(1)

WRDC-TR-90-8007
Volume VIII
Part 40

WRIGHT RESEARCH & DEVELOPMENT CENTER

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 40 - SGML Tagger Unit Test Plan

S. Barker, F. Glandorf

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

[illegible] 25 July 91

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

DATE

FOR THE COMMANDER:

Bruce Rasmussen 25 July 91

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

DATE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
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) UTP620344901	5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR- 90-8007 Vol. VIII, Part 40
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services 6b. OFFICE SYMBOL *(if applicable)*	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI
6c. ADDRESS *(City,State, and ZIP Code)* 2970 Presidential Drive Fairborn, OH 45324-6209	7b. ADDRESS *(City, State, and ZIP Code)* WPAFB, OH 45433-6533
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF 8b. OFFICE SYMBOL *(if applicable)* WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464
8c. ADDRESS *(City, State, and ZIP Code)* Wright-Patterson AFB, Ohio 45433-6533	10. SOURCE OF FUNDING NOS.
11. TITLE SGML T See block 19	PROGRAM ELEMENT NO. 78011F · PROJECT NO. 595600 · TASK NO. F95600 · WORK UNIT NO. 20950607

12. PERSONAL AUTHOR(S)
Structural Dynamics Research Corporation: Barker, S. , Glandorf, F., et al.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Yr.,Mo.,Day)*	15. PAGE COUNT
Final Report	4/1/87-12/31/90	1990 September 30	55

16. SUPPLEMENTARY NOTATION

WRDC/MTI Project Priority 6203

17. COSATI CODES

FIELD	GROUP	SUB GR.
1308	0905	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify block no.)*

19. ABSTRACT *(Continue on reverse if necessary and identify block number)*

This unit test plan establishes the methodology and procedure to be used to test the capabilities Electronic Documentation System (EDS) SGML Tagger computer program.

BLOCK 11.

INTEGRATED INFORMATION SUPPORT SYSTEM
Vol VIII -User Interface Subsystem

Part 40 - SGML Tagger Unit Test Plan

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NO. *(Include Area Code)* (513) 255-7371 · 22c. OFFICE SYMBOL WRDC/MTI

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface,and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION 1

GENERAL

1.1 Purpose

This unit test plan establishes the methodology and procedures used to adequately test the capabilities of the computer program identified as the SGML Tagger known in this document as the SGML Tagger. The SGML Tagger is one configuration item of the Integrated Information Support System (IISS) User Interface (UI).

1.2 Project References

[1] Systran, ICAM Documentation Standards, IDS150120000C, 15 September 1983.

[2] International Organization for Standardization, Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML), ISO 8879, 15 October 1986.

[3] International Organization for Standardization, Office Document Architecture/Office Document Interchange Format, ISO/DP 8613/1-6, October 1985 (Draft).

[4] American National Standards Institute, American National Standard for Information Systems - Computer Graphics - Metafile for the Storage and Transfer of Picture Description Information, ANSI X/3.122-1986, August 27, 1986.

[5] Structural Dynamics Research Corporation, Form Processor User's Manual, UM 620244200A, 16 February 1987.

[6] Structural Dynamics Research Corporation, Virtual Terminal Operator Guide, OM 620244000A, 16 February 1987.

[7] M.E. Lesk, LEX - Lexical Analyzer Generator, IS Workbench for VAX/VMS Programmers Guide.

[8] Structural Dynamics Research Corporation, Form Processor Development Specification, DS 620244700A, 16 February 1987.

[9] Structural Dynamics Research Corporation, Electronic Documentation System Development Specification, DS 620344900, 15 December 1987.

1.3 Terms and Abbreviations

American Standard Code for Information Interchange (ASCII): The character set defined by ANSI x3.4 and used by most computer vendors.

Attribute: A characteristic used to qualify an element within a document.

Backus-Naur Form (BNF): A widely used notation form specifying the syntax of a language.

Character Set: A mapping of a character repertoire onto a code set such that each character is associated with its coded representation.

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

Compound Document: A document which may contain mixed content (text, graphics, etc.).

Computer Graphics Metafile (CGM): A standard file format for the storage and retrieval of picture description information.

Computer Program Configuration Item (CPCI): An aggregation of computer programs or any of their discrete portions, which satisfies an end-use function.

Conforming SGML Application: An SGML application that requires documents to be conforming SGML documents, and whose documentation meets the requirements of this International Standard.

Conforming SGML Document: AN SGML document that complies with all provisions of this International Standard.

Context-Directed Editor: An EDS application which guides the user through the process of document creation and revision by using the document type definition as a model for which logical elements may be included in the document.

Cursor Position: The position of the cursor after any command is issued.

Descriptive Markup: Information added to a document that enables an application program to process the document.

Device Driver (DD): Software modules written to handle I/O for a specific kind of terminal. The modules map terminal-specific commands and data to a neutral format. Device Drivers are part of the UI Virtual Terminal.

Document Type Definition (DTD): Rules determined by an application that apply SGML to the markup of documents of a particular type. A document type definition includes a formal specification, expressed in a document type declaration, of the element types, element relationships and attributes, and references that can be represented by markup. It thereby defines the vocabulary of the markup for which SGML defines the syntax. A DTD can also include comments that describe the semantics of elements and attributes, and any application conventions.

Electronic Documentation System (EDS): An integrated set of software tools and application programs which operate upon a document through various stages of a document life cycle consisting of editing (creating/revising), formatting, imaging, storage, and transfering.

Element: A component of the hierarchical structure defined by a document type definition; it is identified in a document instance by descriptive markup, usually a start-tag and end-tag.

Element Declaration: A markup declaration that contains the formal specification of the part of an element type definition that deals with the content and markup minimization.

Entity: A collection of characters that can be referenced as a unit.

Field: Two-dimensional space on a terminal screen.

Form: A 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, or windows.

Form Definition (FD): Form definition Language after compilation. It is read at run-time by the Form Processor.

Form Definition Language (FDL): The language in which electronic forms are defined.

Form Editor (FE): A 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.

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

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

Forms Driven Form Editor (FDFE): A subset of the FE which consists of a forms-driven application used to create Form Definition files interactively.

Imaging: The printing of a document.

IISS Function Screen: The first screen that is displayed after logon. It allows the user to specify the function to access and the device type and device name on which to work.

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: A 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.

Layout Style: The specification of format and presentation for logical elements.

Layout Structure: The hierarchy of all layout elements (pages, frames, blocks, etc.) for a document.

Logical Structure: The hierarchy of all logical elements (paragraphs, sections, etc.) within a document.

Look-Ahead LR (LALR): An effecient, bottom-up parsing technique.

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

Message Line: A line on the terminal screen that is used to display messages.

Neutral Data Definition Language (NDDL): A language used to manipulate and populate information in the Common Data Model (CDM) or IISS System Database.

Neutral Data Manipulation Language (NDML): A language developed by the IISS project to provide uniform access to common data, regardless of data base manager or distribution criteria. It provides distributed retrieval and single node update.

Office Document Architecture (ODA): A standard which supports the interchange of electronic compound documents in such a way as to allow their imaging, processing, or reformatting.

Office Document Interchange Format (ODIF): A standard for encoding document structures defined by the ODA which would enable the exchange of compound documents between systems operating within a MAP/TOP environment.

Operating System (OS): Software supplied with a computer which allows it to supervize 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 at one time with provisions for viewing any portion of the data buffer.

Parser: An application program that determines how closely a document conforms to a document type definition which defines a specific documentation standard.

Physical Device: A hardware terminal.

Previous Cursor Position: The position of the cursor when the previous edit command was issued.

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

Standard Generalized Markup Language (SGML): A language for describing document structures, consisting of descriptive markup which is added to a document to indicate where logical elements such as sections and paragraphs begin and end.

Subform: A form that is used within another form.

Tag: Descriptive markup indicating the start or end of a logical element.

Tagger: An application program which provides a mechanism for automatically tagging existing documents which have been created by word processing systems.

User Data: Data which is either input by the user or output by the application programs to items.

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 Management System (UIMS): The run-time UI. It consists of the Form Processor, Virtual Terminal, Application Interface, the User Interface Services, and the Text Editor.

User Interface Services (UIS): A subset of the IISS User Interface that consists of a package of routines that aid users in controlling their environment. It includes message management, change password, and application definition services.

User Interface/Virtual Terminal Interface (UI/VTI): Another name for the User Interface.

Virtual Terminal (VT): A subset of the IISS User Interface that performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by the UI software which constitutes the virtual terminal definition. Specific terminals are then mapped against the virtual terminal software by specific software modules written for each type of real terminal supported.

Virtual Terminal Interface (VTI): The callable interface to the VT.

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

Window Manager: A facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor.

Yet Another Compiler Compiler (YACC): An LALR parser generator.

SECTION 2

DEVELOPMENT ACTIVITY

2.1 Statement of Pretest Activity

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

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 application discovered a few minor bugs which were then corrected and retesting proved successful. Testing included exceptional conditions and error conditions for editing. 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

The Electronic Documentation System provides an integrated set of tools that enable the creation and revision of compound documents with a well-defined logical and layout structure.

EDS operates in the IISS environment on those hosts which support IISS.

It is capable of accepting input from and producing output for those devices which are supported within the IISS environment.

EDS interfaces with other IISS application programs by accepting graphical and textual input from these programs for inclusion into EDS documents. EDS includes application programs (software modules) which translate external graphical and textual information into the standard file formats used by EDS. The file formats used by EDS are based on current standards designated for document processing and exchange, as well as device-independent storage of graphics files.

EDS application programs which require user input use the User Interface Form Processor as a mechanism for accepting and validating user input.

As shown in Figure 3-1, in order to process various text and graphic input/output from multiple external sources, EDS must provide interfaces to the external environment at multiple levels.

```
+------------------+                          +----------------+
|  External Word   |                          |    External    |
| Processor Files  |                          | Graphics Files |
+--------+---------+                          +----------+-----+
  +------+----+         +---------------+                |
  | DOCUMENT  |         | +-----------+ |          +-----+----+
  |  TAGGER   |         | |           | |          | Graphics |
  +------+----+         | +-----------+ |          |   File   |
        |             +-+---------------+-+        |Translator|
        |  SGML/ODA   +----+--------+-----+        +-----+----+
        |              User|        ^ EDS                |
        |   +--------------+--------+---------------+    |
        |   |    Electronic Documentation System    |    |
        |   +-------------+---------------+---------+    |
        +-->| DTD Builder | File Collector|  Editor |    |
            +-------------+---------------+---------+    | CGM
            | Layout Macro| NDML Processor|Formatter|<---+
      +---->+-------------+---------------+---------+
      |     |             |               |         |
      |     +-------------+------+--------+---------+
      |                          | Virtual Terminal
      |                          | Graphics Protocol
      |               +----------+--------+
      |               | Virtual Terminal  |
      |               +----------+--------+
      |  +--------------+        +-+
      +--+ Common Data  |          |
         |    Model     |          |
         +--------------++---------+-------+
                         | Device Driver   |
                         +---+-----+-------+
             +---------------+     |
        +----+-----+      +--------+-------+
        +----------+      | +------------+ |
        +----------+      | |            | |
        +----------+      | +------------+ |
                        +-+----------------+--+
         <Paper>        +---------------------+
                             <Display>
```

Figure 3-1 EDS Interface Diagram

3.2 Testing Schedule

The execution of the SGML Tagger is dependent upon the User Interface Form Processor.

3.3 First Location Testing

These tests of the SGML Tagger require the following:

Equipment: Air Force VAX, terminals supported by the virtual terminal as listed in the UI Terminal Operator's Guide.

Support Software: The Integrated Information Support System, LEX (lexical analyzer generator), and the VAX11 C compiler.

Personnel: One integrator familiar with the IISS.

Training: The EDS User Manual has been provided with the current release.

Deliverables: The EDS subsystem of the IISS UI/VTI.

Test Materials: This test is interactive and can be manually performed as outlined in this test plan. It uses the file DECDX.DAT as input to the Autotag utility and the file DXUTP.DX as input to the generated tagger.

Security considerations: None.

3.4 Subsequent Location Testing

The requirements as listed above need to be met.

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
User C Declarations	*				
Character Class Definitions		*			
State Declarations			*		
Pattern Declarations				*	
Action Declarations					*

A - Steps 27-31 of section 5.3.1.
B - Steps 11-15 of section 5.3.1.
C - Steps 6-10 of section 5.3.1.
D - Steps 16-26 of section 5.3.1.
E - Step 19 of section 5.3.1.

The steps outlined in section 5.3 show the correspondence between the test and the functional requirements as listed in this section. These functional requirements match those as specified in the EDS Development Specification.

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 SGML Tagger might create by faulty entry of form field information. These tests have been done, however, through the normal testing done by the developer of these functions. No additional constraints are placed on this unit test besides those listed in section 3.3 of this unit test plan.

4.3 Test Progression

The progression of testing of the SGML Tagger 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

The test results are evaluated by comparing the information returned on the various output screens to that specified as successful for the given test. As outlined in section 5.3, each test of SGML Tagger functionality provides an input screen with the required data entry specified and the resulting output for a successful test. Next the generated tagger must be successfully compiled and linked. When run with the test input file, the file DXUTP.TAG listed in appendix D will result.

SECTION 5

TEST PROCEDURES

5.1 Test Description

This test consists of creating a tagger for DX type documents (output from WPS-Plus) using the EDS Autotag utility on a supplied template file for DX documents. You then use the Unix utility Lex to generate the DX document tagger and tag a supplied WPS output document using the generated tagger.

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. Accurate observation of the resulting successful output must be made to ensure the unit test was done properly.

5.3 Test Procedures

Below is an example of how the SGML Tagger may be invoked in the VAX/VMS environment. To run the unit test plan as outlined: one must be logged on to an IISS account. The NTM must be up and running and the UI symbolic names IISSFLIB, IISSULIB and IISSMLIB must be set properly. IISSFLIB points to the location containing form definitions (FD files). IISSULIB is set to the NTM environment directory. IISSMLIB points to the location containing error messages (MSG files).

5.3.1 DEC DX Documents

The unit test is divided into two stages. In the first stage an SGML tagger specification is created using the EDS utility Autotag. In the second stage a program is generated which tags documents from the tagger specification. For the first stage the NTM must be up and running and the UI symbolic names IISSFLIB, IISSULIB and IISSMLIB must be set properly. This test also uses the files DECDX.DAT, DXUTP.DX, DXUTP.SAV, and DXUTP.SCP. These files are in IISS Configuration Management and must be copied to the NTM environment directory. Note that upper and lower case differences are important.

Stage 1-

1	$ SET DEFAULT <to directory containing your NTM environment>	If scripting is available steps 2 through 32 may be replaced with $ VT100 -RDXUTP.SCP.
2	$ VT100	This starts the VT100 device driver. If the User Interface has been installed at your site with a different device driver, then this step is amended as is appropriate.
3		Fill in the items on the IISS Logon Screen as follows:
	Username: MORENC Password: STANLEY Role : MANAGER Press <ENTER>	
4		Fill in the Function field on the IISS Function Screen as follows:
	Function: AUTOTAG Press <ENTER>	
5		The screen in figure A-1 is displayed. Fill in the field as follows:
	File: DECDX.DAT Press <ENTER>	
6	Press <pf5>	The screen in figure A-2 is displayed.
7	Press <ENTER>	The screen in figure A-3 is displayed. Fill in the field as follows:
	State: DUMMY	

	Press <ENTER>	The message "entered" is displayed.
	Press <QUIT>	The screen in figure A-2 is redisplayed with the newly entered value.
8		Repeat step 7 for the values: BODY, CHAPNUM, READNL, FIGNUM, FIGTITLE, FIGEND, and SECTITLE.
9		Place the cursor on the field with the value: DUMMY.
	Press <ENTER>	The screen in figure A-3 is displayed with the value. Change the word DUMMY to DUMMYX.
	Press <ENTER>	The message "entered" is displayed.
	Press <QUIT>	The screen in figure A-2 is displayed with the newly entered value.
10		Place the cursor on the field with the value: DUMMYX.
	Press <ENTER>	The screen in figure A-3 is displayed with the value.
	Press <pf12>	The message "deleted" is displayed.
	Press <QUIT>	The screen in figure A-2 is displayed without the value.
	Press <QUIT>	Figure A-1 is redisplayed.
11	Press <pf6>	The screen in figure A-4 is displayed to test the character class entry.
12	Press <ENTER>	The screen in figure A-5 is displayed. Fill in the fields as follows:

	Character Class: dummy Regular Expression: "x" Press <ENTER>	Include the double quotes. The message "entered" is displayed.
	Press <QUIT>	The screen in figure A-4 is redisplayed.
13		Repeat step 12 for each of the following values.

```
snum     ([0-9]+("."[0-9]+)+)
dnum     ([0-9]+"-"[0-9]+)
number   ([0-9]+)
format   ({und_on}|{und_off}|{bold_on}|
          {bold_off}|{com_on}|{com_off})
```

14	Press <pf1>	This sets scroll/page mode.
	Press <pf9> Press <pf9>	This pages to the screen with the value: dummy. Place the cursor on this field.
	Press <ENTER>	The screen in figure figure A-5 is displayed with the value: dummy. Fill in the fields as follows:
	Character Class: dummyx Regular Expression: "y" Press <ENTER>	The message "entered" is displayed.
	Press <QUIT>	The screen in figure A-4 is redisplayed with the changed values.
15		Place the cursor on the field with the value: dummyx.
	Press <ENTER>	The screen in figure A-5 is displayed with the value: dummyx.
	Press <pf1> Press <pf1> Press <pf1> Press <pf1>	The application mode is active.

	Press <pf12>	The message "deleted" is displayed.
	Press <QUIT>	The screen in figure A-4 is displayed without the field with the value: dummyx.
	Press <QUIT>	The screen is figure A-1 is displayed.
16	Press <pf7>	The screen in figure A-6 is displayed.
17	Press <ENTER>	The screen in figure A-7 is displayed.
18		In the pattern field enter "DUMMY", including the double quotes.
	Press <ENTER>	The message "entered" is displayed.
19		Place the cursor on the field Action.
	Press <ENTER>	The screen in figure A-8 is displayed. In the field Action enter: /* dummy */.
	Press <ENTER>	The message "entered" is displayed.
	Press <QUIT>	The screen in figure A-7 is redisplayed with the data entered.
20		Place the cursor on the field State.
	Press <ENTER>	The screen in figure A-9 is displayed. In the field State enter: DUMMY.
	Press <ENTER>	The message "entered" is displayed.

Press <QUIT>	The screen in figure A-7 is redisplayed with the data entered.
Press <QUIT>	The screen in figure A-6 is redisplayed with the data entered.
21	Repeat steps 17-20 for the following:

```
Pattern:
{und_on}?"SECTION"{space}*{number}{und_off}?/{center}

Actions:
 /* a section header */
 EMIT("<sectnum>");
 ECHO;
 EMIT("</sectnum>");
 BEGIN SECTITLE;

State:
BODY

Pattern:
({any_char}+{center})+/{dx_nl}{dx_nl}

Actions:
 /* section name */
 EMIT("<sectitle>");
 ECHO;
 EMIT("</sectitle>");
 BEGIN READNL;

State:
SECTITLE

Pattern:
{snum}{space}*/{und_on}?{any_char}+{und_off}?{dx_nl}{dx_nl}

Actions:
 /* a numbered paragraph */
 EMIT("<chapnum>");
 ECHO;
 EMIT("</chapnum>");
 BEGIN CHAPNUM;

State:
BODY
```

```
Pattern:
{und_on}?{any_char}+{und_off}?/{dx_nl}{dx_nl}

Actions:
 /* paragraph name */
 EMIT("<chptitle>");
 ECHO;
 EMIT("</chptitle>");
 BEGIN READNL;

State:
CHAPNUM

Pattern:
({any_char}|{ww_eol}|{format})+/{dx_nl}{white_space}*{dx_nl}

Actions:
 /* a paragraph */
 EMIT("<para0>");
 ECHO;
 EMIT("</para0>");
 BEGIN READNL;

State:
BODY

Pattern:
{dx_nl}{white_space}*{dx_nl}

Actions:
 /* eat new lines after paragraphs and headers */
 EMIT("\n");
 BEGIN BODY;

State:
READNL

Pattern:
(({any_char}|{format})+{dx_nl})+{dx_nl}

Actions:
 /* a figure, maybe */
 EMIT("<figbody>");
 ECHO;
 EMIT("</figbody>");
```

```
State:
BODY

Pattern:
"Figure"{white_space}+/{dnum}{white_space}+{any_char}+{center}

Actions:
 /* figure title */
 EMIT("<figref>");
 ECHO;
 EMIT("</figref>");
 BEGIN FIGNUM;

State:
BODY

Pattern:
{dnum}{white_space}+

Actions:
 /* figure number */
 EMIT("<fignum>");
 ECHO;
 EMIT("</fignum>");
 BEGIN FIGTITLE;

State:
FIGNUM

Pattern:
{any_char}+

Actions:
 /* figure name */
 EMIT("<figtitle>");
 ECHO;
 EMIT("</figtitle>");
 BEGIN FIGEND;

State:
FIGTITLE

Pattern:
{center}({white_space}*{dx_nl})*
```

```
Actions:
 /* figure cleanup */
 EMIT("\n");
 BEGIN BODY;

State:
FIGEND

Pattern:
.

Actions:
 /* anything else */
 if (yytext[0] == '|' || yytext[0] == '{') yymore();
 else ECHO;

State:
```

22		Place cursor on Pattern "DUMMY".
	Press <ENTER>	The screen in figure A-7 will be displayed with the data. Modify the Pattern to: "DUMMYX".
	Press <ENTER>	The message "entered" will be displayed.
23		Place the cursor on the Action field with the code: /* dummy */.
	Press <ENTER>	The screen in figure A-8 will be displayed with the data. Modify the Code to: /* dummyx */.
	Press <ENTER>	The message "entered" will be displayed.
	Press <QUIT>	The screen in figure A-7 is redisplayed with the data.
24		Place the cursor on the State field with the value: DUMMY.
	Press <ENTER>	The screen in figure A-9 will be displayed with the data. Modify the State to: DUMMYX.

	Press <ENTER>	The message "entered" will be displayed.
	Press <QUIT>	The screen in figure A-7 is redisplayed with the data.
	Press <QUIT>	The screen in figure A-6 is redisplayed with the data.
25		Place the cursor on the Pattern with the value: "DUMMYX".
	Press <ENTER>	The screen in figure A-7 is displayed with the data.
		Place the cursor on the action /* dummyx */.
	Press <ENTER>	The screen in figure A-8 is displayed.
	Press <pf12>	The message "deleted" will be displayed.
	Press <QUIT>	The screen in figure A-7 is redisplayed without the data.
		Place the cursor on the state DUMMYX.
	Press <ENTER>	The screen in figure A-9 will be displayed.
	Press <pf12>	The message "deleted will be dislayed.
	Press <QUIT>	The screen in figure A-7 will be redisplayed without the data.
	Press <pf12>	The message "deleted" will be displayed.
	Press <QUIT>	The screen in figure A-6 is displayed with out the value.

26		Place the cursor on the first Pattern with the value: . (a dot).
	Press <ENTER>	The screen in figure A-7 is displayed with the data.
	Press <pf12>	The message "deleted" will be displayed.
	Press <QUIT>	The screen in figure A-6 is displayed with out the value.
	Press <QUIT>	The screen in figure A-1 is redisplayed.
27	Press <pf8>	The screen in figure A-10 is displayed.
28	Press <ENTER>	The screen in figure A-11 is displayed. Enter the value: /* DUMMY */ in the Code field.
	Press <ENTER>	The message "entered" is displayed.
	Press <QUIT>	The screen in figure A-10 is redisplayed with the data.
29		Repeat step 28 with the value: BEGIN BODY;.
30		Place the cursor on the field with the value: /* DUMMY */.
	Press <ENTER>	The screen in figure A-11 is displayed with the data. Modify the value to: /* DUMMYX */.
	Press <ENTER>	The message "entered" is displayed.
	Press <QUIT>	The screen in figure A-10 is redisplayed with the data.
31		Place the cursor on the field with the value: /* DUMMYX */.

	Press <ENTER>	The screen in figure A-11 is displayed with the data.
	Press <pf12>	The message "deleted" is displayed.
	Press <QUIT>	The screen in figure A-10 is redisplayed with out the data.
	Press <QUIT>	The screen in figure A-1 is redisplayed.
32		In the file field enter: DXUTP.DAT.
	Press <pf16>	The message "ok" is displayed.
	Press <QUIT>	The IISS Function Screen is displayed.
	Press <QUIT>	IISS terminates.
stage 2 -		
33	$ LEX DXUTP.DAT	This creates the C file LEXYY.C
34	$ CC LEXYY	This compiles the application and creates the the file LEXYY.OBJ.
35	$ LINK LEXYY,uidir:[edstag]dx, uidir:[edstag]tagolb/i=DXPRE	This links the application. The directory [edstag] and library tagolb contain the runtime support modules for DECDX document taggers.
36	$ RUN LEXYY file? DXUTP.DX out? DXUTP.TAG	This runs the application and creates a tagged document for the DECDX document.

37 $ DIFF DXUTP.TAG with DXUTP.SAV
The file DXUTP.SAV is under IISS Configuration Management. There should be no differences between these two files. This concludes this unit test of the SGML tagger.

APPENDIX A

SGML TAGGER SCREENS

```
+---------------------------------------------------------------+
|        IISS SGML Auto Tagging Facility                        |
|                                                               |
|                           +-----------------------------+    |
|Auto Tag Definition File:|                             |    |
|                           +-----------------------------+    |
|                                                               |
|   <enter> - Load Auto Tag Definition File                     |
|                                                               |
|   <pf5>   - State Definiton                                   |
|                                                               |
|   <pf6>   - Character Class Definiton                         |
|                                                               |
|   <pf7>   - Pattern/Action/State Definition                   |
|                                                               |
|   <pf8>   - C Declarations                                    |
|                                                               |
|   <pf16>  - Save Auto Tag Definition File                     |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|    +--+                                                       |
|MSG:| 0|                                            applcation|
+----+--+-------------------------------------------------------+
```

Figure A-1 SGML Tagger Main Menu

```
+------------------------------------------------------------+
|    IISS SGML Auto Tagging Facility                          |
|                                                            |
|            State Definition                                |
|                                                            |
|                                                            |
|  <enter> - Select                                          |
|                                                            |
|                                                            |
|                                                            |
|State                                                       |
+---------------------+                                      |
|                     |                                      |
+---------------------+                                      |
|                                                            |
|                                                            |
|                                                            |
|                                                            |
|                                                            |
|                                                            |
|                                                            |
|    +--+                                                    |
|MSG:| 0|                                          applcation|
+----+--+----------------------------------------------------+
```

Figure A-2 State List (Select)

```
+-----------------------------------------------------------------+
|       IISS SGML Auto Tagging Facility
|
|               State Definition
|
|
| <enter> - Insert/Update
|
| <pf12>  - Delete
|
|
|State
+--------------------+
|                    |
+--------------------+
|
|
|
|
|
|
|
|
|
|
|    +--+
|MSG:| 0|                                              applcation|
+----+--+---------------------------------------------------------+
```

Figure A-3 State (Enter/Delete)

```
+---------------------------------------------------------------+
|       IISS SGML Auto Tagging Facility                         |
|                                                               |
|          Character Class Definition                           |
|                                                               |
|       <enter> - Select                                        |
|                                                               |
|                                                               |
|Character Class      Regular Expression Definition             |
+------------------+--------------------------------------+     |
|                  |                                      |     |
+------------------+--------------------------------------+     |
|bold_on              "{#"                                      |
|                                                               |
|bold_off             "{\""                                     |
|                                                               |
|und_on               "{%"                                      |
|                                                               |
|und_off              "{$"                                      |
|                                                               |
|sup_on               "{)"                                      |
|                                                               |
|sup_off              "{("                                      |
|                                                               |
|sub_on               "{+"                                      |
|                                                               |
|sub_off              "{*"                                      |
|                                                               |
|aux_on               "{'"                                      |
|                                                               |
|aux_off              "{&"                                      |
|                                                               |
|dx_tilde             "|>"                                      |
|                                                               |
|dx_l_brace           "|;"                                      |
|                                                               |
|dx_r_brace           "|="                                      |
|                                                               |
|dx_bar               "|<"                                      |
|    +--+                                                       |
|MSG:| 0|                                            applcation|
+----+--+-------------------------------------------------------+
```

Figure A-4 Character Class List (Select)

```
+-------------------------------------------------------------+
|                  IISS SGML Auto Tagging Facility            |
|                                                             |
|                    Character Class Definition               |
|                                                             |
|                                                             |
|                                                             |
|                 <enter> - Insert/Update                     |
|                                                             |
|                 <pf12>  - Delete                            |
|                                                             |
|                                                             |
|Character Class                                              |
+---------------------+                                       |
|                     |                                       |
+---------------------+                                       |
|                                                             |
|Regular Expression Definition                                |
+-------------------------------------------------------------+
|                                                             |
+-------------------------------------------------------------+
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|    +--+                                                     |
|MSG:| 0|                                           applcation|
+----+--+-----------------------------------------------------+
```

Figure A-5 Character Class (Enter/Delete)

```
+---------------------------------------------------------------+
|                 IISS SGML Auto Tagging Facility               |
|                                                               |
|                 Pattern/Action/State Definition               |
|                                                               |
|                                                               |
|             <enter> - Select                                  |
|                                                               |
|                                                               |
|                                                               |
|Pattern                                 Action                 |
+-----------------------------------------+                     |
|                                         |                     |
+-----------------------------------------+                     |
|.                                        /* anything else */   |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|    +--+                                                       |
|MSG:| 0|                                            applcation|
+----+--+-------------------------------------------------------+
```

Figure A-6 Pattern List (Select)

```
+--------------------------------------------------------------+
|                  IISS SGML Auto Tagging Facility             |
|                                                              |
|                  Pattern/Action/State Definition             |
|                                                              |
|                                                              |
|         <enter> - Insert/Update/Select (Action or State)     |
|                                                              |
|         <pf12>  - Delete                                     |
|                                                              |
|                                                              |
|Pattern                                                       |
+--------------------------------------------------------------+
|                                                              |
+--------------------------------------------------------------+
|                                                              |
|Action                                                        |
+--------------------------------------------------------------+
|                                                              |
+--------------------------------------------------------------+
|                                                              |
|                                                              |
|State                                                         |
+---------------------+                                        |
|                     |                                        |
+---------------------+                                        |
|                                                              |
|                                                              |
|    +--+                                                      |
|MSG:| 0|                                            applcation|
+----+--+------------------------------------------------------+
```

Figure A-7 Pattern (Enter/Delete)

```
+-----------------------------------------------------------------------------+
|                      IISS SGML Auto Tagging Facility                        |
|                                                                             |
|                                  C Code                                     |
|                                                                             |
|                                                                             |
|               <enter> - Insert/Update                                      |
|                                                                             |
|               <pf12>  - Delete                                             |
|                                                                             |
|                                                                             |
|Code                                                                         |
+-----------------------------------------------------------------------------+
|                                                                             |
+-----------------------------------------------------------------------------+
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|     +--+                                                                   |
|MSG:|  1|entered                                                 applcation|
+----+--+---------------------------------------------------------------------+
```

Figure A-8 Pattern-Action (Enter/Delete)

```
+--------------------------------------------------------------+
|                 IISS SGML Auto Tagging Facility              |
|                                                              |
|                         State Definition                     |
|                                                              |
|                                                              |
|        <enter> - Insert/Update                               |
|                                                              |
|        <pf12>  - Delete                                      |
|                                                              |
|                                                              |
|                                                              |
|State                                                         |
+---------------------+                                        |
|                     |                                        |
+---------------------+                                        |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|                                                              |
|    +--+                                                      |
|MSG:| 1|entered                                    applcation|
+----+--+------------------------------------------------------+
```

Figure A-9 Pattern-State (Enter/Delete)

```
+-------------------------------------------------------------+
|                 IISS SGML Auto Tagging Facility             |
|                                                             |
|                     C Declarations Section                  |
|                                                             |
|                                                             |
|            <enter> - Select                                 |
|                                                             |
|                                                             |
|                                                             |
|Code                                                         |
+-------------------------------------------------------------+
|                                                             |
+-------------------------------------------------------------+
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|    +--+                                                     |
|MSG:| 0|                                           applcation|
+----+--+-----------------------------------------------------+
```

Figure A-10 C Code (Select)

```
+-------------------------------------------------------------+
|               IISS SGML Auto Tagging Facility               |
|                                                             |
|                           C Code                            |
|                                                             |
|                                                             |
|             <enter> - Insert/Update                         |
|                                                             |
|             <pf12>  - Delete                                |
|                                                             |
|                                                             |
|Code                                                         |
+-------------------------------------------------------------+
|                                                             |
+-------------------------------------------------------------+
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|    +--+                                                     |
|MSG:| 0|                                           applcation|
+----+--+-----------------------------------------------------+
```

Figure A-11 C Code (Enter/Delete)

APPENDIX B

FILE DECDX.DAT

This file is the template for all DECDX documents to be tagged.

```
%p 3000
%{
/* DECDX - DEC DX document template
 *
 * Description
 *   This lex source file is used as a template for DEC DX
documents. It
 *   would be loaded at the start of creating a tagger for a
particular
 *   document and saved under a different name relavent to the
document.
 */
#include <stdtyp.h>
#include <stdio.h>
#include <ctlchr.h>
#include <ctype.h>

#undef input
#undef unput
#undef YYLMAX
#undef ECHO

#define YYLMAX 10000
#define ECHO echo(yytext)
#define EMIT(x) fprintf(outfp, "%s", x)
#define yywrap() 1

static yylook(), yyback(), yyinput(), yyoutput(), yyunput();
extern FILE *outfp;
%}
bold_on                 "{#"
bold_off                "{\""
und_on                  "{%"
und_off                 "{$"
sup_on                  "{)"
sup_off                 "{("
sub_on                  "{+"
sub_off                 "{*"
aux_on                  "{'"
aux_off                 "{&"
```

```
dx_tilde                   "|>"
dx_l_brace                 "|;"
dx_r_brace                 "|="
dx_bar                     "|<"
lin_mod                    "|G"
com_on                     "|H"
dx_tab                     "|I"
dx_nl                      "|J"
dx_ff                      "|L"
com_off                    "|M"
space                      " "
hyphen                     "-"
ww_hyphen                  {aux_on}{und_on}{dx_nl}
ww_eol                     {aux_on}{dx_nl}{aux_off}
page_mark                  {aux_on}{dx_ff}{aux_off}
breaking_hyphen            {aux_on}{hyphen}{aux_off}
paragraph_mark             {sup_on}{dx_nl}{sup_oft}
center                     {sub_on}{dx_nl}{sub_off}
ruler_modified             {bold_on}{lin_mod}{bold_off}
tde_on                     {com_on}{space}"#5"{com_off}
tde_off                    {com_on}{space}"$5"{com_off}
tde_char                   {com_on}{space}4.{com_off}
any_char
([^|{]|{dx_tab}|{dx_bar}|{dx_l_brace}|{dx_r_brace}|{dx_tilde})
white_space                ({space}|{dx_tab})
%%
%{
%}
.   {
 /* anything else */
 if (yytext[0] == '|' || yytext[0] == '{') yymore();
 else ECHO;
}
%%
static char buf[YYLMAX], *pbuf = buf;
int charyy = 0;
static input()
   {
   char c;

   if (pbuf > buf) c = *pbuf--;
   else c = zzlex();
   return c;
   }
static unput(c)
   char c;
   {
```

```
   *++pbuf = c;
   }
static yyreject()
   {
   extern int yylength;

   while (yylength > 0)
     *++pbuf = yytext[--yylength];
   }
```

APPENDIX C

FILE DXUTP.DAT

This file is the result of running the AUTOTAG program, steps 2 though 32 of this unit test plan.

```
%p 3000
%{
/* DECDX - DEC DX document template
 *
 * Description
 *   This lex source file is used as a template for DEC DX
documents. It
 *   would be loaded at the start of creating a tagger for a
particular
 *   document and saved under a different name relavent to the
document.
 */
#include <stdtyp.h>
#include <stdio.h>
#include <ctlchr.h>
#include <ctype.h>

#undef input
#undef unput
#undef YYLMAX
#undef ECHO

#define YYLMAX 10000
#define ECHO echo(yytext)
#define EMIT(x) fprintf(outfp, "%s", x)
#define yywrap() 1

static yylook(), yyback(), yyinput(), yyoutput(), yyunput();
extern FILE *outfp;
%}
bold_on                 "{#"
bold_off                "{\""
und_on                  "{%"
und_off                 "{$"
sup_on                  "{)"
sup_off                 "{("
sub_on                  "{+"
sub_off                 "{*"
aux_on                  "{'"
```

```
aux_off                 "{&"
dx_tilde                "|>"
dx_l_brace              "|;"
dx_r_brace              "|="
dx_bar                  "|<"
lin_mod                 "|G"
com_on                  "|H"
dx_tab                  "|I"
dx_nl                   "|J"
dx_ff                   "|L"
com_off                 "|M"
space                   " "
hyphen                  "-"
ww_hyphen               {aux_on}{und_on}{dx_nl}
ww_eol                  {aux_on}{dx_nl}{aux_off}
page_mark               {aux_on}{dx_ff}{aux_off}
breaking_hyphen         {aux_on}{hyphen}{aux_off}
paragraph_mark          {sup_on}{dx_nl}{sup_off}
center                  {sub_on}{dx_nl}{sub_off}
ruler_modified          {bold_on}{lin_mod}{bold_off}
tde_on                  {com_on}{space}"#5"{com_off}
tde_off                 {com_on}{space}"$5"{com_off}
tde_char                {com_on}{space}4.{com_off}
any_char
([^|{]|{dx_tab}|{dx_bar}|{dx_l_brace}|{dx_r_brace}|{dx_tilde})
white_space              ({space}|{dx_tab})
snum                     ([0-9]+("."[0-9]+)+)
dnum                     ([0-9]+"-"[0-9]+)
number                   ([0-9]+)
format
({und_on}|{und_off}|{bold_on}|{bold_off}|{com_on}|{com_off})
%start BODY
%start CHAPNUM
%start READNL
%start FIGNUM
%start FIGTITLE
%start FIGEND
%start SECTITLE
%%
%{
BEGIN BODY;
%}
<BODY>{und_on}?"SECTION"{space}*{number}{und_off}?/{center} {
 /* a section header */
 EMIT("<sectnum>");
 ECHO;
 EMIT("</sectnum>");
 BEGIN SECTITLE;
}
<SECTITLE>({any_char}+{center})+/{dx_nl}{dx_nl} {
 /* section name */
 EMIT("<sectitle>");
 ECHO;
 EMIT("</sectitle>");
 BEGIN READNL;
}
<BODY>{snum}{space}*/{und_on}?{any_char}+{und_off}?{dx_nl}{dx_nl
} {
 /* a numbered paragraph */
 EMIT("<chapnum>");
 ECHO;
 EMIT("</chapnum>");
 BEGIN CHAPNUM;
}
<CHAPNUM>{und_on}?{any_char}+{und_off}?/{dx_nl}{dx_nl} {
 /* paragraph name */
 EMIT("<chptitle>");
 ECHO;
 EMIT("</chptitle>");
 BEGIN READNL;
}
<BODY>({any_char}|{ww_eol}|{format})+/{dx_nl}{white_space}*{dx_n
l} {
 /* a paragraph */
 EMIT("<para0>");
 ECHO;
 EMIT("</para0>");
 BEGIN READNL;
}
<READNL>{dx_nl}{white_space}*{dx_nl} {
 /* eat new lines after paragraphs and headers */
 EMIT("\n");
 BEGIN BODY;
}
<BODY>(({any_char}|{format})+{dx_nl})+{dx_nl} {
 /* a figure, maybe */
 EMIT("<figbody>");
 ECHO;
 EMIT("</figbody>");
}
<BODY>"Figure"{white_space}+/{dnum}{white_space}+{any_char}+{cen
ter} {
 /* figure title */
 EMIT("<figref>");
 ECHO;
 EMIT("</figref>");
 BEGIN FIGNUM;
}
<FIGNUM>{dnum}{white_space}+ {
 /* figure number */
 EMIT("<fignum>");
 ECHO;
 EMIT("</fignum>");
 BEGIN FIGTITLE;
}
<FIGTITLE>{any_char}+ {
 /* figure name */
 EMIT("<figtitle>");
 ECHO;
 EMIT("</figtitle>");
 BEGIN FIGEND;
}
<FIGEND>{center}({white_space}*{dx_nl})* {
 /* figure cleanup */
 EMIT("\n");
 BEGIN BODY;
}
. {
 /* anything else */
 if (yytext[0] == '|' || yytext[0] == '(') yymore();
 else ECHO;
}
%%
static char buf[YYLMAX], *pbuf = buf;
int charyy = 0;
static input()
   {
   char c;

   if (pbuf > buf) c = *pbuf--;
   else c = zzlex();
   return c;
   }
static unput(c)
   char c;
   {
   *++pbuf = c;
   }
static yyreject()
   {
   extern int yylength;
while (yylength > 0)
  *++pbuf = yytext[--yylength];
}
```

APPENDIX D

FILE DXUTP.TAG

This file is the result of running the generated tagger program in step 37 of this unit test plan.

```
<sectnum>SECTION 1</sectnum><sectitle>

SCOPE
</sectitle>
<chapnum>1.1  </chapnum><chptitle>Identification</chptitle>
<para0>     This specification establishes the detailed
requirements
for performance, design, test, and qualification of a computer
program identified as the Forms Definition Language Compiler,
hereinafter  referred to as FLAN.  FLAN is one configuration
item of the Integrated Information Support System User Interface

(IISS UI).</para0>
<chapnum>1.2  </chapnum><chptitle>Functional Summary</chptitle>
<para0> FLAN is used to compile Form Definition Language source
files into binary Form Definition File format.  Form Definition
files are  used as input to the Form Processor in displaying the

forms.  REVFLAN is used to create a Form Definition Language
source file from one or more version 1.0 Form Definition Files
which were derived from the DEC FMS utility before FLAN was
implemented and which had no source file.  MAKINC creates
program variable declarations which correspond to the structure
of a form and are useful to application programs which make use
of Form Processor calls to PDATA and GDATA.</para0>
<para0>     The parser and some procedures of FLAN are used by
the
Forms Driven Forms Editor (FDFE), the Report Writer (RW), and
the Rapid Application Generator (RAP) (all three of which are
configuration items).  This ensures that the language which is
accepted by them is identical.</para0>
<sectnum>
SECTION 2</sectnum><sectitle>

DOCUMENTS
</sectitle>
<chapnum>2.1  </chapnum><chptitle>Reference Documents</chptitle>
<para0>[1]  Systran, ICAM Documentation Standards,
IDS150120000C,
```

15 September 1983.</para0>
<para0>[2] IISS Integration Task Force, Final Report,
1984.</para0>
<chapnum>2.2 </chapnum><chptitle>Terms and
Abbreviations</chptitle>
<para0> 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.</para0>
<para0> 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.</para0>
<sectnum>
SECTION 3</sectnum><sectitle>

REQUIREMENTS
</sectitle>
<chapnum>3.1 </chapnum><chptitle>Computer Program
Definition</chptitle>
<para0> 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.</para0>
<para0> 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
configuration management software can be used in managing Forms
Definition Language files in a manner similar to other source
files.</para0>
<chapnum>3.1.1 </chapnum><chptitle>System Capacities</chptitle>
<para0> FLAN is written in the C programming language and runs
on a
DEC VAX minicomputer under the VMS operating system.</para0>
<figbody>

```
             MYFORMS.FDL
        +------------------------+
        |                        |
        |  CREATE FORM F1        |  ***********
        |     Background Black   |  * process *
        |     Prompt             |  ***********
        |        Center at 2 40  |
        |        "Form F1"       |
        |     Item A             |  +---------+
        |        .               |  |  data   |
        |        .               |  +---------+
        |                        |
        |  CREATE FORM F2        |
        |        .               |
        |        .               |
        |                        |
        |  CREATE FORM F3        |
        |        .               |
        |        .               |
        |________________________|
                    |
          **********************
          *        FLAN        *
          **********************
                    |
      +-------------+-------------+
      |             |             |
      |  F1.FD      |  F2.FD      |  F3.FD
+-----+------+  +---+------+  +---+------+
|            |  |          |  |          |
+-----+------+  +---+------+  +---+------+
      |             |             |
      +-------------+-------------+
                    |
      +-------------+--------------+
      |             |              |
 *************  ************  ***********
 *  Form     *  * REVFLAN  *  * MAKINC  *
 * Processor *  *          *  *         *
 *           *  ************  ***********
 *************       |              |
             +----------+  +---------------+
             |  .FDL    |  |    variable   |
             |          |  |  declarations |
             +----------+  +---------------+
```

</figbody><figref>Figure </figref><fignum>3-1
</fignum><figtitle>FLAN Interfaces</figtitle>
<chapnum>3.1.2.2 </chapnum><chptitle>Detailed Interface Definition</chptitle>
<chapnum>3.1.2.2.1 </chapnum><chptitle>Form Language Compiler Interfaces</chptitle>
<chapnum>3.1.2.2.1.1 </chapnum><chptitle>Form Definition Language</chptitle>
<para0> The syntax of the Form Definition Language accepted as input by FLAN is documented in Appendix A. This language is intended to provide access to all Form Processor functionality.

It is also intended to be a LALR(1) Grammar for ease of parsing.

A number of automatic parser generators are available for LALR(1) grammars, most notably the UNIX* utility YACC.</para0>
<chapnum>3.1.2.2.1.2 </chapnum><chptitle>Binary Form File Format</chptitle>
<para0> The structure of the records in the binary Form Definition
Files produced as output by FLAN are documented in the Form Processor Development Specification. These records are contained in the include file fpd.h. The sequence of records in

the file is:</para0>
<figbody> 1) The version record which identifies the file format.
2) A FIELD record.
3) The name of the permanent attribute for this field.
4) A TEXT record and string, if any. This repeats for each prompt associated with this field.
5) An ENODE record, if any. The expression is in a prefix order.
6) The help text if this field is an item.
7) The ATTMAP record, attribute mapping if this is a form.

</figbody><sectnum>
SECTION 4</sectnum><sectitle>

QUALITY ASSURANCE PROVISIONS
</sectitle>
<chapnum>4.1 </chapnum><chptitle>Introduction and Definitions</chptitle>
<para0> "Testing" is a systematic process that may be preplanned

and explicitly stated. Test techniques and procedures may be defined in advance and a sequence of test steps may be

specified. "Debugging" is the process of isolation and
correction of the cause of an error.</para0>
<para0>   "Antibugging" is defined as the philosophy of
writing
programs in such a way as to make bugs less likely to occur and
when they do occur, to make them more noticeable to the
programmer and the user. In other words, as much error checking

as is practical and possible in each routine should be
performed.</para0>
<sectnum>
SECTION 5</sectnum><sectitle>

PREPARATION FOR DELIVERY
</sectitle>
<para0>   The implementation site for the constructed software
is the
ICAM Integrated Support System (IISS) Test Bed site located at
Arizona State University, Tempe, Arizona. The software
associated with each CPCI release is delivered on a media which
is compatible with the IISS Test Bed. The release is clearly
identified and includes instructions on procedures to be
followed for installation of the release. Integration with the

other IISS CPCI's will be done on the IISS TEST BED on a
scheduled basis.</para0>