

AD A331 239

LEVEL #

0

COMPUTER GENERATED ACQUISITION DOCUMENTS SYSTEM

(CGADS)

TECHNICAL DOCUMENTATION

DTIC
ELECTE
S FEB 19 1982 D

January, 1982

A

Prepared for

Submitted by

United States Air Force
Electronic Systems Division
Air Force Systems Command
Hanscom Air Force Base
Massachusetts, 01731

Booz, Allen & Hamilton, Inc.
Suite 217, Civil Air Terminal
Bedford, Massachusetts 01730

DTIC FILE COPY

82 02 19 042

REPORT DOCUMENTATION PAGE	1. REPORT NO. DOD/DF-82/003a	2.	3. Recipient's Accession No.
4. Title and Subtitle COMPUTER GENERATED ACQUISITION DOCUMENTS SYSTEM (CGADS), Technical Documentation		5. Report Date January 1982	
7. Author(s)		6. 8. Performing Organization Rept. No.	
9. Performing Organization Name and Address Ecoz, Allen & Hamilton, Inc. Suite 217 Civil Air Terminal Bedford, MA 01730		10. Project/Task/Work Unit No. 11. Contract(C) or Grant(G) No. (C) (G)	
12. Sponsoring Organization Name and Address United State Air Force Electronic Systems Division Air Force Systems Command Hanscom AFB, MA 01731		13. Type of Report & Period Covered 14.	
15. Supplementary Notes For magnetic tape, see 17-4-288			
16. Abstract (Limit: 200 words) The Computer Generated Acquisition Documents System is a computer based system designed to automate the generation of Air Force acquisition documents and is intended to standardize their contents. The system has two modes of operation, file creation and update, and document production. This document describes the CGADS system.			
17. Document Analysis a. Descriptors Document production Data base update b. Identifiers/Open-Ended Terms c. COSATI Field/Group			
18. Availability Statement:	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 46	
	20. Security Class (This Page) UNCLASSIFIED	22. Price	

This document describes the Computer Generated Acquisition Documents System (CGADS).

1.0 System Description

CGADS is a computer based system designed to automate the generation of Air Force acquisition documents and is intended to standardize their contents.

The system has two modes of operation, file creation and update, and document production.

In the file creation and update mode, information relative to an acquisition document is interactively entered into the system and stored on a series of interrelated files. The information entered is the information which will be used to produce a document.

The system guides the data entry process by displaying a series of menus which offer the operator a number of options in selecting a function to be performed. After a function has been selected, a more detailed level of guidance is offered in the form of a purposefully ordered arrangement of questions, the answers to which prompt the system to accept the information which updates its files. In this way the system provides on-line update capability to the data bases which are used to store the information which ultimately produces a document.

The document production mode is also an interactive process. In it, a user interactively interfaces with the system to produce a document. A user is asked to enter the system by establishing an identity and by designating a document to be produced. As in the update mode, the system guides the user in developing a document by displaying a series of menus from which an option is selected. The user is then offered the opportunity to complete a series of tasks, i.e., to respond to a series of functionally related questions, within a user selected functional area. The user's responses to the questions enable the system to retrieve the appropriate information from its data bases to produce a document.

2.0 Definitions

- PRODUCT** a computer generated acquisition document.
- DRAFT PRODUCT** a product generated by the system is referred to as a draft product when some of the information required to produce the finished product is not available from the system's data bases. In those cases a message stating that tailoring is required by the Program Office will be substituted. As time goes on it is intended that most if not all of the data to complete a product will be maintained by the CGAD System.
- AREA** an area refers to a grouping of related tasks. In developing a product certain functional areas need to be addressed. For example, the areas of engineering, configuration and data, program management, logistics, packaging and transportation are addressed to complete a Statement of Work for the Production Phase. The tasks to be performed in generating a product are grouped within these areas.
- TASK** a task refers to a group of functionally related questions. For example, the task, Logistics Support Analysis (LSA) in the area of Preoperational Maintenance generates the question, "Has LSA been developed previously during the validation and full scale development phases?". The system provides a basic outline of an acquisition document. By answering the questions, the user specifies what data is to be included in the product.

The answers enable the system to retrieve information from its data bases to complete the outline.

- QUESTIONS** questions refer to that part of the system which elicits responses or input from the user in order to ascertain or determine what information will be output as part of a product. Answering questions is the work the user does to complete a task, i.e. part of a document. The system prompts the user with questions. The user's answers permit the production of information from the system to develop the sections, paragraphs, and text that go into a document.
- NOTES TO TERMINAL** messages output to the terminal after a question has been answered, which provide information to suggest that the question may have been answered incorrectly.
- DOCUMENTS** mil standards, mil specs, mil handbooks, technical manuals and other regulatory or reference material used in the product.
- HEADERS** document references and titles to be included in a product.
- PARTS** parts refer to the section numbers, paragraph numbers, appendices, etc. of a document (document parts) which are to be included as part of a product.
- ESSENTIAL ENTRIES** entries which must be included as part of a product.

OPR office of Primary Responsibility for a task.

LIBRARY refers to a group of data stored on the
system's data bases and organized in linked
lists.

3.0 File Descriptions

File Names:

DOCFILE

USERS

TSKnn

INDnn

STRnn

HEADERS

STDFILE

PRINTOUT

WPnnn

3.1 DOCFILE

File type: direct access

Record length: 256 characters

The DOCFILE contains product titles and pointers to related information. It has 99 records to accommodate up to 99 products. Record number 100 is used to store a count of the number of products currently in the system.

3.1.1 Records 1 to 99 (nn represents record number)

<u>position</u>	<u>description</u>
1-70	the product title.
71-78	a pointer to the product OPR in the STRnn file. The value of the pointer is the number of the record pointed to in the STRnn file.

- 79-86 a pointer to a record in the TSKnn file which has pointers to the product's first task. The value of the pointer is the number of the record pointed to in the TSKnn file.
- 87-94 a pointer to the last record in the TSKnn file. The value of the pointer is the number of the record pointed to in the TSKnn file. It is also a count of the number of records in the TSKnn file.
- 95-102 a pointer to the last record in the STRnn file. The value of the pointer is the number of the record pointed to in the STRnn file. It is also a count of the number of records in the STRnn file.
- 103-110 a pointer to the last record in the INDnn file. The value of the pointer is the number of the record pointed to in the INDnn file. It is also a count of the number of records in the INDnn file.
- 111-118 a count of the total number of questions that have been input for the product. Deletions are also a part of the total. Used to determine the number of records that are allocated in the USERS file to store answers to the questions.
- 119-126 a date. The date of the last modification to a product task.

example:

1	70

Statement of Work for the Conceptual Phase

71	78	79	86	87	94	95	102	103	110
	1179		24		29		1181		862
	118	119	126						
	80	12/03/81							

3.1.2 Record number 100

<u>position</u>	<u>description</u>
-----------------	--------------------

1-8 a count of the number of products currently in the system.

9-16 a pointer to the last record of the last entry in the HEADERS file. The value of the pointer is the number of the record pointed to in the HEADERS file.

example:

8	16
6	1760

3.1.3 Data Structures

Program references: records 1 to 99

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1-70	TTL	A70
71-78	DOC_OPR	I8
79-86	FST_TSK	I8
87-94	LST_TSK	I8
95-102	LST_STR	I8
103-110	LST_STR_IND	I8
111-118	LST-ANS	I8
119-126	DAT	A8

Program references: record 100

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1-8	LST_DOC	I8
9-16	LST-HDR	I8

3.2 USERS

File type: direct access

Record length: 100 characters

This file is used to store information about users of the system. The file contains three record types. There is one to one relationship between a USER'S file record and a product being worked on.

3.2.1 Record 1

<u>position</u>	<u>description</u>
1-8	a count of the total number of users in the system, i.e. the user number assigned to the last user entered in the system.
9-16	the record number of the last record assigned to a user in the USERS files

example

1	8	9	16
	569		1438

3.2.2 Records 2 TO 201

<u>position</u>	<u>description</u>
1-40	user information - name, office, phone number.
41-70	user's program name.
71-78	a date - the date the user entered the user record into the system.
79-86	a pointer to a record in the USERS file where the answers (to system questions) entered by a particular user begin. The value of the pointer is the record number of the first record in the USERS file where that user's answers are stored.
87-89	the number of records allocated for a user's answers to questions for a related product.
90-92	the identification number of the product being worked on by a user. This number also serves as a pointer to the DOCFILE record which contains the product title and related information. The value of the pointer is the number of the record pointed to in the DOCFILE.

example:

1				40	41		70
SKIP	ROEDER		TOH	2926		IMPACT	
71	78	79	86	89	92		
11/23/81			203	2	1		

3.2.3 Records 202 to end

This record contains an array of one hundred 1 character fields which store the Y, N, or U answers to the system questions.

3.2.4 Data Structures

Program references: record 1

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1-8	LST_USR	I8
9-16	LST_USR_REC	I8

Program references: records 2 to 201

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1-70	USR_INF	A70
71-78	USR_DAT	A8
79-86	USR_ANS	I8
87-89	USR_REC	I3
90-92	DOC	I3

Program references: REC_NBR

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1-100	ANS_REC	A100

3.3 TSKnn

File type: direct access

Record length: 64 Characters

The system creates one TSKnn file for each product, where nn equals the product number assigned to the product by the system. For example, the TSKnn file for product 1, Statement of Work for the Conceptual Phase, is TSK01.

The TSKnn files contain information and pointers to the different tasks which are to be completed within a given area for a product.

Tasks are in the form of questions to be answered by the user after an area has been selected. Tasks are broken down into areas of work to be completed for each product, such as, for product 1, the areas are: 1-Engineering, 2-Configuration, 3-Program Management, 4-Logistics, 5-Packaging and Transportation.

<u>position</u>	<u>description</u>
1-8	a pointer to the task title in the STRnn file. The value of the pointer is the record number of the STRnn file containing the task title.
9-16	the task's area number
17-24	the task number within the area.
25-32	a pointer to the OPR information in the HEADERS file. The value of the pointer is the number of the record pointed to in the HEADERS file.

33-40

a pointer to a record in the INDnn file which contains a pointer to a record in the STRnn file which contains the first question to be answered for a selected task. The value of the pointer is the number of the record pointed to in the INDnn file.

41-48

a pointer to the next task in the TSKnn file. The value of the pointer is the number of the record in the TSKnn file of the next task to be performed. If zero, there are no more tasks.

49-56

a date. The date on which the task was last modified.

example:

1	8	9	16	17	24	25	32	33	40	41	48	49	56
	1		1		3		61		1		17		10/08/81

3.3.1 Data Structures

Program references: REC_NBR, LST

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1-8	TSK_TTL	I8
9-16	TSK_ARA	I8
	AREA	
17-24	TSK_NBR	I8
25-32	TSK_OPR	I8
33-40	TSK_FST_ELE	I8
41-48	NXT_TSK	I8
	NXT	
49-56	TSK_DAT	A8
57	TSK_FIN	A1

3.4 INDnn

File type: direct access

Record length: 20 characters

The system creates one INDnn file for each product, where nn equals the product number assigned to the product by the system.

The INDnn files contain pointers to the questions located in the STRnn files and indicate a series of actions to be taken depending on the answers to the questions.

The file contains several record types. A record is identified by a unique character in position 1. The record types are: 1, A, 2, 3, 4, 5, 6, 7, M, S, H, O, and P.

3.4.1 Record Types

3.4.1.1 Record type "1"

<u>position</u>	<u>description</u>
1	value equals 1. Indicates that this record points to a question.
<u>position</u>	<u>description</u>
2-9	a pointer to a question in the STRnn file. The value of the pointer is the number of the first of a series of records in the STRnn file which contain a question.
10-17	A pointer to the next record in the INDnn file which contains a pointer to the next question in the same task. The value of the pointer is the number of the record pointed to in the INDnn file. If the pointer equals 0, there are no more questions in this task.

1	2	9	10	17
1		56		31

3.4.1.2 Record type "A"

position

description

1 value equals A. Indicates that this record contains answer information.

2-9 a pointer to a position in an entry in the USERS file which stores answers to the questions. The value of the pointer is a position in an array of answers.

10-17 unused.

example:

1	2	9	10	17
A		1		0

3.4.1.3 Record types "2", "3", and "4"

<u>position</u>	<u>description</u>
1	value equals 2, 3 or 4. Indicates that this record contains information for an answer--Y, N, or U respectively.
2-9	this item tells the program where to go next in a sequence of questions, referred to as a task. If the value is positive it indicates that a number of questions are to be skipped in the task. The value specifies the number of questions to be skipped. If the value is 0, no questions are skipped; if -1, the program does not go on to another question, but exits the task.
10-17	a pointer to a record in the INDnn file which contains a pointer to the first string of information for an answer. The value of the pointer is the number of the record pointed to in the INDnn file. If the pointer equals 0, it means that the system has no more information to output in response to the answer given.

example:

1	2	9	10	17
2		2		46
3		0		98
4		-1		29

3.4.1.4 Record types "5", "6", and "7"

<u>position</u>	<u>description</u>
1	value equals 5, 6, or 7. Indicates that this record points to an essential entry, an action message, or notes to terminal respectively.
2-9	a pointer to the essential entry, action message, or notes to terminal in the STRnn file. The value of the pointer is the number of the record pointed to in the STRnn file.
10-17	a pointer to the next record in the INDnn file which contains a pointer to the next string of information generated by a particular answer. The value of the pointer is the number of the record pointed to in the INDnn file. If the pointer equals 0, there is no more information.

example:

1	2	9	10	17
5		48		96
6		185		84
7		250		0

3.4.1.5 Record types "M", "S", "H", and "O"

<u>position</u>	<u>description</u>
1	value equals M, S, H, or O. Indicates that this record points to a mil standard, a mil spec, a mil handbook, or other documents respectively.
2-9	a pointer to the header records for a mil standard, mil spec, mil handbook, or other documents in the HEADERS file. The value of the pointer is the number of the record pointed to in the HEADERS file.
10-17	same as the item in positions 10-17 of the record type described in paragraph 3.4.1.4.

example:

1	2	9	10	17
M		41		38
S		161		57
H		81		199
O		541		302

3.4.1.6 Record type "P"

<u>position</u>	<u>description</u>
1	value equals P. Indicates that this record points to a document part.
2-9	a pointer to a record in the STDFILE containing a document part. The value of the pointer is the number of the record pointed to in the STDFILE.
10-17	same as the item in positions 10-17 of the record type described in paragraph 3.4.1.4.

example:

1	2	9	10	17
P		78		156

3.4.2 Data Structures

Program references: NXT, LST_STR_IND (DOC), LST, REC_NBR,
NXT_QUS(A),

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1	TYP	A1
1	ANS	A1
2-9	ADD	I8
2-9	QUS_SKP	I8
10-17	NXT_IND	I8
10-17	NXT_QUS	I8
10-17	NXT	I8
10-17	NXT_SAV	I8

Program reference: QUS_ADD+1

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1	TYP	A1
2-9	ANS-ADD	I8

3.5 STRnn

File type: direct access

Record length: 71 characters

The system creates one STRnn file for each product, where nn equals the product number assigned to the product by the system.

The STRnn file records contain strings of text which are to be included in the draft product such as tasks, titles, questions, cautions, essential entries, action messages, and notes to the terminal. The STRnn files also contain product OPR information.

positiondescription

1-70

a string of text as described above. An entry may require one or more records to complete a string of text. The last record of an entry contains "~" in record position 71 to signify the last record of the string.

71

value is blank or "~".

example:

1

70 71

|

| |

SOW: COORDINATE WITH SYSTEM ENGINEERING AND HAVE THE FINDINGS OF LSA BE AN INTEGRAL PART OF THE TECHNICAL REVIEWS. INSURE THAT LSA IS AN AGENDA ITEM ON TECHNICAL REVIEWS.

CDRL: DATA IS STILL TO BE GENERATED, REVISED, AND ACTED UPON DUE TO DESIGN CHANGES. DI-L-30316 APPLIES.

~

3.5.1 Data Structure

Program reference: REC_NUM

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1-70	STR	A70
71		A1

Program reference" REC_NUM, REC_NBR, TSK_TTL

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1-71	STR	
1-71	STR (MAX_LIN)	
1-71	STR (ILIN)	

3.6 HEADERS

File type: direct access

Record length: 73 characters

The HEADERS file contains five types of headers to be included in the draft product. They are document headers and OPR information. The document headers are either mil standards, mil specs, mil handbooks, or other documents. The header records are located via a pointer in the TSKnn file.

<u>position</u>	<u>description</u>
1-73	text containing the header information, as described above. Twenty records are reserved for each entry, whether or not they are used. The end of text is indicated by "~" in position 73 of the last record of the entry.

examples:

a document header:

1		73
MIL-STD-1528	Production Management	
1 Aug 72		~

an OPR header:

1		73
IGERALD ZAHN	ESD/TOMM	HANSCOM AFB X 13593
		AV-478-4903
		(617)-861-4903
		~

3.6.1 Data Structures

Program references: REC_NUM

<u>position</u>	<u>item name</u>	<u>Fortran format</u>
1-73	STR(MAX_LIN)	
1-73	STR(ILIN)	
1-73	HDR_LIN	
1-73	STD_STR	

3.7 STDFILE

File type: direct access

Record length: 32 characters

The STDFILE file contains pointers to document header records in the HEADERS file, and pointers to document parts contained within the STDFILE.

A record type is identified by a unique character in record position 1 which represents the type of data the record points to. The record identifiers are:

- M - mil standard headers
- S - mil spec headers
- H - mil handbook headers
- O - other document headers
- R - OPR information
- Y - not used
- P - document parts

The identifier identifies a record which points to a record which is a member in a library of data. A library of data is a group of data organized in a linked list. One of the lists (the parts library) is on the STDFILE. The others are on the HEADERS file.

3.7.1 Record numbers 1 to 7

The first seven records of the file contain pointers to the first record in the STDFILE which points to a library.

<u>position</u>	<u>description</u>
1	value is M, S, H, O, R, Y, or P. An identifier which indicates the record type.
2-9	a pointer to the first record of the identifier type in the STDFILE. The value of the pointer is the number of the record pointed to in the STDFILE.
10-17	Same as above.

example:

1	2	9	10	17
M		189		189

3.7.2 Record number 8

This record maintains a count of the number of records in the file. It is identified by an "F" in record position 1.

<u>position</u>	<u>description</u>
1	value is F.
2-9	a count of the number of records in the STDFILE file.
10-17	same as above

3.7.3 Record types "M", "S", "H", "O", "R", and "Y"

<u>position</u>	<u>description</u>
1	value is M, S, H, O, R, or Y. An identifier which indicates the type of header pointed to.
2-9	a pointer to the identified header in the HEADERS file. The value of the pointer is the number of the record pointed to in the HEADERS file.
10-17	a pointer to the next record of the same type in the STDFILE. The value of the pointer is the number of the record pointed to in the STDFILE.
18-25	a key/unique identifier which allows the system to directly locate an entry in the HEADERS file. For M, S, H, or O type records this identifier is extracted from the document header. For example, MIL STD 1552 would be assigned the value 1552.00 as an entry identifier. For O type records (other documents) some of which do not have an identifying number, a number is assigned. For R type records (OPR headers) the identifier represents the area and task number to which it applies. For example,

1.13 means area 1, task 13.

example:

1	2	9	10	17	18	25
M		321		162		1521.00

3.7.4 Record Type "P"

position

description

1	value is P. Identifies a document parts pointer.
2-9	a pointer to a document parts record. The value of the pointer is the record number of the STDFILE which contains the document part.
10-17	a pointer to the next "P" record. The value of the pointer is the record number of the STDFILE which contains the next "P" record.

example:

1	2	9	10	17
P		103		274

3.7.5 Document part record

position

description

1-32	this item contains the text referred to as a document part.
------	---

example:

1

32

|

|

3.5.1.2

3.8 PRINTOUT

File type: sequential

Record length: 81 characters

The PRINTOUT file contains output from the CREATE program with Fortran carriage controls which is to be output to the printer. It is used in the file creation and update program only.

3.9 WPnnn

File type: sequential

Record length: 71 characters

The system creates one WPnnn file for each user, where nnn equals the user number assigned to the user by the system.

The WPnnn files are used by the document production program to store information which is to be output to the printer.

4.0 System Modules

CGADS has two modules: the programs CREATE and AUTO. They are stored on the files CREATE, AUTO, SYSSUB, CRESUB and AUTSUB. The files are described below, in sections 4-1 through 4-5, respectively.

4.1 CREATE

This file contains the program CREATE which creates and updates the CGADS data base files. The files are created and updated on-line interactively via the CREATE program. Data is entered in response to a series of questions from the program.

Program CREATE calls on subroutines and functions which reside in the files SYSSUB and CRESUB. SYSSUB and CRESUB are described in subsections 4.3 and 4.4, respectively.

4.1.1 Input/Output Files

Seven files are used by the program. They are both created and updated by the program CREATE. Subsequent to creation, information relative to an acquisition document is entered into the system and stored on these files.

The files are described in section 3.0; they are listed below:

DOCFILE
TSKnn
INDnn
STRnn
STDFILE
HEADERS
USERS

In addition, an eighth output file named PRINTOUT is produced. Formatted output is stored on it. This output can consist of a listing of document headers, a listing of document parts, and/or a task description.

4.1.2 Compilation, Loading and Execution

Compilation, loading and execution is accomplished by including the files CREATE, SYSSIB and CRESUB in the load stream, together with system libraries as needed.

As of this writing, the load stream is in the file C.CREATE which contains:

```
F77 CREATE
SEG
VL #CREATE
LO B_CREATE
LO B_CRESUB
LI VAPPLB
LI
SA
QU
```

```
CO-TTY
```

The command "SEG #CREATE" will execute the program.

4.1.3 User Input

In order to place data in the files, CREATE requires input from a user. The user must execute CREATE and then follow the instructions which are displayed by the program. Data thus entered updates the CGADS files. If desired, the user can obtain a printout of the results of the interactive session.

4.1.4 CREATE Overview

Program CREATE begins with the screen presentation of a "product" menu. A product must be selected or added. If the select option is chosen, a list of current products (from file DOCFILE) is printed and a product must be chosen by entering its product number.

If the add option is chosen, the title of the new product must be entered and the system assigns a product number to it. The title is stored in file DOCFILE.

CREATE then returns the product menu to the screen. Several choices are available: to modify the product title or OPR; to examine and select from the document header library or from the document part library; to print a product description; to delete users; to quit; to access the task menu.

Of the foregoing choices, only one permits entry of task data: accessing the task menu. The task menu presents several options. The important options in terms of data entry are to add a task and to modify a task.

In order to add or to modify a task, the area number and the task number must be entered. A search of the file TSKnn for the area and task number is made.

If a task is to be added, the search must ascertain that a record with the area and task numbers to be added does not exist in the TSKnn file. The user is then instructed to enter the task title; the title is stored in the STRnn file. A new record is written in the TSKnn file, and TSKnn pointers are updated as appropriate. Then a selection to "modify" (in this instance, to supply data where none existed) that particular task must be made.

Adding and modifying now follow the same program sequence. A search of TSKnn for a record with the area and task numbers is made. The title of the task is printed on the screen from the STRnn file; the OPR information, from the HEADERS file. The title may be changed; the OPR may be changed, or added if non-existent.

Then questions may be inserted, deleted, or edited. The STRnn file, where questions are stored, and the INDnn file, where the pointer to STRnn is kept, are updated accordingly.

In addition, the sequence of actions determined by each possible response (yes, no, or undecided) to each task question may be modified. The sequence is controlled by chaining pointers in the INDnn file; changes in the sequence are reflected in changes in the INDnn file.

At the close of an update session, the user may obtain a printout of the task description(s) which were updated during the session.

4.2 AUTO

This file contains the program AUTO which produces a computer generated acquisition document. The program uses a series of interrelated data files as a data base. Information from the data base is input to the program in the process of producing a product.

Program AUTO calls on subroutines and functions which reside in the files SYSSUB and AUTSUB. SYSSUB and AUTSUB are described in subsections 4.3 and 4.5, respectively.

4.2.1 Input Files

Seven files are used for input. They comprise the data base, and are created in program CREATE. They are listed below, and are described in section 3.0:

USERS
DOCFILE
TSKnn
INDnn
STRnn
HEADERS
STDFILE

4.2.2 Output Files

Information for the output is collected and stored on a temporary scratch file called SCRnnn, where nnn corresponds to the users number. The user identification number is assigned by the system, and is used to identify a user's logical record, i.e., the user's piece of work. When required for the output, the appropriate information is read from SCRnnn and written to file WPnnn, which stores the formatted output of the product.

4.2.3 Compilation Loading and Execution

Compilation, loading and execution is accomplished by including the files AUTO, SYSSUB and AUTSUB in the load stream, together with system libraries as needed.

As of this writing, the load stream is in the file C.AUTO which contains:

```
F77 AUTO
SEG
VL #CGADS
LO B_AUTO
LO B_SYSSUB
LO B_AUTSUB
LI VAPPLB
LI
SA
QU

CO -TTY
```

The command "SEG #CGADS" will execute the program.

4.2.4 User Input

In order to produce an acquisition document, AUTO requires input from a user. The user must execute the program and then follow the instructions displayed by the program. If properly executed, this procedure results in a printout of the generated product. The printout is either an acquisition document or a draft of an acquisition document.

4.2.5 AUTO Overview

Once the user begins the program, initial information is displayed on the terminal screen. The user then can initiate an identifying action whereby the system will assign a user number or the user must select a user number previously entered. If the user elects to enter a new identification, the number of the document desired must be chosen from a list of current products. The list of current products is printed from the DOCFILE file.

User information exists in the USERS file. A new user identification is written to that file.

If the user does not choose to enter a new identification, a list of users from the USERS file is displayed at the terminal. In thus choosing a previously entered user number, the current user is also implicitly choosing a product.

This product, associated with the user and a product ID number, controls transfer to a specific TSK file--TSKnn, where nn is the product number. Transfer is made to the record in TSKnn containing information for the first task in the product's first area.

AUTO next presents the user with the opportunity to "run functional tasks." A listing of product areas is displayed, and the user must select one area.

A search is made in TSKnn until the selected area is found. Task titles for that area are read from the STRnn file and listed by number and title at the terminal.

The user may then choose to "accomplish a task" in the selected area by entering a task number.

AUTO next searches the INDnn file for the first record containing information for the chosen task. The first such record always points to a question in the STRnn file. That question is displayed at the terminal as "QUESTION # 1."

The user's response to the question may be "Y" (yes), "N" (no), or "U" (undecided). The response initiates a sequence of information retrievals which is controlled by chaining pointers in the INDnn file.

The information retrieved is displayed on the screen and saved in the SCRnnn file in preparation for the printout of the product. Such information can be one or more of the following: essential entries; action messages; document headers; document parts; task OPR information.

AUTO retrieves essential entries and action messages from the STRnn file; document headers and task OPR information from the HEADERS file; document parts from the STDFILE.

After response to the first question is complete, the user may continue to retrieve and answer the remaining task questions, if there are any. When the task is complete, the user has these choices: to return to the task menu and choose another task to accomplish; to quit with no printout or word processor file; to exit with 1-9 printouts (as user specifies) and/or a word processor file of accomplished tasks. Word processor files are files intended to be transferred to word processors.

4.3 SYSSUB

The file SYSSUB contains subroutines and a single function required by both the CREATE and AUTO programs.

The subroutines are:

ASK
CONTIN
HDR_70
PRT_STR
RD_TSK
WRT_STR

The function is:

NUM_CON

4.4 CRESUB

The file CRESUB contains subroutines and a single function required by the CREATE program.

The subroutines are:

ADD_STR
ADD-HDR
PRT_FND
QUS_MOD
QUS_WRT
STD_FND
QUS_RES
RES_WRT

The function is:

FLT_CON

4.5 AUTSUB

The file AUTSUB contains subroutines required by the program AUTO.

The subroutines are:

ANS_GET
ANS_PUT
ESS_WRT
ACT_WRT
QUS_RUN
STD_APP
STD_WRT
STD_INS
STD_TSK
PRT_INS