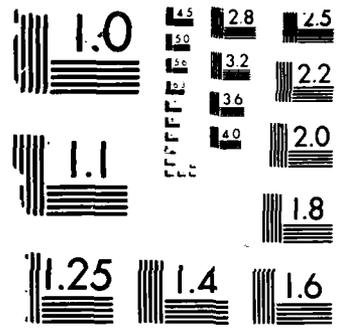


AD-A182 584

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS) VOLUME 8 1/1
USER INTERFACE SUBS (U) GENERAL ELECTRIC CO
SCHENECTADY NY PRODUCTION RESOURCES CONSU F GLANDORF
01 NOV 85 UTP-628144401 F/G 12/5 ML

UNCLASSIFIED

									END			
									8-87			
									DTIC			

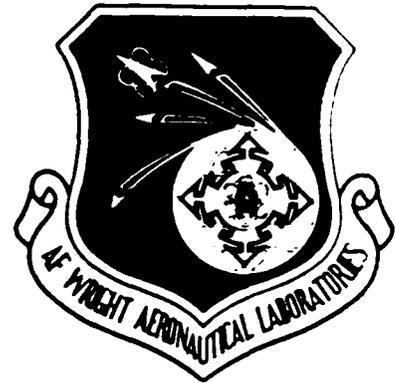


MICROCOPY RESOLUTION TEST CHART
NBS 1963-A

2

AD-A182 584

AFWAL-TR-86-4006
Volume VIII
Part 16



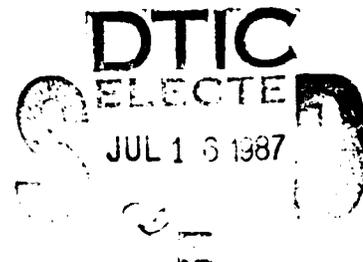
INTEGRATED INFORMATION
SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 16 - Forms Language Compiler Unit Test Plan

General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345

Final Report for Period 22 September 1980 - 31 July 1985
November 1985

Approved for public release; distribution is unlimited.

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533

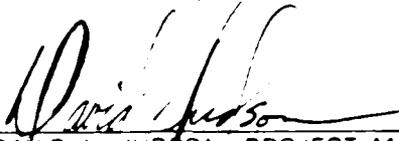


NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.



DAVID L. JUDSON, PROJECT MANAGER
AFWAL/MLTC
WRIGHT PATTERSON AFB OH 45433

5 Aug 1986

DATE

FOR THE COMMANDER



GERALD C. SHUMAKER, BRANCH CHIEF
AFWAL/MLTC
WRIGHT PATTERSON AFB OH 45433

7 Aug 86

DATE

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization, please notify AFWAL/MLTC, W-PAFB OH 45433 to help us maintain a current mailing list."

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

1179 574

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified		1b RESTRICTIVE MARKINGS	
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION AVAILABILITY OF REPORT Approved for public release; distribution is unlimited	
2b DECLASSIFICATION/DOWNGRADING SCHEDULE			
4 PERFORMING ORGANIZATION REPORT NUMBER(S)		5 MONITORING ORGANIZATION REPORT NUMBER(S) AFVAL-TR-86-4006 Vol VIII, Part 16	
6a NAME OF PERFORMING ORGANIZATION General Electric Company Production Resources Consulting	6b OFFICE SYMBOL (if applicable)	7a NAME OF MONITORING ORGANIZATION AFVAL/MLTC	
6c ADDRESS (City, State and ZIP Code) 1 River Road Schenectady, NY 12345		7b ADDRESS (City, State and ZIP Code) WPAFB, OH 45433-6533	
8a NAME OF FUNDING SPONSORING ORGANIZATION Materials Laboratory Air Force Systems Command, USAF	8b OFFICE SYMBOL (if applicable) AFVAL/MLTC	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-80-C-5155	
8c ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433		10 SOURCE OF FUNDING NOS	
		PROGRAM ELEMENT NO 78011F	TASK NO 7500
		WORK UNIT NO 62	WORK UNIT NO 01
11 TITLE (Include Security Classification) (See Reverse)			
12 PERSONAL AUTHOR(S) Glandorf, Frank			
13a TYPE OF REPORT Final Technical Report	13b TIME COVERED 22 Sept 1980 - 31 July 1985	14 DATE OF REPORT (Yr, Mo, Day) 1985 November	15 PAGE COUNT 36
16 SUPPLEMENTARY NOTATION ICAM Project Priority 6201			
The computer software contained herein are theoretical and or references that in no way reflect Air Force-owned or -developed computer software			
17 COSAT CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB GR	
1308	0905		
19 ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>This unit test plan establishes the methodology and procedures used to adequately test the capabilities of the computer programs identified as the Form Definition Language Compiler (FLAN), the Reverse FLAN Compiler (REVFLAN) and Make Includes (MAKINC). FLAN is a compiler which translates Form Definition Language source files into binary Form Definition File format. The binary Form Definition Files are then used as input by the Form Processor for display and entry of data under the control of other application program.</p> <p>MAKINC is a program that creates program variable declarations which correspond to the structure of a form and may be used in application program which make use of the Form Processor calls PDATA and GDATA.</p> <p>REVFLAN is a program used to create an FDL source file from one or more version 1.0 FD files which were created using DEC FMS. The resulting FDL file may then be compiled using FLAN to produce version 2.0 FD files.</p>			
20 DISTRIBUTION AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL David L. Judson		22b TELEPHONE NUMBER (include Area Code) 613-255-0976	22c OFFICE SYMBOL AFVAL/MLTC

11. Title

Integrated Information Support System (IISS)
Vol VIII - User Interface Subsystem
Part 16 - Forms Language Compiler Unit Test Plan

A S D 86 0033
9 Jan 1986

Accession For	
ADIS GRARI	<input checked="" type="checkbox"/>
DIS TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A-1	

PREFACE

This unit test plan covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows.

TASK 4.2

<u>Subcontractors</u>	<u>Role</u>
Boeing Military Aircraft Company (BMAC)	Reviewer
D Appleton Company (DACOM)	Responsible for IDEF support, state-of-the-art literature search
General Dynamics/ Ft. Worth	Responsible for factory view function and information models

<u>Subcontractors</u>	<u>Role</u>
Illinois Institute of Technology	Responsible for factory view function research (IITRI) and information models of small and medium-size business.
North American Rockwell	Reviewer.
Northrop Corporation	Responsible for factory view function and information models.
Pritsker and Associates	Responsible for IDEF2 support.
SofTech	Responsible for IDEFO support.

TASKS 4.3 - 4.9 (TEST BED)

<u>Subcontractors</u>	<u>Role</u>
Boeing Military Aircraft Company (BMAC)	Responsible for consultation on applications of the technology and on IBM computer technology.
Computer Technology Associates (CTA)	Assisted in the areas of communications systems, system design and integration methodology, and design of the Network Transaction Manager.
Control Data Corporation (CDC)	Responsible for the Common Data Model (CDM) implementation and part of the CDM design (shared with DACOM).
D. Appleton Company (DACOM)	Responsible for the overall CDM Subsystem design integration and test plan, as well as part of the design of the CDM (shared with CDC). DACOM also developed the Integration Methodology and did the schema mappings for the Application Subsystems.

<u>Subcontractors</u>	<u>Role</u>
Digital Equipment Corporation (DEC)	Consulting and support of the performance testing and on DEC software and computer systems operation.
McDonnell Douglas Automation Company (McAuto)	Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.
On-Line Software International (OSI)	Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.
Rath and Strong Systems Products (RSSP) (In 1985 became McCormack & Dodge)	Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.
SofTech, Inc.	Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.
Software Performance Engineering (SPE)	Responsible for directing the work on performance evaluation and analysis.
Structural Dynamics Research Corporation (SDRC)	Responsible for the User Interface and Virtual Terminal Interface Subsystems.

Other prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Boeing Military Aircraft Company (BMAC)	1701, 2201, 2202	Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC).

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Control Data Corporation (CDC)	1502, 1701	IISS enhancements to Common Data Model Processor (CDMP).
D. Appleton Company (DACOM)	1502	IISS enhancements to Integration Methodology.
General Electric	1502	Operation of the Test Bed and communications equipment.
Hughes Aircraft Company (HAC)	1701	Test Bed enhancements.
Structural Dynamics Research Corporation (SDRC)	1502, 1701, 1703	IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI).
Systran	1502	Test Bed enhancements. Operation of Test Bed.

TABLE OF CONTENTS

	<u>Page</u>
SECTION 1.0 GENERAL	1-1
1.1 Purpose	1-1
1.2 Project References	1-1
1.3 Terms and Abbreviations	1-2
SECTION 2.0 DEVELOPMENT ACTIVITY	2-1
2.1 Statement of Pretest Activity	2-1
2.2 Pretest Activity Results	2-1
SECTION 3.0 SYSTEM DESCRIPTION	3-1
3.1 System Description	3-1
3.2 Testing Schedule	3-3
3.3 First Location Testing	3-3
3.4 Subsequent Location Testing	3-4
SECTION 4.0 SPECIFICATIONS AND EVALUATIONS	4-1
4.1 Test Specifications	4-1
4.2 Test Methods and Constraints	4-2
4.3 Test Progression	4-2
4.4 Test Evaluation	4-2
SECTION 5.0 TEST PROCEDURES	5-1
5.1 Test Description	5-1
5.2 Test Control	5-1
5.3 Test Procedures	5-1
5.3.1 Access to IISS	5-1
5.3.2 Choosing the FLAN Function	5-3
5.3.3 Standalone Version of FLAN	5-6

APPENDICES

A	FLAN1.FDL	A-1
B	FLAN2.FDL	B-1

FIGURES

3.1	FLAN Interfaces	3-2
5.1	IISS Logon Screen	5-2
5.2	IISS Function Screen	5-3
5.3	FLAN Screen	5-4
5.4	ARTEST Screen	5-5
5.5	FLAN1.FDL/TESTFORM Screen	5-6

SECTION 1

GENERAL

1.1 Purpose

This unit test plan establishes the methodology and procedures used to adequately test the capabilities of the computer programs identified as the Forms Definition Language Compiler known in this document as FLAN, the Reverse FLAN Compiler known as REVFLAN, and Make Includes known as MAKINC. FLAN, REVFLAN and MAKINC are configuration items of the Integrated Information Support System (IISS) User Interface (UI).

1.2 Project References

- [1] ICAM Documentation Standards, 15 September 1983, IDS15012000C.
- [2] IISS Integration Task Force, Final Report, 1984.
- [3] General Electric Company, System Design Specification, 7 February 1983.
- [4] Structural Dynamics Research Corporation, Forms Language Compiler Development Specification, DS 620144401B, 1 November 1985.
- [5] Structural Dynamics Research Corporation, Forms Driven Form Editor Unit Test Plan, UTP620144402, 1 November 1985.
- [6] Structural Dynamics Research Corporation, Report Writer Unit Test Plan, UTP620144501, 1 November 1985.
- [7] Structural Dynamics Research Corporation, Rapid Application Generator Unit Test Plan, UTP620144502, 1 November 1985.
- [8] Structural Dynamics Research Corporation, Text Editor Unit Test Plan, UTP620144600, 1 November 1985.
- [9] Structural Dynamics Research Corporation, Application Interface Unit Test Plan, UTP620144700, 1 November 1985.

- [10] Structural Dynamics Research Corporation, User Interface Services Unit Test Plan, UTP620144100, 1 November 1985.
- [11] Structural Dynamics Research Corporation, Form Processor Unit Test Plan, UTP620144200, 1 November 1985.
- [12] Structural Dynamics Research Corporation, Virtual Terminal Interface Unit Test Plan, UTP620144300, 1 November 1985.
- [13] Systran, ICAM Documentation Standards, 15 September 1983.

1.3 Terms and Abbreviations

Application Definition Language: an extension of the Forms Definition Language that includes retrieval of database information and conditional actions. It is used to define interactive application programs.

Attribute: field characteristic such as blinking, highlighted, black, etc. and various other combinations. Background attributes are defined for forms or windows only. Foreground attributes are defined for items. Attributes may be permanent, i.e., they remain the same unless changed by the application program, or they may be temporary, i.e., they remain in effect until the window is redisplayed.

Common Data Model: (CDM), IISS subsystem that describes common data application process formats, form definitions, etc. of the IISS and includes conceptual schema, external schemas, internal schemas, and schema transformation operators.

Display List: is similar to the open list, except that it contains only those forms that have been added to the screen and are currently displayed on the screen.

Field: two-dimensional space on a terminal screen.

Form: structured view which may be imposed on windows or other forms. A form is composed of fields. These fields may be defined as forms, items, and windows.

Form Definition: (FD), forms definition language after compilation. It is read at runtime by the Form Processor.

Forms Definition Language: (FDL), the language in which electronic forms are defined.

Form Editor: (FE), subset of the IISS User Interface that is used to create definitions of forms. The FE consists of the Forms Driven Form Editor and the Forms Language Compiler.

Form Hierarchy: a graphic representation of the way in which forms, items and windows are related to their parent form.

Forms Language Compiler: (FLAN), subset of the FE that consists of a batch process that accepts a series of forms definition language statements and produces form definition files as output.

Form Processor: (FP), subset of the IISS User Interface that consists of a set of callable execution time routines available to an application program for form processing.

Integrated Information Support System: (IISS), a test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Item: non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.

Message: descriptive text which may be returned in the standard message line on the terminal screen. They are used to warn of errors or provide other user information.

Operating System: (OS), software supplied with a computer which allows it to supervise its own operations and manage access to hardware facilities such as memory and peripherals.

Page: instance of forms in windows that are created whenever a form is added to a window.

Paging and Scrolling: a method which allows a form to contain more data than can be displayed with provisions for viewing any portion of the data buffer.

UTP620144401
1 November 1985

Qualified Name: the name of a form, item or window preceded by the hierarchy path so that it is uniquely identified.

Subform: a form that is used within another form.

User Interface: (UI), IISS subsystem that controls the user's terminal and interfaces with the rest of the system. The UI consists of two major subsystems: the User Interface Development System (UIDS) and the User Interface Management System (UIMS).

User Interface Development System: (UIDS), collection of IISS User Interface subsystems that are used by applications programmers as they develop IISS applications. The UIDS includes the Form Editor and the Application Generator.

Window: dynamic area of a terminal screen on which predefined forms may be placed at run time.

SECTION 2

DEVELOPMENT ACTIVITY

2.1 Statement of Pretest Activity

During system development, the computer programs were tested progressively. Functionality was incrementally tested and as bugs were discovered by this testing, the software was corrected.

Several existing forms which had been created using the DIGITAL EQUIPMENT CORPORATION utility FMS were written in the forms definition language. A form was compiled and the result was displayed and visually compared with the existing form. This testing was conducted by the individual program developer in a manual mode. Any errors were noted by the developer and corrections to the program were then made after a testing session.

2.2 Pretest Activity Results

Testing of the forms discovered a few minor bugs which were then corrected and retesting proved successful. Testing included exceptional conditions and error conditions for the language. The overall test results during development showed no major programming errors. Only minor bugs were discovered and corrected.

SECTION 3

SYSTEM DESCRIPTION

3.1 System Description

FLAN is a compiler which translates Form Definition Language source files into binary Form Definition File format. The binary Form Definition Files are then used as input by the Form Processor (another configuration item of the IISS UI) for display and entry of data under the control of other application programs.

The format of the binary Form Definition Files produced by FLAN is constrained to agree with the format expected by the Form Processor configuration item.

The syntax of the Form Definition Language accepted as input is described in the Forms Language Compiler Development Specification.

The interface block diagram for FLAN is shown in figure 3-1. The top box represents the file MYFORMS which is input to the FLAN compiler (second box). FLAN produces a Form Definition object file (FD) for each CREATE FORM statement in the source file. Each FD file is input for the Form Processor which is part of the User Interface system. The compilation of an FDL file which results in an FD file is the same as program language compilation. The FDL file is the source; the FD file is the object.

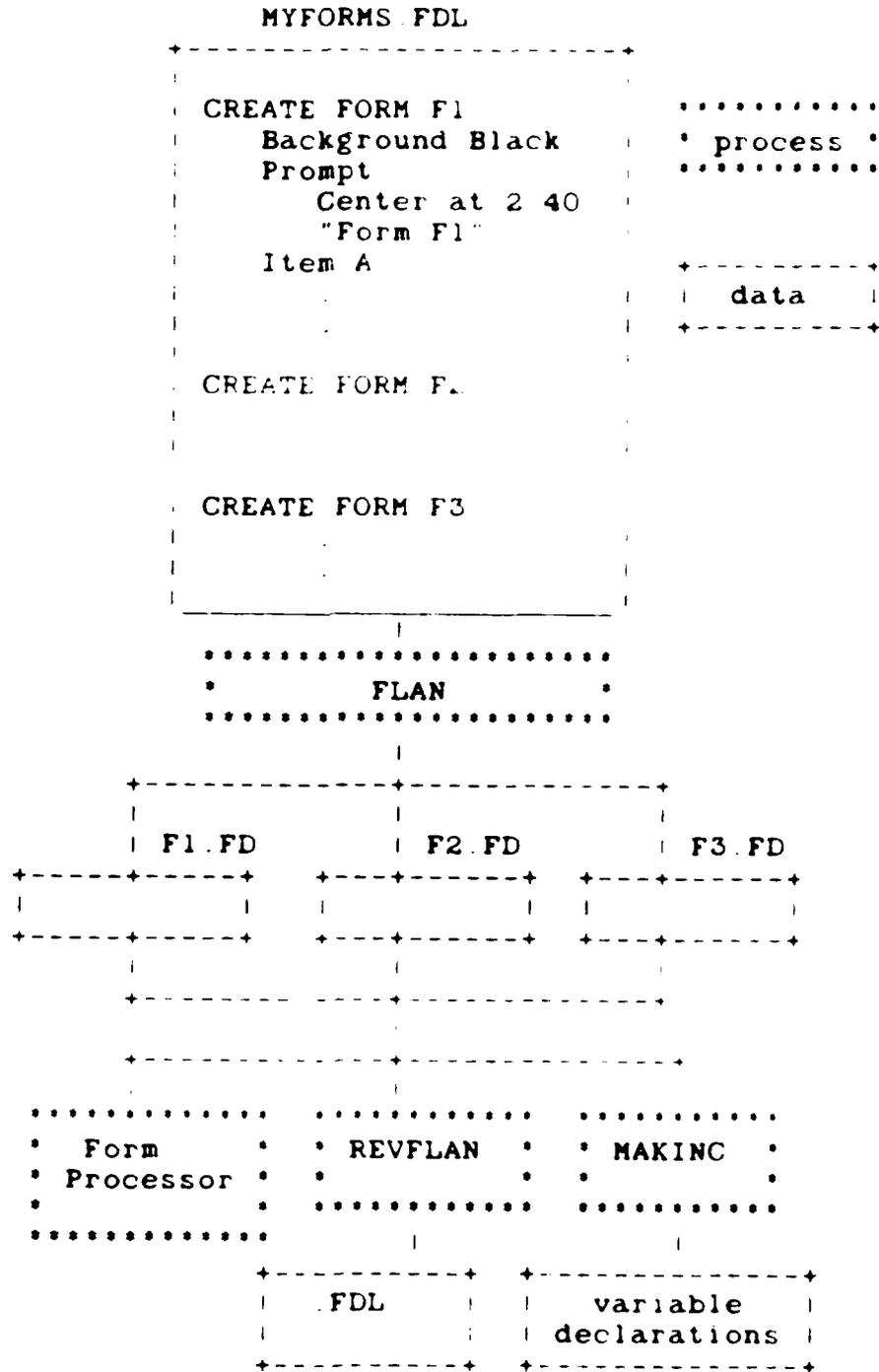


Figure 3-1 FLAN Interfaces

While FLAN is normally invoked from the IISS function screen another version is available which can be invoked from the host system. This second version is required so current configuration management software can be used in managing FDL files in a manner similar to other source files.

In order to ease the conversion of forms which were not created using the Forms Definition Language, REVFLAN is used. REVFLAN is a program used to create an FDL source file from one or more version 1.0 FD files which were created using the DEC FMS. The resulting FDL file may then be compiled using FLAN to produce version 2.0 FD files. REVFLAN is invoked from the host system.

MAKINC is a program that creates program variable declarations which correspond to the structure of a form and may be used in application programs which make use of the Form Processor calls PDATA and GDATA. The following programming languages are supported: PL/I, COBOL, and C. MAKINC is invoked from the host system.

3.2 Testing Schedule

The execution of FLAN is dependent upon the NTM subsystem of IISS and testing of FLAN must be done only after the NTM has been successfully tested. Within the UI subsystem, FLAN uses the Forms Processor and must be tested only after its successful test.

3.3 First Location Testing

These tests of FLAN require the following:

Equipment: Air Force VAX

Support Software: The Integrated Information Support System, the ORACLE database management system, a C compiler and the UI VTI subsystem.

Personnel: One integrator familiar with the IISS FLAN.

Training: FLAN training and manuals have been previously provided with all past releases.

Deliverables: The Forms Language Compiler subsystem of the IISS UI VTI.

Test Materials: This test is interactive and can be manually performed as outlined in this test plan. It also could be run as a script file if so desired. No script file has been provided because it is believed that on first testing it should be observed and then may be run again to create a script file for later testing reruns.

Security considerations: None.

3.4 Subsequent Location Testing

The requirements as listed above need to be met; however, in subsequent testing it may be advantageous to create a script file of the outlined tests and run this saving the output of the test for future comparisons.

A script file, FLANUTP.SCP and the saved output, FLANUTP.SAV exist under IISS Configuration Management. This script may be generated by typing RUN VT100 and at the args prompt "-RFLANUTP.SCP -SFLANUTP.SAV". To compare results with those obtained by during successive executions, use the command file Diffile.com (found under Configuration Management). The only differences should be the date and time stamp on the IISS function screen.

SECTION 4
SPECIFICATIONS AND EVALUATIONS

4.1 Test Specification

The following requirements are demonstrated by the outlined tests:

Functional Requirements	Test Activity												
	A	B	C	D	E	F	G	H	I	J	K	L	
Specification of forms:													
background attributes		*											
form prompts		*											
size		*											
fields		*											
Specification of fields:													
type of field				*									
arrays					*								
location						*							
size						*							
display attributes							*						
field prompts								*					
domain (item only)									*				
help(message and form)										*			
value (item only)											*		
generate form descriptions												*	
semantic error messages													*

- A - input of forms fat1 and fat2.
- B - input of form testform.
- C - input of field types: items, windows and forms.
- D - input of item field i4.
- E - input of all fields.
- F - input of all items and windows.
- G - input of items i1, i5, i6, i7, i8, i9, i10, i10, window w1, form fat1.

H input of items 15, 16, 17, 18.
I input of items 19, 110
J input of items 11, 12, 13
K form descriptions used by form processor.
L input of file FLAN2.FDL

The steps outlined in Section 5.3 and the files in appendices A and B show the direct correspondence between the test and the functional requirements as listed in this section.

4.2 Testing Methods and Constraints:

The tests as outlined in Section 5.3 must be followed. The required input is stated for each test. This testing tests the normal mode of operation of these functions and does not completely exercise all the error combinations that a user of the FLAN might create by faulty entry of field information. These tests have been done, however, through the normal testing done by the developer of these functions. No data recording is required. No additional constraints are placed on this unit test besides those listed in Section 5.3 of this unit test plan.

4.3 Test Progression

The progression of testing of the FLAN is fully outlined in Section 5.3 of this unit test plan. This progression should be followed exactly to insure the successful testing of this IISS configuration item.

4.4 Test Evaluation

There are several stages in the testing of FLAN.

Stage 1: Input the file FLAN1.FDL TO FLAN. This will produce the FL files TESTFORM, IIOHELP, FAT1 and FAT2 in the NTE directory.

Stage 2: Run SDARTESTZZ from the IISS function screen and add the form TESTFORM to screen. This will produce a screen like the one in figure 5-5.

Stage 3: Input the file FLAN2.FDL to FLAN. This will produce the error messages listed in section 5.3.3.

SECTION 5
TEST PROCEDURES

5.1 Test Description

A general description of this unit test was provided in Section 3.

5.2 Test Control

As outlined, this unit test is a manual test which may be done by anyone. The required input data are documented for each function being tested and the resulting successful output is also documented. The order of the testing is also completely documented. The test control information is completely described in Section 5.3. Verification of the test is by a manual comparison of the test output with the expected results as they are documented here.

5.3 Test Procedures

To run the unit test plan, one must be logged on to an IISS account. The NTM must be up and running and the UI group logical names IISSFLIB, IISSULIB and IISSMLIB must be set properly. IISSFLIB points to the directory containing form definitions (.FD files). Therefore, IISSULIB should point to the NTM environment directory so that when SDARTEST is subsequently executed it may find the .fd files. IISSMLIB points to the directory containing error messages (.MSG files). The FLAN test must be started as follows.

5.3.1 Access to IISS

To log on the IISS, the following form must be filled in

USER ID: _____
PASSWORD: _____
ROLE: _____

Msg: 0 application

Figure 5-1 IISS Logon Screen

- (1) USER ID is the identification name of the user, and is 1 to 10 alpha-numeric characters. USER ID is input as "MORENC".
- (2) PASSWORD must be the password associated with the USER ID, and is 1 to 10 alpha-numeric characters. PASSWORD is input as "STANLEY".
- (3) ROLE is any of the identifiers which are associated with the USER ID, and is 1 to 10 alpha-numeric characters. It will be checked against functions and applications which are selected by the user. ROLE is input as "MANAGER".

When this form is correctly completed and the ENTER key is pressed, the form in figure 5-2 is displayed.

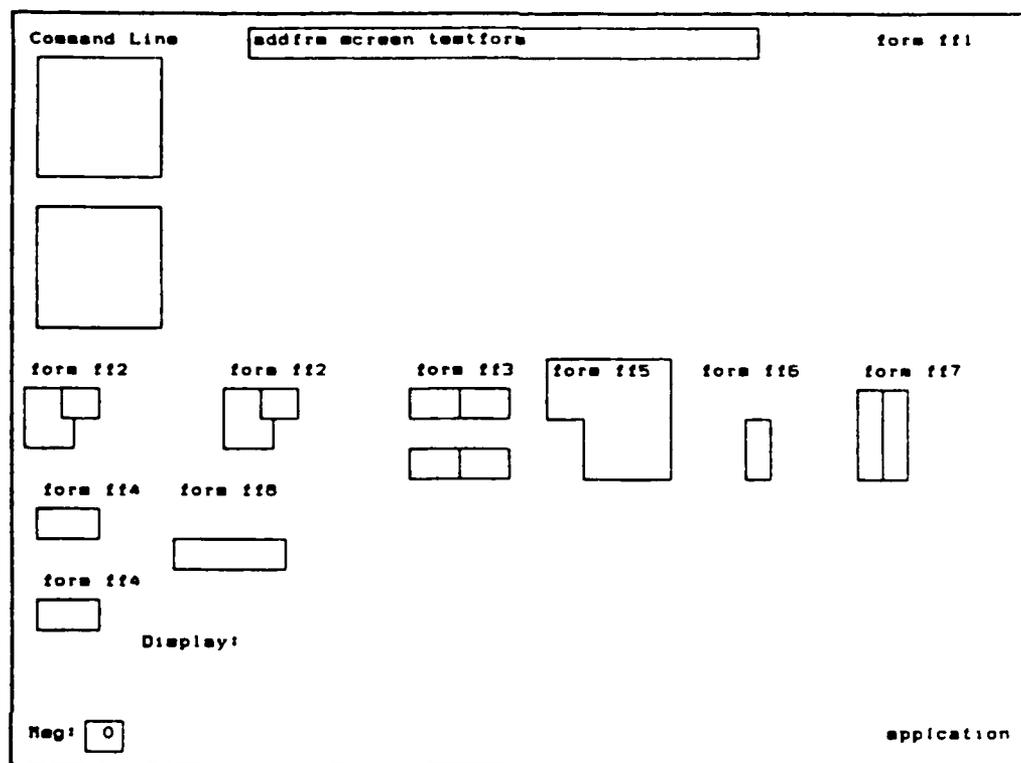


Figure 5-4 ARTEST Screen

In the command line type "ADDFRM SCREEN TESTFORM" and press <enter>. Compare the appearance of the screen with the figure 5-5.

size/display domains		help	
<input type="checkbox"/> 1	<input type="checkbox"/> (must enter must fill numeric)	<input type="checkbox"/> message	
22	<input type="checkbox"/> (left lower)	<input type="checkbox"/> form	
22	<input type="checkbox"/> (right upper)		
333	<input type="checkbox"/> (max 10 min 0)		
333	<input type="checkbox"/> (max 10 min 0)		
	form windows		location
array	<input type="checkbox"/> fat1 <input type="checkbox"/>		12 13
	fat2		11 12 13 14
			11 14
			15 1b
			15 1b
			16 1c
			16 1c
			17 1a
			17 18 19 1a
			18 19
neg: <input type="checkbox"/> 0			
			application

Figure 5-5 FLAN1.FDL/TESTFORM Screen

When finished viewing, press <quit> and when the function screen is displayed, press <quit>.

5.3.3 Standalone Version of FLAN

The standalone version of FLAN is invoked by typing "run [flandir]flansa.exe" where [flandir] is the IISS production directory containing the FLAN executable. When the "args:" prompt appears type "[formdir]flan2.fdl" where [formdir] is the directory containing the FLAN2.FDL file. The messages printed should be identical to the following list.

- 6: ERROR - must specify relative field name 13: ERROR
- size not specified or invalid
- 18: ERROR - value too big for field
- 22: ERROR - no display attribute specified
- 64: ERROR - unterminated string

66: ERROR - value too big for field
70: WARNING - string too long
72: ERROR - duplicate field name: J
81: ERROR - duplicate display attribute specified
86: ERROR - unknown display attribute: UGLY
88: ERROR - no display attribute specified
92: ERROR - domain only legal for items
98: ERROR - duplicate justification specified
104: ERROR - duplicate case specified
110: ERROR - duplicate minimum specified
116: ERROR - duplicate maximum specified
122: ERROR - help only legal for items
122: ERROR - field NOTHING referenced in item BB not defined
122: ERROR - item HH off left of screen
122: ERROR - item G off top of screen
122: ERROR - form TESTERR prompt off top of screen
122: ERROR - form TESTERR prompt off left of screen
122: ERROR - circular reference in location of item DC
122: ERROR - circular reference in location of item CD
122: ERROR - overlap between item A and item CC
122: ERROR - overlap between item A and item E
122: ERROR - overlap between item A and item F
122: ERROR - overlap between item E and item F
122: WARNING - form TESTERR too wide for standard screen
122: WARNING - form TESTERR too long for standard screen
122: ERROR - form TESTERR too narrow: fields extend to
column 157
122: ERROR - form TESTERR too short: fields extend to row 25
130: ERROR - duplicate help specified 136: WARNING - help
message too long, truncated
142: ERROR - value only legal for items
156: ERROR - unknown function FUNC
162: ERROR - invalid argument for INDEX
168: ERROR - duplicate size specified
173: ERROR - unterminated comment Unable to continue...

APPENDIX A

FLAN1.FDL

FLAN input file with correct syntax and semantics to test all features.

```
/* test forms for the flan compiler */

create form testform
  prompt center at 2 below i3 and column 6 "array" /* form
prompt */

item i1                                /* item field */
  size 1 at 2 3                        /* field size */
  display as input                     /* field display attribute */
  value "1"                             /* item value */
  prompt at 1 2 "size/display" /* field prompt */

item i2
  size 2 by 2
  at below i1
  display as output
  value "2222"

item i3
  size 3 by 2
  at below i2
  display as text
  value "333333"

item i4 (2 v, 3 h 2, 2 v 2)           /* array */
  at 3 below i3 and column 2
  size 1
  display as input

form fat1                              /* forms */
  at 3 below i3 and column 12
  size 6
  prompt at above "forms"

form fat2
  at below fat1
  size 6
```

```
window w1                                /* windows */
  at 3 right of fat1
  size 5
  background white
  prompt at above "windows"

window w2
  at below w1
  size 5
  background black

item i5                                    /* domains */
  at 2 15
  size 2
  display as input
  domain (must enter must fill numeric)
  prompt at right "(must enter must fill numeric)"
  prompt at 1 above and col 15 "domains"

item i6
  at below i5
  size 4
  display as input
  domain (left lower)
  prompt at right "(left lower)"

item i7
  at below i6
  size 4
  display as input
  domain (right upper)
  prompt at right "(right upper)"

item i8
  at below i7
  size 4
  display as input
  domain (max 10 min 0)
  prompt at right "(max 10 min 0)"
```

```
item i9                                /* help */
  at 2 60
  size 1
  display as input
  help "help message for i9"
  prompt at right "message"
  prompt at 1 above "help"

item i10
  at below i9
  size 1
  display as input
  help i10help
  prompt at right "form"

item i10                                /* location tests */
  at 13 40
  size 5 by 5
  display as input
  prompt at 4 above "location"

item i11
  size 2 by 2
  display as input
  bottom right at 1 above i10 and 2 left of i10
  value "1111"

item i12
  size 2 by 2
  display as input
  bottom left at 2 above top left of i10
  value "1212"

item i13
  size 2 by 2
  display as input
  bottom right at above top right of i10
  value "1313"

item i14
  size 2 by 2
  display as input
  bottom left at 1 above i10 and 2 right of i10
  value "1414"
```

- item 15
size 2 by 2
display as input
top right at left of top left of 10
value"1515"
- item 16
size 2 by 2
display as input
bottom right at 2 left of bottom left of 10
value"1616"
- item 17
size 2 by 2
display as input
top right at 1 below 10 and 2 left of 10
value"1717"
- item 18
size 2 by 2
display as input
top left at 1 below bottom left of 10
value"1818"
- item 19
size 2 by 2
display as input
top right at 2 below bottom right of 10
value"1919"
- item 1a
size 2 by 2
display as input
top left at 1 below 10 and 1 right of 10
value"1a1a"
- item 1b
size 2 by 2
display as input
top left at 2 right of top right of 10
value"1b1b"

UTP620144401
1 November 1985

```
item 10  
  size 2 by 2  
  display as input  
  bottom left at right of bottom right of 10  
  value 1000  
  
create form 110help  
  size 80 by 23  
  prompt center at 10 40 "help form form item 110"  
  
create form fat1                                " form background and size "  
  background white  
  size 5  
  prompt at 1 2 "fat1"  
  
create form fat2  
  background black  
  size 5  
  prompt at 1 2 "fat2"
```

APPENDIX B

FLAN2.FDL

FLAN input file to test all semantic error messages.

```
* flan forms to force all semantic error messages */

create form testerr
size 1 /*("form %s too narrow: fields extend to column %d",
  ("form %s too short: fields extend to row %d", */
prompt at left "testerr" /*("must specify relative field
name "
prompt at 25 2 "off bottom"

item a /*("size not specified or invalid");*/
at 1 2
display as input

item b /*("value too big for field");*/
size 1
value "22"
display as input

item cc /*("no display attribute specified");*/
size 1
at 1 4

item bb /*("field %s referenced in %s %s%s not defined",*/
at below nothing
size 1
display as input

item cd /*("circular reference in location of %s %s%s",*/
at below d
size 1
display as input

item dc
at above cd
size 1
display as input

item e /*("overlap between %s %s%s and %s %s%s",*/
at 2 2
size 1
display as input
```

item f
at 2 2
size 1
display as input

item g /*("%s %s off top of screen",*/
at -1 10
size 1
display as input

item hh /*("%s %s off left of screen", */
at 1 -1
size 1
display as input

item i /*("unterminated string");*/
at 1 6
size 1
display as input
value "hello

item j /*("string too long");*/
at 1 8
size 150
display as input
value
"12345678911234567892123456789312345678941234567895123456789612
345 67897123
456789812345678991234567890123456789112345678921234567893123456
7894"

item j /*("duplicate field name: %s", */
at 3 2
size 1
display as input

item k /*("duplicate display attribute specified"),*/
at 3 4
size 1
display as input
display as input

item l /*("unknown display attribute: %s", */
at 3 6
size 1
display as ugly

UTP620144401
1 November 1985

```
window m at 3 8  
size 1  
display as black  
domain (upper) /*("domain only legal for items");*/
```

```
item n /*("duplicate justification specified");*/  
at 3 10  
size 1  
display as input  
domain (left right)
```

```
item o /*("duplicate case specified");*/  
at 3 12  
size 1  
display as input  
domain (upper lower)
```

```
item p /*("duplicate minimum specified");*/  
at 3 14  
size 1  
display as input  
domain (min 10 min 2)
```

```
item q /*("duplicate maximum specified");*/  
at 3 16  
size 1  
display as input  
domain (max 10 max 2)
```

```
window rr /*("help only legal for items");*/  
at 3 18  
size 1  
display as black  
help "hello"
```

```
create form testform  
item s /*("duplicate help specified");*/  
at 3 20  
size 1  
display as input  
help "hello"  
help "hello"
```

```
item t ("help message too long, truncated");  
at 3 2.  
size 1  
display as input  
help  
"12345678911234567892123456789312345678941234567895123456789612  
34567897"
```

```
window u ("value only legal for items");  
at 3 24  
size 1  
display as blank  
value "hello"
```

```
create form testform  
item vv ("duplicate value specified");  
at 3 26  
size 10  
display as input  
value "hello"  
value "hello"
```

```
item w ("unknown function %s", %s);  
at 3 38  
size 20  
display as input  
value func( hello )
```

```
item x ("invalid argument for INDEX");  
at 3 60  
size 20  
display as input  
value index 10
```

```
item y ("duplicate size specified");  
at 4 1  
size 1  
size 1  
display as input
```

```
create form endless  
  else if (c == EOF) {fatal("unterminated comment"); return
```

END

8-87

DTIC