Change_Map_Files()	_readmapheader()	s03ygra2.c
4	_Change_Map_Location()	_readmapheader()	s03ygra2.c
4	_unit_menu_selection()	_Reposition_FEBA()	s03ygra2.c
4	_unit_menu_selection()	_Reposition_Subunits_by_Rules()	s03ygra2.c
4	_unit_menu_selection()	_Reposition_Subunits_by_User()	s03ygra2.c
4	_unit_menu_selection()	_Reposition_Unit_by_User()	s03ygra2.c
4	_Reposition_FEBA()	_repositionfeba()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_repositionprogeny()	s03ygra2.c
4	_repositionsubunitsbyuser()	_repositionprogeny()	s03jpsu.c
4	_repositionunitbyuser()	_repositionprogeny()	s03ipuu.c
4	_Reposition_Subunits_by_User()	_repositionsubunitsbyuser()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_repositionsubunitsbyrule()	s03ygra2.c

4	_Reposition_Unit_by_User()	_repositionunitbyuser()	s03ygra2.c
2	_getfilenames()	_resetmaplocpoint()	s03qgftfl.c
4	_Change_Map_Files()	_resetmaplocpoint()	s03ygra2.c
4	_Change_Map_Location()	_resetmaplocpoint()	s03ygra2.c
4	_createaccessibilitymap()	_road()	s03kpsr2.c
0	_main_menu_selection()	_Rules()	s03wgrap.c
0	main()	_run_demo_script()	s03gmain.c
0	_main_menu_selection()	_Scenario_Generation()	s03wgrap.c
4	_Zoom_Map()	_setcoordsforzoommapb()	s03ygra2.c
6	_setcoordsforzoommapb()	_setcoordsforzoommap()	s03nmap.c
4	_displayunit()	_setcootdsforzoommap()	s03mcopy.c
0	main()	_setup_screen_windows()	s03gmain.c
4	_createaccessibilitymap()	_slopetorearcell()	s03kpsr2.c
4	_repositionsubunitsbyrule()	_spiralsearch()	s03kpsr.c
4	_clipssuitability0101()	_terrain()	s03kpsr.c
4	_createaccessibilitymap()	_terrain()	s03kpsr2.c
4	_suitability0201()	_terrain()	s03kpsr2.c
4	_clipssuitability0101()	_transfstdtoreal()	s03kpsr.c
4	_copyunit()	_transfstdtoreal()	s03mcopy.c
4	_defineunit()	_transfstdtoreal()	s03ldefu.c
4	_distancetofeba()	_transfstdtoreal()	s03kpsr.c
4	_disttootherunits()	_transfstdtoreal()	s03kpsr2.c

Table 3.4. (Continued) Alphabetical listing of Scenarist C functions (Column III) and the corresponding C function (Column II) from which it is called. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
<u>I</u>	<u>II</u>	<u>III</u>	<u>IV</u>
4	_drawunit()	_transfstdtoreal()	s03pdraw.c
4	_elevation()	_transfstdtoreal()	s03kpsr.c
4	_mapcellsizestdcoords()	_transfstdtoreal()	s03kpsr.c
4	_horizonangle()	_transfstdtoreal()	s03kpsr.c
4	_outputunits()	_transfstdtoreal()	s03mcopy.c
4	_repositionprogeny()	_transfstdtoreal()	s03ipuu.c
4	_road()	_transfstdtoreal()	s03kpsr2.c
4	_slopetorearcell()	_transfstdtoreal()	s03kpsr2.c
4	_suitability0101()	_transfstdtoreal()	s03kpsr.c
4	_suitability0201()	_transfstdtoreal()	s03kpsr2.c
4	_terrain()	_transfstdtoreal()	s03kpsr.c
4	_Units()	_unit_menu_selection()	s03wgrap.c
0	_main_menu_selection()	_Units()	s03wgrap.c
0	_display_screen_windows()	_windowa()	s03wgrap.c
0	_main_menu_selection()	_windowa()	s03wgrap.c
4	_unit_menu_selection()	_windowa()	s03ygra2.c
4	_unit_menu_selection()	_windowa()	s03ygra2.c
6	_map_menu_selection()	_windowa()	s03wgrap.c
0	_display_screen_windows()	_windowb()	s03wgrap.c
1	_project_selection()	_windowb()	s03wgrap.c
3	_About()	_windowb()	s03wgrap.c
4	_defineunit()	_windowb()	s03ldefu.c
5	_Rules()	_windowb()	s03wgrap.c
6	_Map()	_windowb()	s03wgrap.c
7	_Scenario_Generation()	_windowb()	s03wgrap.c
8	_Hardware_Setup()	_windowb()	s03vmous.c
4	_copyunit()	_windowb1()	s03mcopy.c
4	_copyunit()	_windowb2()	s03mcopy.c
4	_copyunit()	_windowb3()	s03mcopy.c
4	_defineunit()	_windowbenter()	s03ldefu.c
4	_Place_Unit_on_Map()	_windowbenter()	s03ygra2.c
4	_Reposition_Subunits_by_Rules()	_windowbenter()	s03ygra2.c
4	_repositionsubunitsbyrule()	_windowbenter()	s03kpsr.c
4	CLIPSval_0101(),CLIPSval_0201()	_windowbenter()	s03gmain.c
0	_display_screen_windows()	_windowc()	s03wgrap.c
4	_copyunit()	_windowc()	s03mcopy.c

4	<code>_defineunit()</code>	<code>_windowc()</code>	<code>s03ldefu.c</code>
4	<code>_Reposition_Subunits_by_Rules()</code>	<code>_windowc()</code>	<code>s03ygra2.c</code>
4	<code>_repositionsubunitsbyuser()</code>	<code>_windowc()</code>	<code>s03jpsu.c</code>
4	<code>_repositionunitbyuser()</code>	<code>_windowc()</code>	<code>s03ipuu.c</code>
4	<code>CLIPsval_0101(),CLIPsval_0201()</code>	<code>_windowc()</code>	<code>s03gmain.c</code>
4	<code>_Place_Unit_on_Map()</code>	<code>_windowcenter()</code>	<code>s03ygra2.c</code>
0	<code>_display_screen_windows()</code>	<code>_windowd()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_windowd()</code>	<code>s03wgrap.c</code>

Table 3.4. (Continued) Alphabetical listing of Scenarist C functions (Column III) and the corresponding C function (Column II) from which it is called. Functional area (identified in Table 3.1) is given in Column I and C source code file is given in Column IV.

COLUMN NO.			
<u>I</u>	<u>II</u>	<u>III</u>	<u>IV</u>
1	<code>_project_selection()</code>	<code>_windowd()</code>	<code>s03wgrap.c</code>
4	<code>_unit_menu_selection()</code>	<code>_windowd()</code>	<code>s03ygra2.c</code>
6	<code>_map_menu_selection()</code>	<code>_windowd()</code>	<code>s03ygra2.c</code>
0	<code>_display_screen_windows()</code>	<code>_windowe()</code>	<code>s03wgrap.c</code>
4	<code>_defineunit()</code>	<code>_windowe()</code>	<code>s03ldefu.c</code>
4	<code>CLIPsval_0101(),CLIPsval_0201()</code>	<code>_windowe()</code>	<code>s03gmain.c</code>
4	<code>_Reposition_Subunits_by_Rules()</code>	<code>_windoweenter()</code>	<code>s03ygra2.c</code>
4	<code>_repositionsubunitsbyrule()</code>	<code>_windoweenter()</code>	<code>s03kpsr.c</code>
0	<code>_display_screen_windows()</code>	<code>_windowscreen()</code>	<code>s03wgrap.c</code>
0	<code>_main_menu_selection()</code>	<code>_windowscreen()</code>	<code>s03wgrap.c</code>
4	<code>_unit_menu_selection()</code>	<code>_windowscreen()</code>	<code>s03ygra2.c</code>
0	<code>_display_intro_screens()</code>	<code>_windowscreenblue()</code>	<code>s03wgrap.c</code>
6	<code>_readmapdata()</code>	<code>_windowscreenblue()</code>	<code>s03qgtfl.c</code>
6	<code>_readmapheader()</code>	<code>_windowscreenblue()</code>	<code>s03qgtfl.c</code>
6	<code>_map_menu_selection()</code>	<code>_Zoom_Map()</code>	<code>s03ygra2.c</code>
0	<code>main()</code>	<code>close_dribble()</code>	<code>s03gmain.c</code>
9	<code>_Quit()</code>	<code>close_dribble()</code>	<code>s03wgrap.c</code>
4	<code>usrfuncs()</code>	<code>define_function()</code>	<code>s03gmain.c</code>
0	<code>main()</code>	<code>init_clips()</code>	<code>s03gmain.c</code>
0	<code>main()</code>	<code>load_rules()</code>	<code>s03gmain.c</code>
0	<code>main()</code>	<code>reset_clips()</code>	<code>s03gmain.c</code>
0	<code>main()</code>	<code>run()</code>	<code>s03gmain.c</code>
4	<code>_clipssuitability0101()</code>	<code>run()</code>	<code>s03kpsr.c</code>
0	<code>main()</code>	<code>set_conserve()</code>	<code>s03gmain.c</code>

3.5. Summary of Scenarist Data Files

The Scenarist data base files and a brief description of each file is given in Table 3.5. below.

The xxxx in a file name is used to indicate a project/sequence number within project combination. For example, 0201 indicates project 02 and file sequence 01 within that project.

Table 3.5. Alphabetical listing of Scenarist data base files.

<u>File Name</u>	<u>Description</u>
<code>clipxxxx.fil</code>	CLIPS rule file
<code>eqptxxxx.fil</code>	Equipment file (descriptions of equipment)
<code>febaxxxx.fil</code>	FEBA file (locations of up to five points on FEBA)
<code>genuxxxx.fil</code>	Generic unit file (binary file -- no listing)
<code>geoaxxxx.fil</code>	Vector map area objects (e.g., towns, lakes)
<code>geocxxxx.fil</code>	Elevation map cell data
<code>geodxxxx.fil</code>	Terrain-type map cell data
<code>geolxxxx.fil</code>	Vector linear objects (e.g., roads, rivers)
<code>geopxxxx.fil</code>	Vector point objects
<code>georxxxx.fil</code>	Road availability map cell data
<code>highlits.fil</code>	Summary descriptive information about the Scenarist
<code>intro.fil</code>	Scenarist introduction screen
<code>outxxxx.fil</code>	Formatted output file containing specific unit data

platxxxx.fil	Platform file (descriptions of platforms)
projxxxx.fil	Project File: Names of all other Scenarist data files
scraxxxx.fil	Scratch file (binary file -- no listing)
specxxxx.fil	Specific unit file (binary file -- no listing)
symbxxxx.fil	Military unit symbols, labels
titlxxxx.fil	Map parameters and map title

4.0. Data Flow Diagrams

The data flow diagram is a form of dependency diagram that is commonly used to show overall data flow through a system. It shows data stores and external sources and destinations of data. Its primary purpose as a systems analysis tool is to show the basic procedural components of the system and the data that pass among them. The data flow diagrams shown in the following sections are organized by the main processes used in the Scenarist and shown in Table 3.1. The functional area number serves as the cross reference between the data flow diagram and the data presented in Table 3.1.

4.1. Project File Selection - Functional Area 1.

The data flow diagram for the Project File Selection function is shown in Figure 4.1-1.

4.2. Set Initial Map Location Point - Functional Area 2.

The data flow diagram for the Set Initial Map Location Point function is shown in Figure 4.2-1.

4.3. Brief System Description - Functional Area 3.

The data flow diagram for the Brief System Description function is shown in Figure 4.3-1.

4.4. Processing Military Units - Functional Area 4.

The data flow diagrams for the Processing Military Units function are shown in Figure 4.4-1 to 4.4-10.

4.4-1

4.4-2

4.4-3

4.4-4

4.4-5

4.4-6

4.4-7

4.4-8

4.4-9

4.4-10

4.5. Entry Point for CLIPS Integrated Rule Editor - Functional Area 5.

The data flow diagram for the Rules function is shown in Figure 4.5-1.

4.6. Processing Maps - Functional Area 6.

The data flow diagrams for the Processing Maps function are shown in Figure 4.6-1 to Figure 4.6-11.

4.6-2

4.6-3

4.6.4

4.6-5

4.6.6

4.6-7

4.6-8

4.6-9

4.6-10

4.7. Entry Point for Large-scale Scenario Generation - Functional Area 7.

The data flow diagram for the Large-scale Scenario Generation function is shown in Figure 4.7-1.

4.8. Display Hardware Characteristics - Functional Area 8.

The data flow diagram for the Display Hardware Characteristics function is shown in Figure 4.8-1.

4.9. Exit From Scenarist - Functional Area 9.

The data flow diagram for the Exit From Scenarist function is shown in Figure 4.9-1.

5.0. Input/Output Data Elements

5.1. Data Structures

This section is no longer used. (See discussion in Section 1.1 -- data structures are now described in the Final Report.)

5.2. Output Data Interface File

Data can be transferred between the Scenarist and other system(s) via an ASCII output file. This formatted file contains unit data, including the coordinates of the final placement and unit identification information. To do this, the user selects the Output Unit option from the Unit Menu screen and inputs a disk file name. Hard copy output also can be generated using this option. An example of this output for a test deployment is presented in Appendix B.

6.0. Description of System Modules

The description of system modules uses the same basic structure as followed in the other sections. That is, the following sections are organized by the main Scenarist processes. To as great extent as possible, the name of the menu selection item, such as Draw Terrain Map, has been chosen to suggest the function that is performed when that option is clicked using the left mouse button. The descriptions add to this intuitive notion and provide the programmer with more detail.

6.1. Project File Selection

The Project File Selection option occurs when the Scenarist software starts executing. The local DOS directory is searched for files matching the wild card pattern PROJ*.FIL. File names matching this pattern are displayed on the screen and the user clicks on the highlighted file name to select it. The user is asked to confirm the selection.

The project file is an ASCII file containing the names of the data base files and user-defined suitability function name that the Scenarist uses at startup. The project file, PROJ0201.FIL, used for the start of the TRAILBLAZER example is shown below.

File name: PROJ0201.FIL

```
intro.fil
```

titl01.fil
geod0201.fil
geoc0201.fil
geor0201.fil
geoa02.fil
geol02.fil
geop02.fil
genu01.fil
spec02.fil
scra01.fil
symb01.fil
plat01.fil
eqpt01.fil
feba02.fil
_suitability0201
_preprocessing0201
_action0201
_clipssuitability0201
clip0201.fil

The content and format of the various files used by the Scenarist system are described in the Final Report.

6.2. Set Initial Map Location Point

The Set Initial Map Location Point option occurs when the Scenarist software starts executing. The project file identifies the map files and, at this point, the user can specify a map coordinate for either the center of the map display or the upper left-hand corner.

The user can alter the location point and/or the map files at any time during execution by selecting under the Processing Maps function either Change Map Location or Change Map Files.

6.3. Brief System Description

This option provides the user with point of contact address and phone information for both US Army CECOM and Vista Research Corporation.

6.4. Processing Military Units

This option provides the user with ten sub-options for handling the creation, modification, positioning, and displaying of unit data. Each option is described below. Sample input data used in the Beqaa Valley and TRAILBLAZER applications is shown in Appendix A of this document. (Note: See the Test Report for additional information about the TRAILBLAZER application.)

Define Unit

This option is used to define a new generic or specific unit in the data base.

Copy Unit

This option creates a new generic or specific unit in the data base by copying the contents of an existing record into the data structure of the new unit record.

The Scenarist allows the user four copy options:

- Copy from Generic to Specific
- Copy from Generic to Generic
- Copy from Specific to Specific
- Copy from Specific to Generic

Delete Unit

This option is used to delete a generic or specific unit from the data base.
Reposition Unit by User

This option allows the user to reposition a unit that has NO parent unit. The user inputs the unit code of the unit to be moved and the Scenarist checks on whether a parent unit is defined for this unit. If there is no parent unit, then the input process continues; otherwise the user is returned to the unit menu and is prompted for another selection.

This process moves the selected unit to the user designated location repositions all subunits automatically.

Reposition Subunits by User

This option allows the user to reposition a subordinate unit. Input data are required, if already defined in the record, for location point, radius, and placement code for each large minor subarea item and location points and placement code for point items.

Reposition Subunits by Rules

This option repositions subunits using the CLIPS expert system knowledge base.

Reposition FEBA

This option allows the user to review and/or change the location of the FEBA. The existing points are displayed on the screen and the user is prompted for the changes.

Display Unit

This option allows the user to display the unit on the graphics terminal. The unit is drawn in the map window on the elevation map using the zoom characteristics in effect at the time this option is chosen.

Output Unit

This option allows the user to obtain hard copy output of the specific unit and generic unit files. The required printer is a HP Laser Jet Series II compatible printer.

Return to Previous Menu

This option returns the user to the Main Menu selection screen.

6.5. Entry Point for CLIPS Integrated Rule Editor

This option enables the user to display the data contained in the CLIPS rule file, about rule definition and source.

This option is also the "hook" in the Scenarist code to interface with a CLIPS expert system rule editor (a suggested area for future development).

6.6. Processing Maps

Draw Terrain Map

This option draws the terrain map on the graphics monitor. Map resolution, map location point, and zoom window characteristics have been set prior to selecting this option.

Draw Elevation Map

This option draws the elevation map on the graphics monitor. Map resolution, map location point, and zoom window characteristics have been set prior to selecting this option.

Draw Road Map

This option draws the road availability map on the graphics monitor. Map resolution, map location point, and zoom window characteristics have been set prior to selecting this option. (Note: A road availability cell map is distinguished from a vector road map. The road availability cell map indicates whether a road is present in a map cell. A vector road map draws lines (representing roads) on the screen. A vector road map is drawn by calling the function Add Vector Map with Labels or the function Add Vector Map without Labels.

Add Vector Map with Labels

This option adds the vector map with labels to whatever already is drawn in the map window. Map resolution, map location point, and zoom window characteristics have been set prior to selecting this option.

In some situations, the display of the label information can make the graphics window appear very "busy". The Scenarist provides the Add Vector Map without Labels as an alternative selection.

Add Vector Map without Labels

This option adds the vector map without labels to whatever already is drawn in the map window. Map resolution, map location point, and zoom window characteristics have been set prior to selecting this option.

Place Unit on Map

This option allows the user to identify a specific unit, by unit id code, and the Scenarist will place the unit on the map using the coordinate information in the data base.

Zoom Map

This option supports three choices:

1. Return to the original map.
2. Set the zoom map coordinates as the bounding rectangle of a unit.
3. Specify the upper left corner coordinates and width of the map.

Change Map Location

This option supports two choices:

1. Specify upper left corner for map intersection location point.
2. Specify map center.

Change Map Files

This option allows the user to access different map data files for the terrain-type file (GEODxxx.FIL), the elevation file (GEOCxxx.FIL), and the road file (GEORxxxx.FIL). Maps of different resolution can be accessed to facilitate the analysis of unit placement. The Scenarist can handle simultaneously map files of different resolutions (i.e. terrain-type file at 100-meter resolution and elevation file at 30-meter resolution).

Print Map

This option allows the user to obtain hard copy output of the graphics screen. The required printer is a HP Laser Jet Series II compatible printer.

Return to Previous Menu

This option returns the user to the Main Menu selection screen.

6.7. Entry Point for Large-Scale Scenario Generation

This option is the "hook" in the Scenarist code to interface with processes that would reposition an entire data base of units using the Scenarist and the CLIPS knowledge base.

The analysis of the placements would be accomplished following the processing and would be supported by a number of reports indicating the degree to which canonical placements were not suitable based on the rules in the knowledge base.

The Scenarist would "thread" its way through this trace data and the user would review/approve the placements. Final unit placement codes would be modified in the data base records at this time.

This is an area for future development.

6.8. Display Hardware Characteristics

This option displays to the screen several of the PC system hardware characteristics determined from a query of the video configuration data structure.

6.9. Exit from Scenarist

This option exits the user from the Scenarist software.

Appendix A. Scenarist Test Deployment

A. Units Used in Beqaa Valley Problem

Generic Unit file:genu01.fil
Specific Unit file: spec01.fil
FEBA File: feba01.fil

1. Define a Generic Division of Type 1

Name: Division1
Code: 1 1 1 1 0 0 0 0 0 0
Type: 1
Front/Rear boundary points: .5 .5
NFRONT: 2
NFRONT sets of (echelon,no.,type,idno): 5 1 1 0 5 2 1 0
NFRONT-1 boundary points: .5 .5
NREAR: 0
NALLAREA: 0
NMAJORSUBAREA: 0
NMINORSUBAREAREL: 0
NMINORSUBAREAABS: 0
NPOINT: 0

2. Copy Generic Division of Type 1 to a Specific Division, BLUE

New Unit's Name: Division1
Old Code: 1 1 1 1 0 0 0 0 0 0
New Code: 1 1 1 1 0 0 0 0 0 0
Parent Code: 1 1 1 0 0 0 0 0 0 0
Idno,parentidno: 141 131
4 Corners: 92000 30000 96000 46000 74000 48000 72000 34000
(Note: for Spearfish, SD, map use (in file spec02.fil):
610000 4925000 590000 4927000 588000 4905000 608000 4903000)
Objective: yes (i.e., input a "1")
(Note: for Spearfish, SD, map use 0 (in file spec02.fil)
Mission type: 1
2 Corners: 124000 36000 128000 32000

No. pts on avenue of approach: 1
Coords: 112000 38000

3. Copy Generic Division of Type 1 to a Specific Division, RED

New Unit's Name: RedDivision1
Old Code: 1 1 1 1 0 0 0 0 0 0 0
New Code: 2 1 1 1 0 0 0 0 0 0 0
Parent Code: 2 1 1 0 0 0 0 0 0 0 0
Idno,parentidno: 241 231
4 Corners: 110000 44000 106000 28000 128000 30000 130000 40000
(Note: for Spearfish, SD, map use (in file spec02.fil):
591000 4937000 611000 4935000 613000 4955000 593000 4957000)

Objective: no (i.e., hit "enter")

4. Define a Generic Brigade of Type 1

Name: MechBde1
Code: 1 1 1 1 1 0 0 0 0 0 0
Type: 1
Front/Rear Boundary Points: 0 0
NFRONT: 0
NREAR: 0
NALLAREA: 2
NALLAREA sets of Echelon,no.,type,idno: 8 1 1 0 8 2 2 0
NALLAREA sets of symbol location pts: .4 .4 .6 .55
(Note: this example is not realistic -- in reality, the
radar units are division assets, not brigade assets.)
NMAJORSUBAREA: 0
NMINORSUBAREAREL: 0
NMINORSUBAREAABS: 0
NPOINT: 0

5. Copy Generic Brigade of Type 1 to a Specific Brigade (Right Front of Division), BLUE

Name: MechBde1
Old Code: 1 1 1 1 1 0 0 0 0 0 0
New Code: 1 1 1 1 1 0 0 0 0 0 0
Parent Code: 1 1 1 1 0 0 0 0 0 0 0
Idno,parentidno: 151 141
Objective: no

6. Copy Generic Brigade of Type 1 to a Specific Brigade (Left Front of Division), BLUE

Name: MechBde2
Old Code: 1 1 1 1 1 0 0 0 0 0 0
New Code: 1 1 1 1 2 0 0 0 0 0 0
Parent Code: 1 1 1 1 0 0 0 0 0 0 0
Idno,parentidno: 152 141
Objective: no

7. Define a Generic Field Artillery Battery of Type 1

Name: FA_Batt1
Code: 1 1 1 1 1 0 0 1 0 0 0
Type: 1
Front/Rear Boundary Points: 0 0
NFRONT: 0
NREAR: 0
NALLAREA: 0
NMAJORSUBAREA: 0
NMINORSUBAREAREL: 0
NMINORSUBAREAABS: 4
NMINORSUBAREAABS sets of (echelon,no.,type,idno):

10 1 1 0 10 2 2 0 10 3 2 0 10 4 3 0

NMINORSUBAREEAABS sets of (location point x,y, radius):
.4 .4 50 .13 .4 50 .87 .4 50 .4 .75 50
NPOINT: 0

8. Copy Generic Field Artillery Battery of Type 1 to Specific Field Artillery Battery (in brigade MechBdel, on right front of Division1), BLUE

Name: FA_Batt1
Old Code: 1 1 1 1 1 0 0 1 0 0 0
New Code: 1 1 1 1 1 0 0 1 0 0 0
Parent Code: 1 1 1 1 1 0 0 0 0 0 0
Idno,parentidno: 181 151
Objective: no

9. Define a Generic Air Defense Artillery Battery of Type 1

Name: ADA_Batt1
Code: 1 1 1 1 1 0 0 2 0 0 0
Type: 2
Front/Rear Boundary Points: 0 0
NFRONT: 0
NREAR: 0
NALLAREA: 0
NMAJORSUBAREA: 0
NMINORSUBAREAREL: 0
NMINORSUBAREEAABS: 4
NMINORSUBAREEAABS sets of (echelon,no.,type,idno):
10 1 4 0 10 2 5 0 10 3 5 0 10 4 5 0
NMINORSUBAREEAABS sets of (location point x,y, radius):
.6 .55 50 .13 .2 50 .87 .2 50 .53 .5 50
NPOINT: 0

10. Copy Generic Air Defense Artillery Battery of Type 1 to Specific Field Artillery Battery (in brigade MechBdel, on right front of Division1), BLUE

Name: ADA_Batt1
Old Code: 1 1 1 1 1 0 0 2 0 0 0
New Code: 1 1 1 1 1 0 0 2 0 0 0
Parent Code: 1 1 1 1 1 0 0 0 0 0 0
Idno,parentidno: 182 151
Objective: no

12. FEBA

No. of points on FEBA: 3
Points: 99000 28000 101000 37000 104000 46000
(Note: for Spearfish, SD, map use (in file feba02.fil):
585000 4932000 600000 4931000 615000 493000)

B. Units Used in TRAILBLAZER Problem

Generic Unit file:genu02.fil
Specific Unit file: spec02.fil
FEBA File: feba02.fil

1. Define a Generic Division of Type 2

(Note: this division is assigned type 2 to distinguish it from the division defined above)

Name: Division2
Code: 1 1 1 2 0 0 0 0 0 0
(Note: this division is assigned number 2 to distinguish it from the division defined above.
In order to access units, the Scenarist requires that every unit have a unique code.)
Type: 2
Front/Rear boundary points: .4 .4
NFRONT: 2
NFRONT sets of (echelon,no.,type,idno("ENTI")):5 1 2 0 5 2 2 0
NFRONT-1 boundary points: .5 .5
NREAR: 0

NALLAREA: 1
NALLAREA sets of ENTI: 7 1 1 0 (Note: this is the MI Bn)
NALLAREA sets of symbol location points: .4 .8
NMAJORSUBAREA: 0
NMINORSUBAREAREL: 0
NMINORSUBAREEAABS: 1 (Note: this is the DivHQCo)
NMINORSUBAREEAABS sets of ENTI: 8 1 3 0
NMINORSUBAREEAABS sets of loc pt & rad: .5 .9 200
NPOINT: 0

2. Copy Generic Division of Type 2 to a Specific Division, BLUE

New Unit's Name: Division2
Old Code: 1 1 1 2 0 0 0 0 0 0
New Code: 1 1 1 2 0 0 0 0 0 0
Parent Code: 1 1 1 0 0 0 0 0 0 0
Idno,parentidno: 142 131
4 Corners: 610000 4925000 590000 4927000 588000 4905000 608000 4903000
Objective: no

3. Define a Generic Military Intelligence (MI) Battalion of Type 1

Name: MIBn1
Code: 1 1 1 2 0 0 1 0 0 0
Type: 1
Front/Rear boundary points: 0 0
NFRONT: 0
NREAR: 0
NALLAREA: 1
NALLAREA sets of ENTI: 8 1 5 0 (Note: this is the EW Co)
NALLAREA sets of symbol location points: .25 .35
NMAJORSUBAREA: 0

NMINORSUBAREAREL: 0
NMINORSUBAREEAABS: 1 (Note: this is the MIBnHQCo)
NMINORSUBAREEAABS sets of ENTI: 8 2 4 0
NMINORSUBAREEAABS sets of loc pt & rad: .4 .8 100
NPOINT: 0

4. Copy Generic MI Battalion of Type 1 to a Specific MI Battalion, BLUE

Name: MIBn1
Old Code: 1 1 1 2 0 0 1 0 0 0
New Code: 1 1 1 2 0 0 1 0 0 0
Parent Code: 1 1 1 2 0 0 0 0 0 0
Idno, parentidno: 171, 142
Objective: no

5. Define a Generic Electronic Warfare (EW) Company of Type 1

Name: EWCo1
Code: 1 1 1 2 0 0 1 1 0 0
Type: 1
Front/Rear boundary points: 0 0
NFRONT: 0
NREAR: 0
NALLAREA: 1
NALLAREA sets of ENTI: 9 1 2 0 (Note: this is the SIGINT Processing Platoon (SPP))
NALLAREA sets of symbol location points: .2 .3
NMAJORSUBAREA: 0
NMINORSUBAREAREL: 0
NMINORSUBAREEAABS: 1 (Note: this is the EWHQPlt)
NMINORSUBAREEAABS sets of ENTI: 9 2 1 0
NMINORSUBAREEAABS sets of loc pt & rad: .25 .35 50
NPOINT: 0

6. Copy Generic EW Company of Type 1 to a Specific EW Company, BLUE

Name: EWCo1
Old Code: 1 1 1 2 0 0 1 1 0 0 0
New Code: 1 1 1 2 0 0 1 1 0 0 0
Parent Code: 1 1 1 2 0 0 1 0 0 0 0
Idno, parent idno: 181 171
Objective: no

7. Define a Generic SIGINT Processing Platoon (SPP) of Type 1

Name: SPP1
Code: 1 1 1 2 0 0 1 1 1 0 0
Type: 1
Front/Rear boundary points: 0 0
NFRONT: 0
NREAR: 0

NALLAREA: 2
NALLAREA sets of ENTI: 10 1 6 0 10 2 7 0 (Note: these are the TRAILBLAZER and TEAMPACK sections)
NALLAREA sets of symbol location points: .2 .3 .3 .3
NMAJORSUBAREA: 0
NMINORSUBAREAREL: 0
NMINORSUBAREAABS: 0
NPOINT: 0

8. Copy Generic SPP of Type 1 to a Specific SPP, BLUE

Name: SPP1
Old Code: 1 1 1 2 0 0 1 1 1 0 0
New Code: 1 1 1 2 0 0 1 1 1 0 0
Parent Code: 1 1 1 2 0 0 1 1 0 0 0
Idno, parentidno: 191 181
Objective: no

9. Define a Generic TRAILBLAZER Section of Type 1

Name: TBSec1
Code: 1 1 1 2 0 0 1 1 1 1 0
Type: 1
Front/Rear boundary points: 0 0
NFRONT: 0
NREAR: 0
NALLAREA: 0
NMAJORSUBAREA: 0
NMINORSUBAREAREL: 0
NMINORSUBAREAABS: 5 (Note: these are the five TRAILBLAZER Master Control Stations)
NMINORSUBAREAABS sets of ENTI: 11 1 1 0 11 2 1 0 11 3 1 0 11 4 1 0 11 5 1 0
NMINORSUBAREAABS sets of loc coords, radius: .1 .1 25 .3 .3 25 .5 .1 25 .7 .3 25 .9 .1 25
(Note: these coordinates correspond to the "W" configuration of the canonical TRAILBLAZER laydown)
NPOINT: 0

10. Copy Generic TRAILBLAZER Section of Type 1 to a Specific TRAILBLAZER Section, BLUE

Name: TBSec1
Old Code: 1 1 1 2 0 0 1 1 1 1 0
New Code: 1 1 1 2 0 0 1 1 1 1 0
Parent Code: 1 1 1 2 0 0 1 1 1 0 0
Idno, parentidno: 1101 191
Objective: yes
Mission type: 2
Coordinates of bounding rectangle of objective: 591000 4937000 613000 4955000

11. Define a Generic TEAMPACK Section of Type 2 (Note: no rules were developed for the TEAMPACK -- just the following canonical laydown. The section type is 2 to distinguish it from the TRAILBLAZER section, which is a section of type 1.)

```
Name: TPSec1
Code: 1 1 1 2 0 0 1 1 1 2 0
Type: 2
Front/Rear boundary points: 0 0
NFRONT: 0
NREAR: 0
NALLAREA: 0
NMAJORSUBAREA: 0
NMINORSUBAREAREL: 0
NMINORSUBAREAABS: 3 (Note: these are the three TEAMPACK teams)
NMINORSUBAREAABS sets of ENTI: 11 1 2 0 11 2 2 0 11 3 2 0
NMINORSUBAREAABS sets of loc coords, radius: .1 .05 25 .4 .05 25 .6 .05 25
NPOINT: 0
```

12. Copy Generic TEAMPACK Section of Type 2 to a Specific TEAMPACK Section, BLUE

```
Name: TPSec1
Old Code: 1 1 1 2 0 0 1 1 1 2 0
New Code: 1 1 1 2 0 0 1 1 1 2 0
Parent Code: 1 1 1 2 0 0 1 1 1 0 0
Idno, parentidno: 1102 191
Objective: yes
Mission type: 2
Coordinates of bounding rectangle of objective: 591000 4937000 613000 4955000
```

```
13. FEBA
No. of points on FEBA: 3
Points: 585000 4932000 600000 4931000 615000 493000)
```

```
Code:
1: Side
2: Army
3: Corps
4: Division
5: Brigade
6: Regiment
7: Battalion
8: Company/Battery
9: Platoon
10: Section
11: Squad/Team
(12: Platform)
(13: Equipment)
```

Appendix B. Test Output Data Interface File

Formatted printout of file spec02.fil (Output Unit option), showing organizational structure and locations of all units in file.

```

      Index Idno      Code      Sde Ech Typ      No Echelon Name/Subitem Type
Unit:      1      141      11110000000 1  4  1      1 Div      Division1
Location (coordinates of four corners):
  610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
No. of area subordinate units = 2
Subunit: 1      0      11111000000 1  5  1      1 Bde      MechBde
Placement code: 0
Location (coordinates of four corners):
  610000.00 4925000.00 600000.00 4926000.00 599000.00 4915000.00 609000.00 4914000.00
```

Subunit: 2 0 11112000000 1 5 1 2 Bde MechBde
 Placement code: 0
 Location (coordinates of four corners):
 600000.00 4926000.00 590000.00 4927000.00 589000.00 4916000.00 599000.00 4915000.00
 No. of point subordinate items (units, platforms, or equipment) = 0

Unit: 2 241 21110000000 2 4 1 1 Div RedDivision1
 Location (coordinates of four corners):
 591000.00 4937000.00 611000.00 4935000.00 613000.00 4955000.00 593000.00 4957000.00
 No. of area subordinate units = 2
 Subunit: 1 0 21111000000 2 5 1 1 Bde Bde1
 Placement code: 0
 Location (coordinates of four corners):
 591000.00 4937000.00 601000.00 4936000.00 602000.00 4946000.00 592000.00 4947000.00
 Subunit: 2 0 21112000000 2 5 1 2 Bde Bde1
 Placement code: 0
 Location (coordinates of four corners):
 601000.00 4936000.00 611000.00 4935000.00 612000.00 4945000.00 602000.00 4946000.00
 No. of point subordinate items (units, platforms, or equipment) = 0

Unit: 3 151 11111000000 1 5 1 1 Bde MechBde1
 Location (coordinates of four corners):
 610000.00 4925000.00 600000.00 4926000.00 599000.00 4915000.00 609000.00 4914000.00
 No. of area subordinate units = 2
 Subunit: 1 0 11111001000 1 8 1 1 Co/Btry FA_Batt
 Placement code: 0

Location (coordinates of four corners):
 610000.00 4925000.00 600000.00 4926000.00 599000.00 4915000.00 609000.00 4914000.00
 Subunit: 2 0 11111002000 1 8 2 2 Co/Btry ADA_Batt
 Placement code: 0
 Location (coordinates of four corners):
 610000.00 4925000.00 600000.00 4926000.00 599000.00 4915000.00 609000.00 4914000.00
 No. of point subordinate items (units, platforms, or equipment) = 0

Unit: 4 152 11112000000 1 5 1 2 Bde MechBde2
 Location (coordinates of four corners):
 600000.00 4926000.00 590000.00 4927000.00 589000.00 4916000.00 599000.00 4915000.00
 No. of area subordinate units = 2
 Subunit: 1 0 11112001000 1 8 1 1 Co/Btry FA_Batt
 Placement code: 0
 Location (coordinates of four corners):
 600000.00 4926000.00 590000.00 4927000.00 589000.00 4916000.00 599000.00 4915000.00
 Subunit: 2 0 11112002000 1 8 2 2 Co/Btry ADA_Batt
 Placement code: 0
 Location (coordinates of four corners):
 600000.00 4926000.00 590000.00 4927000.00 589000.00 4916000.00 599000.00 4915000.00
 No. of point subordinate items (units, platforms, or equipment) = 0

Unit: 5 181 11111001000 1 8 1 1 Co/Btry FA_Batt1
 Location (coordinates of four corners):
 610000.00 4925000.00 600000.00 4926000.00 599000.00 4915000.00 609000.00 4914000.00
 No. of area subordinate units = 4
 Subunit: 1 0 11111001010 1 10 1 1 Sec FA_HQ
 Placement code: 0
 Location (coordinates of four corners):
 605600.00 4921000.00 605600.00 4921000.00 605600.00 4921000.00 605600.00 4921000.00
 Subunit: 2 0 11111001020 1 10 2 2 Sec TPQ-36
 Placement code: 0
 Location (coordinates of four corners):
 608300.00 4920730.00 608300.00 4920730.00 608300.00 4920730.00 608300.00 4920730.00
 Subunit: 3 0 11111001030 1 10 2 3 Sec TPQ-36
 Placement code: 0
 Location (coordinates of four corners):
 600900.00 4921470.00 600900.00 4921470.00 600900.00 4921470.00 600900.00 4921470.00
 Subunit: 4 0 11111001040 1 10 3 4 Sec TPQ-37

Placement code: 0
 Location (coordinates of four corners):
 605250.00 4917150.00 605250.00 4917150.00 605250.00 4917150.00 605250.00 4917150.00
 No. of point subordinate items (units, platforms, or equipment) = 0
 Unit: 6 182 11111002000 1 8 2 2 Co/Btry ADA_Batt1

Location (coordinates of four corners):
 610000.00 4925000.00 600000.00 4926000.00 599000.00 4915000.00 609000.00 4914000.00
 No. of area subordinate units = 4
 Subunit: 1 0 11111002010 1 10 4 1 Sec ADA_HQ
 Placement code: 0

Location (coordinates of four corners):
 603450.00 4919550.00 603450.00 4919550.00 603450.00 4919550.00 603450.00 4919550.00
 Subunit: 2 0 11111002020 1 10 5 2 Sec FAAR
 Placement code: 0

Location (coordinates of four corners):
 608500.00 4922930.00 608500.00 4922930.00 608500.00 4922930.00 608500.00 4922930.00
 Subunit: 3 0 11111002030 1 10 5 3 Sec FAAR
 Placement code: 0

Location (coordinates of four corners):
 601100.00 4923670.00 601100.00 4923670.00 601100.00 4923670.00 601100.00 4923670.00
 Subunit: 4 0 11111002040 1 10 5 4 Sec FAAR
 Placement code: 0

Location (coordinates of four corners):
 604200.00 4920030.00 604200.00 4920030.00 604200.00 4920030.00 604200.00 4920030.00
 No. of point subordinate items (units, platforms, or equipment) = 0

Unit: 7 142 11120000000 1 4 2 2 Div Division2

Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 No. of area subordinate units = 4
 Subunit: 1 0 11121000000 1 5 2 1 Bde MechBde
 Placement code: 0

Location (coordinates of four corners):
 610000.00 4925000.00 600000.00 4926000.00 599200.00 4917200.00 609200.00 4916200.00
 Subunit: 2 0 11122000000 1 5 2 2 Bde MechBde
 Placement code: 0

Location (coordinates of four corners):
 600000.00 4926000.00 590000.00 4927000.00 589200.00 4918200.00 599200.00 4917200.00
 Subunit: 3 0 11120010000 1 7 1 1 Bn MIBn
 Placement code: 0

Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 Subunit: 4 0 11120001000 1 8 3 1 Co/Btry DivHQCo
 Placement code: 0

Location (coordinates of four corners):
 598200.00 4906200.00 598200.00 4906200.00 598200.00 4906200.00 598200.00 4906200.00
 No. of point subordinate items (units, platforms, or equipment) = 0

Unit: 8 171 11120010000 1 7 1 1 Bn MIBn1

Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 No. of area subordinate units = 2
 Subunit: 1 0 11120011000 1 8 5 1 Co/Btry EWCo
 Placement code: 0

Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 Subunit: 2 0 11120012000 1 8 4 2 Co/Btry MIBnHQCo
 Placement code: 0

Location (coordinates of four corners):
 600400.00 4908200.00 600400.00 4908200.00 600400.00 4908200.00 600400.00 4908200.00
 No. of point subordinate items (units, platforms, or equipment) = 0

Unit: 9 181 11120011000 1 8 1 1 Co/Btry EWCo1
 Location (coordinates of four corners):

610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 No. of area subordinate units = 2
 Subunit: 1 0 11120011100 1 9 2 1 Plt SPP
 Placement code: 0
 Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 Subunit: 2 0 11120011200 1 9 1 2 Plt EWHQPlt
 Placement code: 0
 Location (coordinates of four corners):
 604300.00 4917800.00 604300.00 4917800.00 604300.00 4917800.00 604300.00 4917800.00
 No. of point subordinate items (units, platforms, or equipment) = 0

 Unit: 10 191 11120011100 1 9 1 1 Plt SPPI
 Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 No. of area subordinate units = 2
 Subunit: 1 0 11120011110 1 10 6 1 Sec TBMCs
 Placement code: 0
 Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 Subunit: 2 0 11120011120 1 10 7 2 Sec TPTeam
 Placement code: 0

 Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 No. of point subordinate items (units, platforms, or equipment) = 0

 Unit: 11 1101 11120011110 1 10 1 1 Sec TBSec1
 Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 No. of area subordinate units = 5
 Subunit: 1 0 11120011111 1 11 1 1 Sq MCS
 Placement code: 1
 Location (coordinates of four corners):
 607800.00 4923000.00 607800.00 4923000.00 607800.00 4923000.00 607800.00 4923000.00
 Subunit: 2 0 11120011112 1 11 1 2 Sq MCS
 Placement code: 1
 Location (coordinates of four corners):
 603400.00 4919000.00 603400.00 4919000.00 603400.00 4919000.00 603400.00 4919000.00
 Subunit: 3 0 11120011113 1 11 1 3 Sq MCS
 Placement code: 2
 Location (coordinates of four corners):
 599900.00 4924900.00 599900.00 4924900.00 599900.00 4924900.00 599900.00 4924900.00
 Subunit: 4 0 11120011114 1 11 1 4 Sq MCS
 Placement code: 2
 Location (coordinates of four corners):
 596500.00 4920800.00 596500.00 4920800.00 596500.00 4920800.00 596500.00 4920800.00
 Subunit: 5 0 11120011115 1 11 1 5 Sq MCS
 Placement code: 3
 Location (coordinates of four corners):
 591900.00 4925700.00 591900.00 4925700.00 591900.00 4925700.00 591900.00 4925700.00
 No. of point subordinate items (units, platforms, or equipment) = 0

 Unit: 12 1102 11120011120 1 10 2 2 Sec TPSec1
 Location (coordinates of four corners):
 610000.00 4925000.00 590000.00 4927000.00 588000.00 4905000.00 608000.00 4903000.00
 No. of area subordinate units = 3
 Subunit: 1 0 11120011121 1 11 2 1 Sq TP
 Placement code: 0
 Location (coordinates of four corners):
 607900.00 4924100.00 607900.00 4924100.00 607900.00 4924100.00 607900.00 4924100.00
 Subunit: 2 0 11120011122 1 11 2 2 Sq TP
 Placement code: 0
 Location (coordinates of four corners):
 601900.00 4924700.00 601900.00 4924700.00 601900.00 4924700.00 601900.00 4924700.00
 Subunit: 3 0 11120011123 1 11 2 3 Sq TP

Placement code: 0

Location (coordinates of four corners):

597900.00 4925100.00 597900.00 4925100.00 597900.00 4925100.00 597900.00 4925100.00

No. of point subordinate items (units, platforms, or equipment) = 0