

AD-A182 580

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS) VOLUME 8
USER INTERFACE SUBS (U) GENERAL ELECTRIC CO
SCHENECTADY NY PRODUCTION RESOURCES CONSU

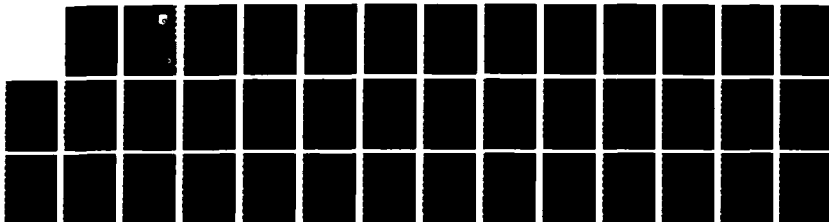
1/1

UNCLASSIFIED

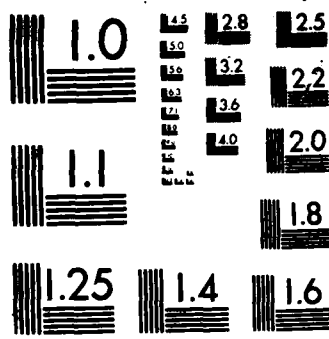
A WEHRMAN ET AL 01 NOV 85 UTP-620144300

F/G 12/5

NL



END
8 87
DTIC



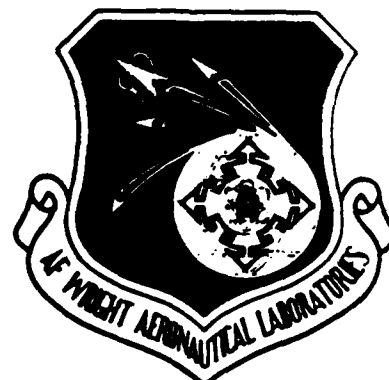
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A182 580

AFWAL-TR-86-4006
Volume VIII
Part 12

DTIC FILE COPY

2



INTEGRATED INFORMATION
SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 12 - Virtual Terminal 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

DTIC
ELECTE
JUL 16 1987
S E D

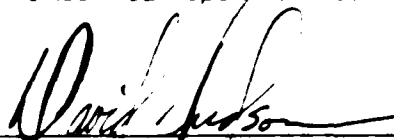
87 7 15 041

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.

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 12	
6a NAME OF PERFORMING ORGANIZATION General Electric Company Production Resources Consulting	6b OFFICE SYMBOL (If applicable) AFVAL/MLTC	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 45433-8533	
8a NAME OF FUNDING/SPONSORING ORGANIZATION Air Force 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 780117	PROJECT NO 7500
		TASK NO 82	WORK UNIT NO 01
11 TITLE (Include Security Classification) (See Reverse)			
12 PERSONAL AUTHOR(S) Wehrman, Al and Morenc, Carol and Robie, Penny			
12a TYPE OF REPORT Final Technical Report	12b TIME COVERED 22 Sept 1980 - 31 July 1985	12c DATE OF REPORT (Yr, Mo, Day) 1985 November	12d PAGE COUNT 34
16 SUPPLEMENTARY NOTATION ICAM Project Priority 8201		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) This unit test plan establishes the methodology and procedures used to adequately test the capabilities of the computer program identified as the Virtual Terminal (VT). The VT translates between VT commands and commands for a particular type of terminal. Part of the VT handles sending and receiving messages across the NTH. Another part interfaces with the terminal specific device driver.			
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) 813-255-8876	22c OFFICE SYMBOL AFVAL/MLTC

11. Title

Integrated Information Support System (IISS)
Vol VIII - User Interface Subsystem
Part 12 - Virtual Terminal Unit Test Plan

A S D 86 0030
9 Jan 1986

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



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

Subcontractors

Role

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.

Subcontractors

Role

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)

Subcontractors

Role

Boeing Military Aircraft
Company (EMAC)

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.

Subcontractors

Role

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 (EMAC)	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 Purpose	1-1
1.2 Project References	1-1
1.3 Terms and Abbreviations	1-3
SECTION 2.0 DEVELOPMENT ACTIVITY	
2.1 Statement of Pretest Activity	2-1
2.2 Pretest Activity Results	2-1
SECTION 3.0 SYSTEM DESCRIPTION	
3.1 System Description	3-1
3.2 Testing Schedule	3-4
3.3 First Location Testing	3-4
3.4 Subsequent Location Testing	3-4
SECTION 4.0 SPECIFICATIONS AND EVALUATIONS	
4.1 Test Specifications	4-1
4.2 Test Methods and Constraints	4-4
4.3 Test Progression	4-4
4.4 Test Evaluation	4-4
SECTION 5.0 TEST PROCEDURES	
5.1 Test Description	5-1
5.2 Test Control	5-1
5.3 Test Procedures	5-1

APPENDICES

A - Virtual Terminal Command Descriptions	A-1
B - Sample Link Command Descriptions	B-1

FIGURES

3-1 Virtual Terminal Interfaces (in Normal IISS Environment)	3-2
3-2 Virtual Terminal Interfaces (for Unit Testing)	3-3
4-1 Functional Requirements Mapping	4-4

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 Virtual Terminal, known in this document as the VT. The VT is one configuration item of the Integrated Information Support System (IISS) User Interface (UI). It consists of Virtual Terminal callable routines and the a monitor which is the main controller of the Virtual Terminal.

1.2 Project References

- [1] ICAM Documentation Standards: 15 September 1983; IDS150120000C.
- [2] Structural Dynamics Research Corporation, IISS Terminal Operator Guide OM 620144000 , 1 November 1985.
- [3] Structural Dynamics Research Corporation, IISS Form Processor User Manual, UM 620144200B, 1 November 1985.
- [4] Structural Dynamics Research Corporation, Virtual Terminal User Manual, UM 620144300B, 1 November 1985.
- [5] Structural Dynamics Research Corporation, Form Processor Development Specification, DS 620144200B, 1 November 1985.
- [6] Structural Dynamics Research Corporation, Virtual Terminal Interface Development Specification, DS 620144300B, 1 November 1985.
- [7] Structural Dynamics Research Corporation, Forms Language Compiler Unit Test Plan, UTP620144401 , 1 November 1985.
- [8] Structural Dynamics Research Corporation, Forms Driven Form Editor Unit Test Plan, UTP62014402 , 1 November 1985.

UTP620144300
1 November 1985

- [9] Structural Dynamics Research Corporation, Report
Writer Unit Test Plan, UTP620144501 , 1 November
1985.
- [10] Structural Dynamics Researc Corporation, Rapid
Application Generator Unit Test Plan, UTP620144502 ,
1 November 1985.
- [11] Structural Dynamics Research Corporation, Text
Editor Unit Test Plan, UTP620144600 , 1 November
1985.
- [12] Structural Dynamics Research Corporation, Application
Interface Unit Test Plan, UTP620144700 , 1 November
1985.
- [13] Structural Dynamics Research Corporation, Form
Processor Unit Test Plan, UTP620144200 , 1 November
1985.

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.

Application Interface: (AI), subset of the IISS User Interface that consists of the callable routines that are linked with applications that use the Form Processor or Virtual Terminal. The AI enables applications to be hosted on computers other than the host of the User Interface.

Application Process: (AP), a cohesive unit of software that can be initiated as a unit to perform some function or functions.

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.

Communication Services: allows on host interprocess communication and inter-host communication between the various Test Bed subsystems.

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

Device Drivers: (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.

Extended Binary Coded Decimal Interchange Code: (EBCDIC), the character set used by a few computer vendors (notably IBM) instead of ASCII.

Field: two-dimensional space on a terminal screen.

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.

Logical Device: a conceptual device which to an application is indistinguishable from a physical device and is then mapped to part or all of a physical device.

Network Transaction Manager: (NTM), IISS subsystem that performs the coordination, communication and housekeeping functions required to integrate the Application Processes and System Services resident on the various hosts into a cohesive system.

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.

Physical Device: a hardware terminal.

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 runtime UI. It consists of the Form Processor, Virtual Terminal, Application Interface, the User Interface Services and the Text Editor.

User Interface Monitor: (UIM), part of the Form Processor that handles messaging between the NTM and the UI. It also provides authorization checks and initiates applications.

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

Virtual Terminal: (VT), 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.

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.

Each Virtual Terminal function was tested individually through Virtual Terminal development.

All pretesting activity was conducted by the individual program developer in a manual mode. The developer manually entered data onto the screen and observed the results. Any errors were noted by the developer, and corrections to the Virtual Terminal software were then made after a testing session.

2.2 Pretest Activity Results

The pretest activity was very successful in the elimination of programming bugs.

SECTION 3

SYSTEM DESCRIPTION

3.1 System Description

The Virtual Terminal translates between neutral commands and commands for a particular type of terminal. Part of the Virtual Terminal handles sending and receiving messages across the NTH. Another part interfaces with the terminal specific device driver. See the Virtual Terminal Development Specification [6].

3.1.1 Interface Requirements

The Virtual Terminal for new IISS applications is the set of Virtual Terminal input/output routines defined in the Virtual Terminal User Manual [4]. The AP uses the input/output routines to get input data from the terminal and send output data to the terminal. The data consists of Virtual Terminal commands which are documented in Appendix A.

An AP that is to be integrated into the IISS and uses the callable interface to the VT must include VT initialization (INITVT) and termination (TERMVT) calls at the beginning and end of its BYPASS of FP I/O routines. If FP routines are called while in bypass mode, results are unpredictable. Since the Form Processor routines INITFP and TERMFP perform functions necessary for the monitoring activity of the entire User Interface, INITFP and TERMFP must be called at the beginning and end of AP's Procedure Division even if the AP only uses the Virtual Terminal routines. See the Form Processor User Manual [3].

The VT interfaces with both the Form Processor and a physical terminal. An application that uses the VT normally calls the routines INITVT, GETVTI, PUTVTI and TERMFP of the Application Interface, but since routines with identical calling sequences are part of the Form Processor callable routines, testing may be simplified by linking directly with these routines instead of the Application Interface. Figures 3-1 and 3-2 illustrate the normal VT interfaces and the interfaces for testing purposes.

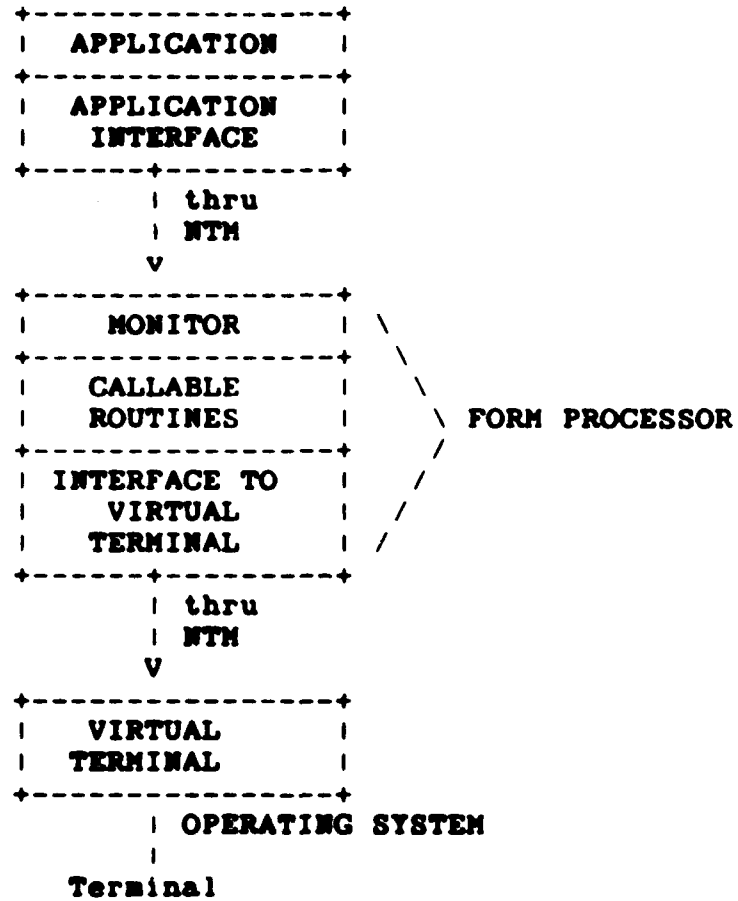


Figure 3-1 Virtual Terminal Interfaces
(in Normal IISS Environment)

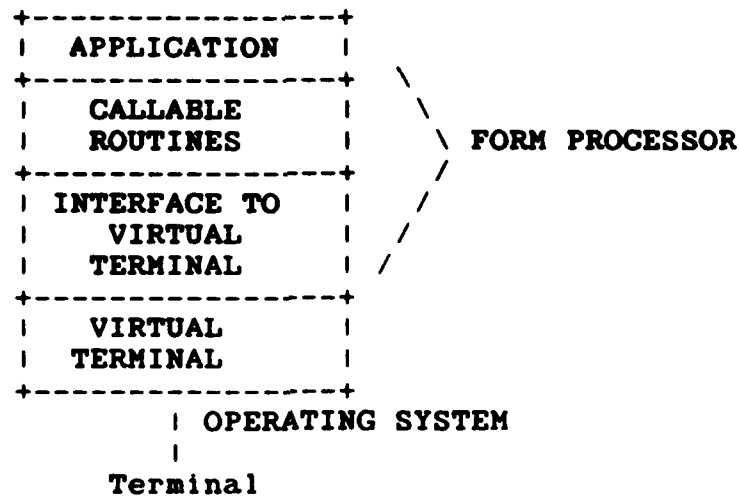


Figure 3-2 Virtual Terminal Interfaces (for Unit Testing)

The VT is implemented as a Device Driver architecture -- there is a separate implementation of the device driver routines for each supported terminal. The device driver routines are linked with the neutral VT software to form the VT subsystem.

3.2 Testing Schedule

As figure 3-2 indicates, the execution of the Virtual Terminal is dependent on some callable routines of the Form Processor subsystem (INITVT, GETVTI, PUTVTI and TERMVT) and the interface to the VT. These two portions of the FP are actually a small part of the entire FP. Since the Form Processor subsystem is dependent on the whole Virtual Terminal subsystem, testing should start with the Virtual Terminal subsystem and then the Form Processor subsystem should be tested.

3.3 First Location Testing

These tests of the Virtual Terminal 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 FP subsystem of the UI.

Personnel: One integrator familiar with the IISS.

Training: The VT User manual.

Deliverables: The Virtual Terminal Subsystem of the IISS
UI/VTI.

Test Materials: This test may be run interactively by
writing a test program with the appropriate
VT calls and then by inputting the
appropriate data and observing the
output as outlined in this test plan.

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 functions should be performed in order to test the Virtual Terminal:

- 1) Go into Form Processor Bypass Mode
- 2) Send the Virtual Terminal commands
 - a) Define Window
 - b) Set Window Precedence
 - c) Remove Window
 - d) Select Window
 - e) Erase Window
 - f) Define Field
 - g) Set Mode
 - h) Reset Mode
 - i) Reset to Initial State
 - j) Refresh Screen
 - k) Set Graphic Rendition
 - l) Define Area Qualification
 - m) Horizontal Tab Set
 - n) Device Status Request
 - o) Record Separator

- 3) Get the Virtual Terminal commands
 - a) Set Transmit State
 - b) Select Window
 - c) Define Field
 - d) Application Program Command
 - e) Cursor Position Report
- 4) Exit Form Processor Bypass Mode
- 5) Obtain and Interpret Terminal Input
 - a) Translate and echo back special key board characters
 - Bell
 - Backspace
 - Backtab
 - Tab
 - Line Feed
 - Form Feed
 - Carriage Return
 - Next Line
 - b) Cursor Maintenance
 - Cursor Up
 - Cursor Down
 - Cursor Forward
 - Cursor Backward
 - Cursor Next Line
 - Cursor Previous Line
 - c) Data Entry and Deletion/Erasure
 - Enter Printable Characters
 - Erase Display
 - Erase Line
 - Insert Line
 - Delete Line
 - Erase Field
 - Delete Character
 - Erase Character
 - d) Media Copy
 - e) Device Control String

Figure 4-1 Maps these functional requirements to test activities.

Functional Requirements\Test activities	A	B	C	D	E	F	G	H	I	J
Go into Form Processor Bypass Mode	*									
Define Window	*	*								
Set Window Precedence					*					
Remove Window	*						*			
Select Window	*	*	*	*	*	*	*			
Erase Window	*									
Define Field	*	*	*	*		*				
Set Mode	*	*		*	*					
Reset Mode	*	*		*	*					
Reset to Initial State	*	*		*	*					
Refresh Screen	*	*		*	*					
Set Graphic Rendition	*	*		*	*					
Define Area Qualification	*	*		*	*					
Horizontal Tab Set	*	*		*	*					
Device Status Request	*	*		*	*					
Record Separator	*	*		*	*					
Set Transmit State		*		*	*					
Application Program Command		*		*	*					
Cursor Position Report		*		*	*					

Continued on next page

Functional Requirements\Test activities	A	B	C	D	E	F	G	H	I	J
Go out of Form Processor Bypass Mode										*
Translate & Echo Special Characters										*
Cursor Maintenance										*
Data Entry & Deletion/Erasure										*
Media Copy (Send Screen to Printer)										*
Device Control String										*

Figure 4-1 Functional Requirements Mapping

Key to Figure 4-1

- A - call initvt
- B - first call putvti
- C - first call getvti
- D - second call putvti
- E - second call getvti
- F - third call putvti
- G - third call getvti
- H - forth call putvti
- I - call termvt
- J - keyboard input

4.2 Testing Methods and Constraints

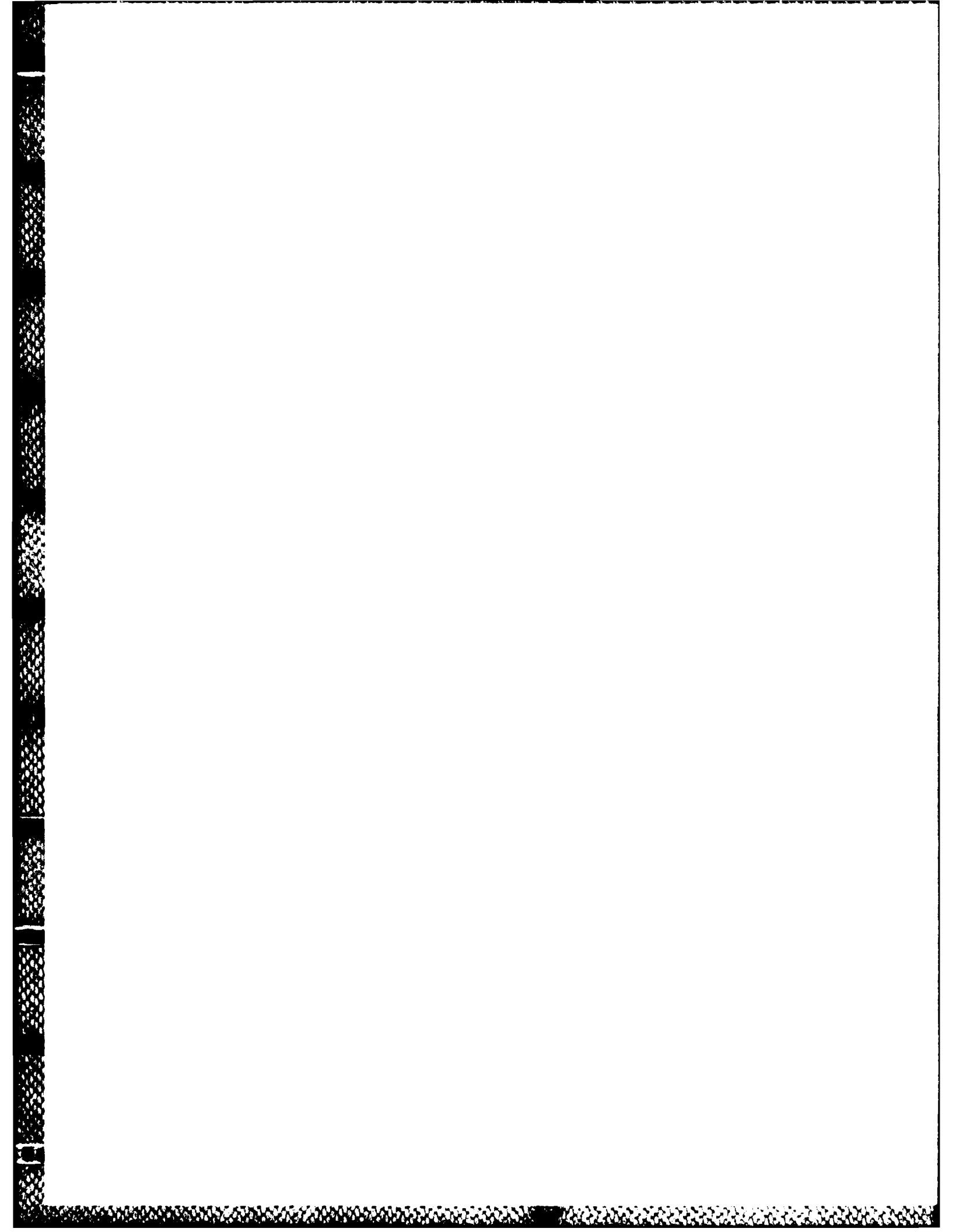
The tests as outlined in Section 5 must be followed. The required calling sequence is stated for the test program. This testing uses the normal mode of operation of these functions and does not completely exercise all the error combinations that a user of the Virtual Terminal might create by faulty entry of command sequences. Much of this testing has 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 3.3 of this unit test plan.

4.3 Test Progression

The progression of testing of the Virtual Terminal is fully outlined in Section 5 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 screens to that specified as successful for the given test, as outlined in Section 5.



SECTION 5

TEST PROCEDURES

5.1 Test Description

This test is performed by writing a test program which uses the callable interface to exercise the functionality of the Virtual Terminal. In section 5.3 the required function calls and the required parameters are documented for each function being tested and the resulting successful results are also documented.

5.2 Test Control

The order of the testing is also documented in section 5-3. The test control information is completely described by the sequence of calls.

5.3 Test Procedures

- I) The program which should be written to test the Virtual Terminal should contain the following calls:

CALL INITFP.

CALL INITVT USING RCODE.

CALL INQLDV USING LDV-ID, RCODE.

CALL PUTVTI USING BUFFER, LENGTH, RCODE.

Where BUFFER contains

SW - Select Window: <ESC>[XXs

EW - Erase Window: <ESC>[XXu

Where XX = logical device ID obtained through INQLDV call
(INQLDV returns an integer but XX MUST BE an ascii number)

DW - Define Window: <ESC>[900;1;1;80;23;0;0;80;23;0w

SW - Select Window: <ESC>[900s

DF - Define Field: <ESC>[12;10;20;5;0;0;7x

DF - Define Field: <ESC>[10;15;10;1;1;0;1x-Testing1-

RS - Record Separator: <RS>

and LENGTH is 89 plus twice the length of XX

CALL GETVTI USING BUFFER, MAX_LENGTH, LENGTH, RCODE.
Where MAX_LENGTH = Length of BUFFER

CALL PUTVTI USING BUFFER, LENGTH, RCODE.
Where BUFFER contains

SW - Select Window: <ESC>[XXs

Where XX = logical device ID obtained through INQLDV call

DW - Define Window: <ESC>[901;1;1;80;23;0;0;80;23;0w

SW - Select Window: <ESC>[901s

DF - Define Field: <ESC>[2;10;20;15;0;0;7x

DF - Define Field: <ESC>[20;15;10;1;1;0;5x-Testing2-

RS - Record Separator: <RS>

and LENGTH is 86 plus the length of XX

CALL GETVTI USING BUFFER, MAX_LENGTH, LENGTH, RCODE.
Where MAX_LENGTH = Length of BUFFER

CALL PUTVTI USING BUFFER, LENGTH, RCODE.
Where BUFFER contains

SW - Select Window: <ESC>[XXs

Where XX = logical device ID obtained through INQLDV call

WP - Set Window Precedence: <ESC>[900p

RS - Record Separator: <RS>

and LENGTH is 10 plus the length of XX

CALL GETVTI USING BUFFER, MAX_LENGTH, LENGTH, RCODE.
Where MAX_LENGTH = Length of BUFFER

CALL PUTVTI USING BUFFER, LENGTH, RCODE.
Where BUFFER contains

SW - Select Window: <ESC>[XXs

Where XX = logical device ID obtained through INQLDV call

RW - Remove Window <ESC>[900r

RS - Record Separator: <RS>

and LENGTH is 10 plus the length of XX

CALL TERMTVT USING RCODE.

CALL TERMFP.

II) Link this program with Form Processor and Virtual Terminal
libraries. see Appendix B.

III) Run your program

- a) First screen should have two fields on it:
 - One beginning at row 12 col. 10 which is 20 col. wide and 5 rows deep. It should be reverse.
 - One beginning at row 10 col. 15 which is 10 col. wide and 1 row deep. Initialized to "-Testing1-". It should be bold.
- b) Move Cursor down then right then up to top left corner of screen.
- c) Depress Tab.
- d) Cursor should be at input field enter something.
- e) Depress a Function Key.
The buffer received by getvti should contain a string containing set window commands followed by define field commands followed by any data which may have entered. The last data in buffer should be an APC command and a CPR command (see Appendix A).
- f) Second screen should have two fields on it:
 - One beginning at row 2 col. 10 which is 20 col. wide and 15 rows deep. It should be reverse.
 - One beginning at row 20 col. 15 which is 10 col. wide and 1 row deep. Ininitialized to "-Testing2-". It should be blinking.
- g) Without entering anything Depress a Function Key.
The buffer received by getvti should contain a string containing set window commands followed by define field commands. There should be no data after define field commands. The last data in buffer should be an APC command and a CPR command (see Appendix A).
- h) Next screen should be the same as your first.
- i) At this time you could test out some of the other commands for (see Appendix A).
 - Bell
 - Backspace
 - Backtab
 - Tab
 - Line Feed
 - Form Feed
 - Carriage Return
 - Next Line
 - Media Copy - "Print Key" control sequence
(AT PRESENT FOR CI600 ONLY)
 - Device Control String - Graphic's Pass-through

- j) Depress a Function Key.
- k) Next screen should be the same as your second.
- l) At this time you could test out some of the other commands for (see Appendix A).
 - Enter Printable Characters
 - Erase Display
 - Erase Line
 - Insert Line
 - Delete Line
 - Erase Field
 - Delete Character
 - Erase Character
- m) Depress a Function Key

Appendix A

Virtual Terminal Command Descriptions

The format of the following command descriptions is the command name and short description, the command syntax, and a detailed description of the command. In the command syntax, characters within angle brackets (e.g. <ESC>) indicate Control Characters. Pn indicates a Numeric Parameter, Ps indicates a Selective Parameter, and all other characters stand for themselves. Parameters are represented in ordinary human-readable decimal form, with Numeric Parameters representing numbers (such as a row number or the number of times to repeat a function), and Selective Parameters standing for selections from a list of options with multiple selections separated by semi-colons. Unless specified otherwise, Numeric Parameters indicate the number of times to repeat the specified function, omitted Numeric Parameters are taken to be 1, and omitted Selective Parameters are taken to be 0.

The Virtual Terminal screen consists of X rows numbered from 1 to X, and Y columns numbered from 1 to Y. The standard ordering of objects is from top to bottom and left to right, with wrap-around from the last object to the first. "Next" in the command descriptions refers to this order, "previous" to its reverse. For example (if Y=80), from row 6 column 80, the next character position is row 7 column 1, and the previous character position is row 6 column 79.

In Forms Mode, any command whose effect is limited to a single field (including Graphic Characters) causes the cursor to move to the next unprotected field before the command takes effect if the cursor is in a protected field when the command is received. If there are no unprotected fields defined, the command is ignored.

Graphic Characters

Cause the specified character to be displayed according to the graphic rendition in effect at the cursor location and advance the cursor to the next character position. This advancing may possibly cause scrolling.

BEL - Sound Bell

·BEL·

Sounds an audible alarm at the terminal.

BS - Backspace

·BS·

Moves the cursor to the previous character position; if the cursor is at the left margin, no action occurs.

HT - Horizontal Tab

·HT·

Moves the cursor to the next horizontal tab stop on the current line or to the right margin if no more tab stops exist. In Forms Mode, moves the cursor to the next field

LF - Line Feed

·LF·

Moves the cursor down to the next line in the current column and may possibly cause scrolling the screen.

FF - Form Feed

·FF·

Clears the screen and moves the cursor to the first unprotected character position. In Forms Mode, only unprotected areas of the screen are erased.

CR - Carriage Return

·CR·

Moves the cursor to the left margin in the current line.

RS - Record Separator

·RS·

Used to indicate end of a formatted buffer.

US - Unit Separator

·US·

Same as HT

IND - Index

·ESC· D

Same as LF.

NEL - Next Line

·ESC· E

Same as CR followed by LF

HTS - Horizontal Tab Set

ESC H

Sets a horizontal tab stop at the current column.

RI - Reverse Index

ESC M

Moves the cursor up to the previous line in the current column, possibly scrolling the screen.

DCS - Device Control String

**ESC P ... ESC **

Transmits the characters between the escape sequences (...) directly to the physical terminal without interpretation. This may be used to activate special features of a particular terminal, but it is the user's responsibility to insure that the physical terminal is of the correct type.

STS - Set Transmit State

ESC S

Indicates that the currently selected window is to be enabled for input. All unguarded fields are made enterable and a data message will be sent when a function key is pressed.

Application Program Command

**ESC Pn ESC **

Generated when a function key is pressed. The parameter is the function key number (0 - n) which must not be omitted. Function key zero is the "ENTER" key.

RIS - Reset to Initial State

ESC c

Resets the terminal to its initial state. The screen is cleared, the cursor is positioned in the upper left corner, and Forms Mode is reset.

REF - Refresh Screen

ESC ?

Retransmits the current screen contents to the terminal. Its main uses are to recover from unsolicited messages or line noise which have corrupted the screen contents, or to update the terminal when in Deferred Display Mode.

ICH - Insert Character

.ESC. [Pn @

Makes room for a character by shifting the rest of the line (field in Forms Mode) one character position to the right; characters shifted past the end of the line (field) are lost. The cursor is left at the first inserted character position (i.e. not moved).

CUU - Cursor Up

.ESC. [Pn A

Moves the cursor to the previous line in the current column, but not past the top margin.

CUD - Cursor Down

.ESC. [Pn B

Moves the cursor to the next line in the current column, but not past the bottom margin.

CUF - Cursor Forward

.ESC. [Pn C

Moves the cursor to the next character position, but not past the right margin.

CUB - Cursor Backward

.ESC. [Pn D

Moves the cursor to the previous character position, but not past the left margin.

CNL - Cursor Next Line

.ESC. [Pn E

Moves the cursor to the left margin of the next line, but not past the bottom margin.

CPL - Cursor Previous Line

.ESC. [Pn F

Moves the cursor to the left margin of the previous line, but not past the top margin.

CUP - Cursor Position

.ESC. [Pn ; Pn H

Moves the cursor to the specified position. The first parameter is the row number, the second parameter is the column number. If both parameters are omitted, the semi-colon may be omitted as well.

CHT - Cursor Horizontal Tab

<ESC> [Pn I

Moves the cursor to the next horizontal tab stop on the current line or the right margin if no more horizontal tab stops exist. In Forms Mode, moves the cursor to the next field.

ED - Erase Display

<ESC> [Ps J

Erases the screen according to the parameter:

- 0 - Erase from the cursor to the end of the screen
- 1 - Erase from the beginning of the screen to the cursor
- 2 - Erase the entire screen

The cursor is not moved. In Forms Mode, only unprotected areas of the screen are erased.

EL - Erase Line

<ESC> [Ps K

Erases the current line according to the parameter:

- 0 - Erase from the cursor to the end of the line
- 1 - Erase from the beginning of the line to the cursor
- 2 - Erase the entire line

The cursor is not moved. In Forms Mode, only unprotected areas of the screen are erased.

IL - Insert Line

<ESC> [Pn L

Makes room for a line by shifting the rest of the screen down one line; lines shifted past the bottom of the screen are lost. The cursor is positioned at the first inserted line (i.e. not moved).

DL - Delete Line

<ESC> [Pn M

Deletes the current line by shifting the rest of the screen up one line.

EF - Erase Field

<ESC> [Ps N

Erases the current field according to the parameter:

- 0 - Erase from the cursor to the end of the field
- 1 - Erase from the beginning of the field to the cursor
- 2 - Erase the entire field

The cursor is not moved.

DCH - Delete Character

ESC [Pn P

Deletes the current character by shifting the rest of the line (field in Forms Mode) one character position to the left.

CPR - Cursor Position Report

ESC [Pn ; Pn R

Generated in reply to a cursor position request (see DSR). The first parameter is the current row, the second parameter is the current column.

NP - Next Page

ESC [Pn U

Same as FF.

PP - Previous Page

ESC [Pn V

Same as FF.

ECH - Erase Character

ESC [Pn X

Erases the current character (the character is NOT deleted). The cursor is not moved. In Forms Mode, only a single field is affected.

CBT - Cursor Backward Tab

ESC [Pn Z

Moves the cursor to the previous horizontal tab stop in the current line or to the left margin if no more horizontal tab stops exist. In Forms Mode, moves the cursor to the previous field.

HPA - Horizontal Position Absolute

ESC [Pn

Moves the cursor to the specified column in the current line.

HPR - Horizontal Position Relative

ESC [Pn a

Same as CUF.

VPA - Vertical Position Absolute

ESC [Pn d

Moves the cursor to the specified line in the current column.

VPR - Vertical Position Relative

⟨ESC⟩ [Pn e

Same as CUD.

HVP - Horizontal and Vertical Position

⟨ESC⟩ [Pn ; Pn f

Same as CUP.

TBC - Tab Clear

⟨ESC⟩ [Ps g

Clears tab stops according to the parameter:

- 0 - Clear the horizontal tab stop at the cursor
- 3 - Clear all horizontal tab stops

SM - Set Mode

⟨ESC⟩ [Ps h (standard modes)

⟨ESC⟩ [? Ps h (private modes)

Sets the indicated modes; standard and private modes can not be mixed. No standard modes are currently supported.

Allowable private mode parameters are:

1 - FRMM - Forms Mode - When set, area qualifications are enforced and reading the terminal results in a full-screen formatted buffer; when reset, area qualifications are not enforced and reads return a single line or command at a time.

3 - CTM - Control Transfer Mode - When set, indicates that control sequences are to be returned to the program; when reset, control sequences terminate a read but are not returned. (Only effective when not in Forms Mode.)

4 - DDM - Deferred Display Mode - When set, indicates that writes are to affect only the internal buffer, not the screen (a REF command should be sent to update the screen); when reset, indicates that writes affect both the internal buffer and the screen.

MC - Media Copy

⟨ESC⟩ [Ps i

Controls the transfer of data between the device and an auxiliary input/output device:

- 0 - Print Screen

RM - Reset Mode

⟨ESC⟩ [Ps 1 (standard modes)

⟨ESC⟩ [? Ps 1 (private modes)

Resets the indicated modes; standard and private modes can not be mixed.

SGR - Set Graphic Rendition

⟨ESC⟩ [Ps m

Sets the specified Graphic Rendition:

0 - Normal (reset existing attributes)

1 - Bright or Bold

2 - Dim

4 - Underlined

5 - Slow Blink (less than 150 per minute)

6 - Fast Blink

7 - Reverse

8 - Concealed (not displayed)

The specified attributes are in effect from the cursor position to the next SGR or the end of the current line, whichever comes first. Note that the specified attributes are IN ADDITION to the currently existing attributes unless Normal is specified.

DSR - Device Status Request

⟨ESC⟩ [Ps n

Requests the indicated status:

6 - Report Cursor Position (via CPR)

DAQ - Define Area Qualification

⟨ESC⟩ [Ps o

Sets the specified Area Qualification:

0 - No Qualification (reset existing qualifications)

1 - Protected and Guarded

7 - Beginning of Field

The specified qualification is in effect from the cursor position to the next DAQ or the end of the current line, whichever comes first. More than one qualification MUST be specified by separating them with semicolons. Note that the specified qualifications are IN ADDITION to the currently existing qualifications unless No Qualification is specified. Area qualifications are only enforced in Forms Mode. DAQ commands take up a single character space on the screen which is displayed as a blank; the cursor is moved to the next character position following a DAQ command. When DAQ and SGR are used together, the SGR command should be given first, followed by the DAQ command. (The screen is

completely protected and guarded unless other qualifications are explicitly specified.)

WP - Set Window Precedence

⟨ESC⟩ [Pn ; ... p

Sets the precedence of the specified windows. Each is in turn placed on top of all other existing windows. Thus, the last window specified will ultimately be the top-most, and all specified windows will be on the top of any unspecified windows.

RW - Remove Window

⟨ESC⟩ [Pn r

Removes a window for formatted screen. The parameter is the window id.

SW - Select Window

⟨ESC⟩ [Pn s

Selects a window for formatted screen. Must use this command before using DF and DW commands - all windows and fields are defined as being children of the window chosen by SW command. The default is the device itself. The parameter is the window id.

EW - Erase Window

⟨ESC⟩ [Pn u

Removes all children windows and fields of a window for formatted screen. The parameter is the window id.

DW - Define Window

⟨ESC⟩ [Pn ; Pn ; Pn ; Pn ; Pn ; Pn ; Pn ; Pn ; Pn ; Ps w

Defines a window for formatted screen. The first parameter is the window id, the second parameter is the display row, the third parameter is the display col, the fourth parameter is the display width, the fifth parameter is the display depth, the sixth parameter is the offset row, the seventh parameter is the offset col, the eighth parameter is the logical width, the ninth parameter is the logical depth, the tenth on are the attributes for the window. All parameters are relative to the parent window.

Attribute flags:

- 0 - Normal (reset existing attributes)
- 1 - Bright or Bold
- 2 - Dim

- 4 - Underlined
- 5 - Slowblink (less than 150 per minute)
- 6 - Fastblink
- 7 - Reverse
- 8 - Concealed (not displayed)

DF - Define Field

`<ESC> [Pn ; Pn ; Pn ; Pn ; PS ; PS x`
Defines a field for formatted screen. The first parameter is the display row, the second parameter is the display col., the third parameter is the display width, and the fourth parameter is the display depth, the fifth parameter is the 'guarded' flag which must consist of exactly one selection from the following list, the sixth parameter on are the attributes for the field. All parameters are relative to the parent window.

Guarded flag:

- 0 - Field is enterable
- 1 - Field is guarded

Attribute flags:

- 0 - Normal (reset existing attributes)
- 1 - Bright or Bold
- 2 - Dim
- 4 - Underlined
- 5 - Slowblink (less than 150 per minute)
- 6 - Fastblink
- 7 - Reverse
- 8 - Concealed (not displayed)

Appendix B

Sample link Command Descriptions

LINK FOR 'C' TEST PROGRAM

```
$ if p1 .eqs. "" then p1 = "VT100"  
$ if p2 .eqs. "" then p2 = "nomap"  
$ link/exe=[]test test/'p2',-  
iiss_prod:[xxx.ui.safp]safp/lib,-  
[xxx.ui.fp]fp/lib/inc=prnfld,-  
[xxx.ui.test]stubs/lib,-  
[xxx.ui.driver]driver/lib/inc=(prwnd,'p1'),-  
[xxx.ui.clib]clib/lib,-  
sys$unix:[lib]crt4,libc4sect/options,libc4o/lib,libc4t/lib
```

LINK FOR 'COBOL' TEST PROGRAM

```
$ if p1 .eqs. "" then p1 = "VT100"  
$ if p2 .eqs. "" then p2 = "nomap"  
$ link/exe=[]test test/'p2',-  
iiss_prod:[xxx.ui.safp]safp/lib,-  
[xxx.ui.fp]fp/lib/inc=prnfld,-  
[xxx.ui.test]stubs/lib,-  
[xxx.ui.driver]driver/lib/inc=(prwnd,'p1'),-  
[xxx.ui.clib]clib/lib/inc=(crt4a),-  
sys$unix:[lib]libc4sect/options,libc4o/lib,libc4t/lib
```

NOTE: Replace 'xxx' with the IISS test directory.

END

8-87

DTIC