

**DATA ADMINISTRATION
EXTENDED OPERATING SYSTEM
3A PROCESSOR**

CONTENTS	PAGE	CONTENTS	PAGE
1. GENERAL	1	Figures	
2. OVERVIEW	2	1. Input Message Flow	8
A. Terminal Administration	2	2. Output Message Flow	9
B. Data Administration	3	3. Layout of Input Message Supplementary Information	9
C. Structure of the I/O Data Administration Package	3	Tables	
3. TERMINAL ADMINISTRATION PACKAGE	3	A. Acronyms and Abbreviations	10
DATA BASE	3	B. Input/Output Data Administration Package Programs	10
A. Input Catalogs	4	C. Input/Output Message Macros	11
B. Dictionary Tables	4	1. GENERAL	
STRUCTURE	4	1.01 ♦ This section describes the data and terminal administration package of the Extended Operating System (EOS).	
A. DADMAD	4		
B. Analyzers	4		
C. Translators	5		
CLIENT INTERFACE	5		
A. DADMAD Interface	6		
B. Old TTY Package Interface	6		
C. Common Systems Interface	7		
4. GLOSSARY	7		



This section is in effect for Generic Release G2A and all subsequent generic releases. Section 254-340-054, Terminal Administration, is in effect for Generic Releases prior to G2A only.

1.02 This section is reissued to include information regarding message classes and to advise which Generic Releases this section covers. Revision arrows are used to emphasize the more significant

NOTICE

Not for use or disclosure outside the
Bell System except under written agreement

SECTION 254-340-040

changes. Additional reasons for reissue include the following:

- To add a table of acronyms and abbreviations
- To expand the table of program listings
- To place the figures and tables at the end of the text.¶

1.03 Using the principle of data administration, a flexible terminal administration package was developed for the distribution of input/output (I/O) messages between devices and EOS or application tasks (referred to as clients). The package is designed so that I/O messages can be distributed to multiple devices. The package also provides centralized routines for the formatting and parsing of messages to reduce the complexity of client interfaces to the package.

1.04 The terminal administration package supports terminals connected to the 3A Processor through a teletypewriter controller (TTYC) or an RS232C interface. However, the package is not limited to TTY messages.

1.05 The terminal administration package is an improvement over a previous package that did not utilize data administration. The previous package will be referred to as the old TTY package. The old TTY package was designed to handle messages associated with EOS programs as well as messages associated with a class of programs referred to as common system programs that were written for the 2B Processor and No. 3 ESS. The ability to handle messages from common systems programs is necessary because of the integration of some of these programs into EOS. This terminal administration package is designed to support three client-terminal interfaces. They will be referred to as the old TTY package interface, the common systems interface, and the data administration interface.

1.06 The contents of this section deal primarily with the new terminal administration package. The old TTY package is discussed only in regard to the way programs that used the old TTY package interface with the new package. A more detailed description of the old TTY package is contained in Section 245-340-054.

1.07 The following sections contain background information related to terminal administration.

SECTION	TITLE
254-340-001	Overview, Extended Operating System, 3A Processor
254-340-030	Processor/Process Management, Creation, Event and Communication Control, Extended Operating System, 3A Processor
254-340-052	Device Handlers, Extended Operating System, 3A Processor
254-340-054	Terminal Administration, Extended Operating System, 3A Processor
254-340-062	File System, Extended Operating System, 3A Processor
254-340-106	Macros and Glossary, Extended Operating System, 3A Processor

1.08 The following programs provide EOS facilities used in implementing terminal administration.

- (a) The operating system tables (LOSTABS) provide the various application parameters which define the system hardware and software configuration.
- (b) The EOS maintenance program (MAINT), PR-4C607, provides EOS task control.
- (c) The teletype application program (TTYAPP), PR-4C613, provides the TTY interface to common systems.

1.09 The acronyms and abbreviations used in this section are listed in Table A.

2. OVERVIEW

A. Terminal Administration

2.01 The structure of the EOS terminal administration package is based on the concept that a high degree of efficiency and flexibility can be achieved by having a centralized package for controlling the distribution of I/O messages between devices and EOS or the application. The terminal administration package passes messages to a client and receives

messages from a client in a highly encoded format that relieves the client of the burden of manipulating American Standard Code for Information Interchange (ASCII) strings. All formatting of output messages and parsing of input messages is performed for the client by centralized routines. Also, the client receiving a message is assured that the message has passed certain basic syntactical checks, that the message was input over a proper I/O channel, and that the system is in the proper state to process the message. The software involved in the control of input messages is shown in Fig. 1. The software involved in the control of output messages is shown in Fig. 2.

B. Data Administration

2.02 The concept of data administration is based on the fact that certain I/O functions such as deciding when to go to a backup device, duplicating data on more than one device, and determining which physical device to send data to can be best performed by a centralized administrator. This concept is implemented in EOS in the form of an I/O Data Administration Package (IODAP). This package accepts data from a client and directs it to the appropriate device(s). It also reads data from a device and directs it to the appropriate client (or other devices). The reads and writes are accomplished through the EOS file system.

2.03 The IODAP is independent of the format of the data being passed between the client and device. Also, the IODAP is not limited to terminal administration. The package will support any device that runs under control of the EOS file system.

C. Structure of the I/O Data Administration Package

2.04 The IODAP consists of three parts: a device interface, a client interface, and an administrator. The EOS file system provides the device interface. On input, it accepts a buffer of data from a device and passes it to the administrator. On output, it accepts data from the administrator and passes it to a device. The administrator is implemented in the form of EOS tasks and functions as a message distributor. The program listings (PRs) that compose the IODAP are listed in Table B. On output, it accepts a buffer of data from a client and directs it to the appropriate devices using the file system. On input, it receives a buffer of data from the file system and directs it to other devices and/or

to the appropriate client. All clients of the IODAP consist of EOS or application tasks. The client interface of the IODAP is provided by a set of system I/O message macros (see Table C). These macros in turn call subroutines that communicate with the administrator via the message handling portion of the EOS interprocess communication mechanism.

2.05 In order for the IODAP package to be adaptable to different applications, application-specified software called analyzers or translators can be invoked. They provide the application with the ability to monitor and modify data sent and received by a particular device. When the administrator receives an input buffer from the file system, it passes the data to a set of input analyzers associated with the device. Also, just before the administrator passes an output buffer to the file system, output analyzers associated with the device receive control. The EOS provides the facilities for the application to define input and output translators. Each translator can be associated with a particular type of data. An input translator is called as the data is copied from the client data space into the administrator buffers.

2.06 A particular device can be assigned multiple input and output analyzers that are called in a predetermined order. However, only one translator can be called for an input or output message.

3. TERMINAL ADMINISTRATION PACKAGE

DATA BASE

3.01 Terminal administration requires definitions of TTY input messages. The definitions of EOS messages are maintained in TTYTBL which contains both a catalog and a dictionary table. The catalog and dictionary tables for the user must be maintained in USRTBL. By maintaining multiple catalogs and dictionary tables, modifications to the EOS terminal data base are transparent to the user and, likewise, modifications to the user terminal data base are transparent to EOS. The only time USRTBL must be reassembled as a result of modifications to TTYTBL is when messages or dictionary words of mutual interest are added.

SECTION 254-340-040

A. Input Catalogs

3.02 The primary purpose of a catalog is to provide information needed to locate the recipient of incoming data. A catalog entry may also contain other information such as the characteristics of the fields of a message, the class of the message, etc. Additional user catalogs can be added as the need for expansion occurs. These catalogs may include unique dictionary tables or may share existing tables.

B. Dictionary Tables

3.03 To save memory, EOS stores ASCII strings used in translating I/O messages in central locations referred to as dictionary tables. Clients can reference strings by specifying a dictionary table and the location of the string within the table. A location is referred to as a dictionary word or SPELL. The SPELLs are used in searching catalogs, screening input message fields, and passing data between the terminal package and a client. A SPELL consists of 11 bits which represent an index into the associated dictionary table. However, the bits do not specify the associated table. The linkage between a SPELL and its dictionary table is the program name of the table. Each program that uses SPELLs must specify the program name of the associated dictionary table. The SPELLs are then linked to the proper dictionary tables by the linkage editor. Only one dictionary table can be associated with an input or output message.

STRUCTURE

3.04 The major components of the EOS terminal administration package are analyzers, translators, and an administrator. The administrator is provided by EOS and is referred to as DADMAD. The EOS also provides some standard analyzers and translators. However, the terminal administration package is flexible enough that additional analyzers and translators can be provided by the application.

A. DADMAD

3.05 DADMAD corresponds to the administrator described as part of the IODAP. DADMAD routes input data from the devices to the clients and other devices. The routing of input data to other devices is referred to as input echoing. DADMAD also routes output data from the clients to the appropriate devices. When an output message

is routed to more than one device, it is referred to as message broadcasting. Each message is assigned to one of 64 message classes. Also, each TTY is assigned to one or more message classes. An output message of a specific class is routed to all TTYs associated with that specific message class. DADMAD handles all error recovery and routing of data to backup devices. DADMAD consists of the following programs:

- DADINT
- DADBRD
- DADRED
- DADSUB
- DADOUT.

3.06 Each message from a TTY type device requires a response to be printed on the terminal before any other output associated with the message can be printed. DADMAD supports this requirement.

3.07 DADMAD supports a class of I/O referred to as linked I/O in which the I/O message exceeds one line on the terminal. With linked I/O, a message is divided into segments (or lines). The segments are linked together and treated as a single message. No other I/O will appear between segments.

3.08 The device handlers for the TTYC and RS232C interface operate in half-duplex mode. This means these device handlers cannot send and receive data simultaneously. Since DADMAD interfaces with these device handlers, it is also under a similar restriction. Therefore, between segments of linked input or when a message is being input, no output is sent to the terminal. Also, between segments of linked output or when a message is being outputted, no input is accepted from the terminal.

B. Analyzers

3.09 The terminal administration package contains provisions for calling both input and output analyzers. DADMAD uses input analyzers to evaluate incoming data and to obtain information needed in routing the data. Input analyzers are also used to modify incoming data. For example,

an envelope may have been placed around the data to identify routing paths or certain characteristics of the data. This envelope may have to be removed to allow further evaluation. Consequently, there are currently two types of input analyzers:

- (a) Envelope processors
- (b) Catalog interpreters.

An envelope analyzer operates alone on incoming data; whereas, a catalog interpreter operates on an input message as dictated by the rules of an associated catalog. A catalog interpreter scans the catalog attached to it based on key field of the input data. The input data is broken down according to the grammatical rules of the catalog. Input analyzers are called only for the first line of linked input messages.

3.10 Output analyzers are called to make any necessary changes to an output buffer before it is passed to the file system. Output analyzers can be used to place an envelope around the data, change the message, or abort the message.

3.11 The EOS provides an input analyzer (ESSALZ) and an output analyzer (TTYALZ). However, the application can specify a list of input and output analyzers for each terminal. The data administrator initialization program (DADINT) builds an analyzer table of any application-specified analyzers for each terminal in the system. DADINT is scheduled for all levels of initialization. The EOS program ALZEX functions as an executive for calling the input and output analyzers. The program goes through the analyzer table for a given terminal during both input and output and calls all the input or output analyzers.

C. Translators

3.12 The structure of the terminal administration package is based on both input translators and output translators. An input translator is also referred to as a parser. Before a message is passed to a client, a parser is called. The parser breaks a message down into its constituent fields and encodes the ASCII data according to the syntactical and grammatical rules of the catalog associated with the message. The parser arranges the fields in the client buffer in the proper format. If a field is missing, a zero is stored in the field when it is placed in the buffer. The client then

only has to pick up the desired fields and do any necessary range checking. This relieves the client from having to do any syntax checking.

3.13 The EOS program PARSER is provided as a common parser for input messages. However, the application can provide its own parsers. It is the function of the input analyzers to specify the parser (or none at all).

3.14 An output translator, also referred to as an expander, relieves the client of having to format ASCII output messages. An expander converts the binary-encoded message from a client to ASCII based on an accompanying format statement. The EOS also provides an expander (EXPNDR); however, the client can specify an application-defined expander (or no expander) to be used instead.

CLIENT INTERFACE

3.15 The client interface to the terminal administration package is provided by the system macros described in Table C.

Input

3.16 A client executes the REC macro to receive an input message from a terminal. The message is parsed, if necessary, and copied into a buffer supplied by the client. In addition to the message itself, the client can request supplementary information pertaining to the message. The layout of this information is shown in Fig. 3. The client does not specify the terminal from which input is coming but only indicates that it is ready for the next line of input. DADMAD is responsible for input message routing. The routing of messages is based on message classes.

3.17 If the class of an input message is associated with more than one terminal, the message is echoed to all terminals defined as part of the message class. This is referred to as input echoing. If the message is linked, the message is not echoed until the entire message has been entered. The echoing of long, linked input messages can tie up output buffer space. In order to prevent this, a catalog feature is provided that enables input echoing to be suppressed on all terminals other than the one on which the message is being input.

3.18 During linked input, the only output sent to the terminal is the client-generated response.

SECTION 254-340-040

This, in effect, dedicates the terminal to the client and defers output from other clients to the terminal until the linked input is complete.

Output

3.19 The client uses the XMIT, XMIT_CONTROL, and XMIT_FORMAT macros to output messages to the terminals. The XMIT macro sends data to the terminal. No client control or format is necessary. If the output message needs to be expanded before it is passed to DADMAD, then the XMIT macro must specify a format table built by the XMIT_FORMAT macro. The XMIT_CONTROL macro specifies control information used by DADMAD.

3.20 The client does not specify (or know) the terminal to which a message is being sent. The client specifies only the class of the message. An output message of a particular class is sent to all terminals associated with the class of the message. This is referred to as message broadcasting. DADMAD is responsible for output message routing.

3.21 For linked output messages, the client must specify the link option on the XMIT_CONTROL macro. Linked messages are output one segment at a time. During linked output, synchronization of the client with DADMAD is accomplished via the EOS event mechanism. DADMAD sets the specified event in the client process or task when it is ready for the next segment of the message.

Responses

3.22 The client must return a response to the craft person whenever an input message is received. Also, a response must be sent to the terminal after each segment of a linked input message. There are two options available to the client for returning a response.

- (a) The response to a message can be obtained in the catalog entry for the message or
- (b) The client can generate the response.

A response contained in a catalog is limited to two characters. If the message passes all checks performed by DADMAD, then the input analyzer retrieves the response and DADMAD sends it to the terminal. If the response is not in the catalog, then the client must generate the response by using the RESPONSE macro. A client-generated response

is not limited to two characters. If the client is expected to generate a response, DADMAD will stop all activity on the terminal until the response is sent.

A. DADMAD Interface

3.23 Client (and EOS) tasks that are written for Generic Release G2A and subsequent releases interface to DADMAD directly. Parsing and expanding of messages is the responsibility of these tasks and not the responsibility of DADMAD.

3.24 When DADMAD receives an input message, it builds an input data block (IPDB). The IPDB contains all the information necessary to parse the message and pass it to the client. Then DADMAD sends the client the IPDB via event 3. The client retrieves the message by means of the REC macro. If any parsing is necessary, the REC system routine will call PARSER.

3.25 For output messages, the client passes the message to DADMAD via the XMIT macro. If the message needs to be expanded, the XMIT system routine calls EXDNDR. An output data block (OPDB) is built and sent to DADMAD.

B. Old TTY Package Interface

3.26 Client (and some EOS) tasks that are written for generic releases prior to G2A use the old TTY package. The interface to DADMAD for these programs is via the EOSMSG program. Communication between EOSMSG and DADMAD is provided by the EOS message mechanism.

3.27 When the input message is recognized as an old TTY package message by the input analyzer ESSALZ, the IPDB is sent instead to EOSMSG via event 3. The EOSMSG parses the ASCII message (via PARSER) and converts the message into binary. The message is then placed in a 16-word buffer referred to as a client input buffer (CIB). Then the message is sent to the old TTY package client via the EOS message mechanism.

3.28 For output, EOSMSG receives a 16-word message buffer, referred to as a client output buffer (COB) via the PRINT_TTY macro. EOSMSG expands the binary data into ASCII (via EXPNDR) and builds an OPDB. Then EOSMSG sends the OPDB to DADMAD via the EOS message mechanism.

3.29 A 2-character response to an input message from an old TTY package client is sent to EOSMSG via event 5. EOSMSG formats the response to correspond to DADMAD requirements. Then the response is sent to DADMAD.

C. Common Systems Interface

3.30 The terminal administration package also supports Electronic Switching System (ESS) style messages. These messages are processed in a similar manner to the old TTY package messages. The major difference is that input messages processed by EOSMSG are sent to the EOS maintenance task (MAINT). The program TTYAPP, which runs under control of MAINT, sends the message to the proper common systems client.♦

4. GLOSSARY

4.01 The following basic terms are defined in the context of their use within this section.

Application—A set of functional system programs which uses the services of EOS.

Buffer—A temporary storage area consisting of one or more words of store.

Catalog—A macro-generated table that defines input message formats.

Dictionary—A table of equivalences between ASCII character strings and binary tokens used to represent the ASCII strings.

LOSTABS—Operating system prototype tables used by the application to define system resources, configuration, parameters, etc.

Macro—A precoded sequence of instructions to which a label has been assigned. This label along with any parameters can be used as an instruction in a source language. On each call, the macro instruction is replaced by its equivalent instruction sequence.

Parse—Break a message down into its constituent fields according to the grammatical rules of an associated catalog (or format table).

Subroutine—A sequence of instructions designed to perform a specific function(s) and that can be invoked within another section of instructions.

Syntax—The rules governing message structure.

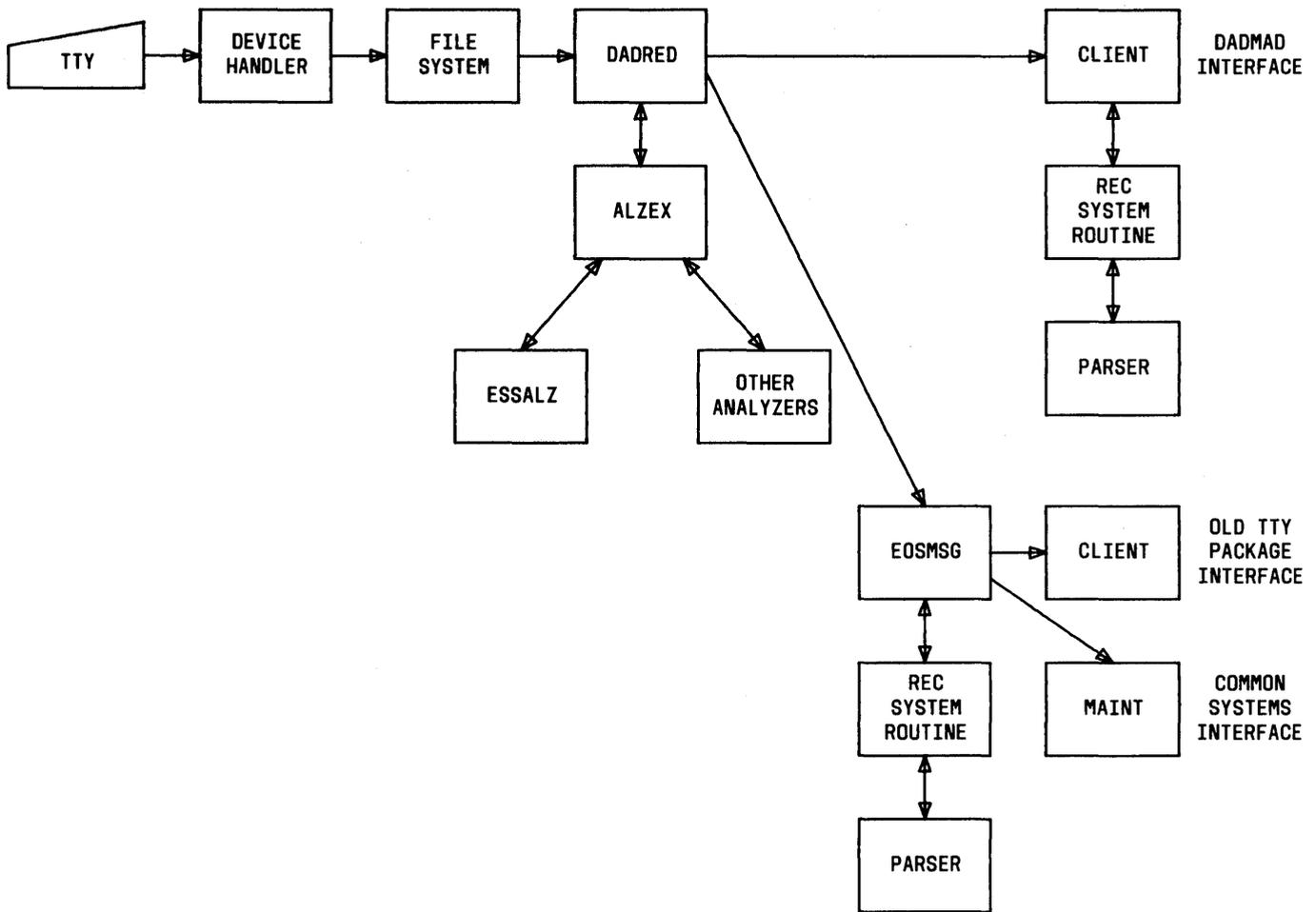


Fig. 1—Input Message Flow

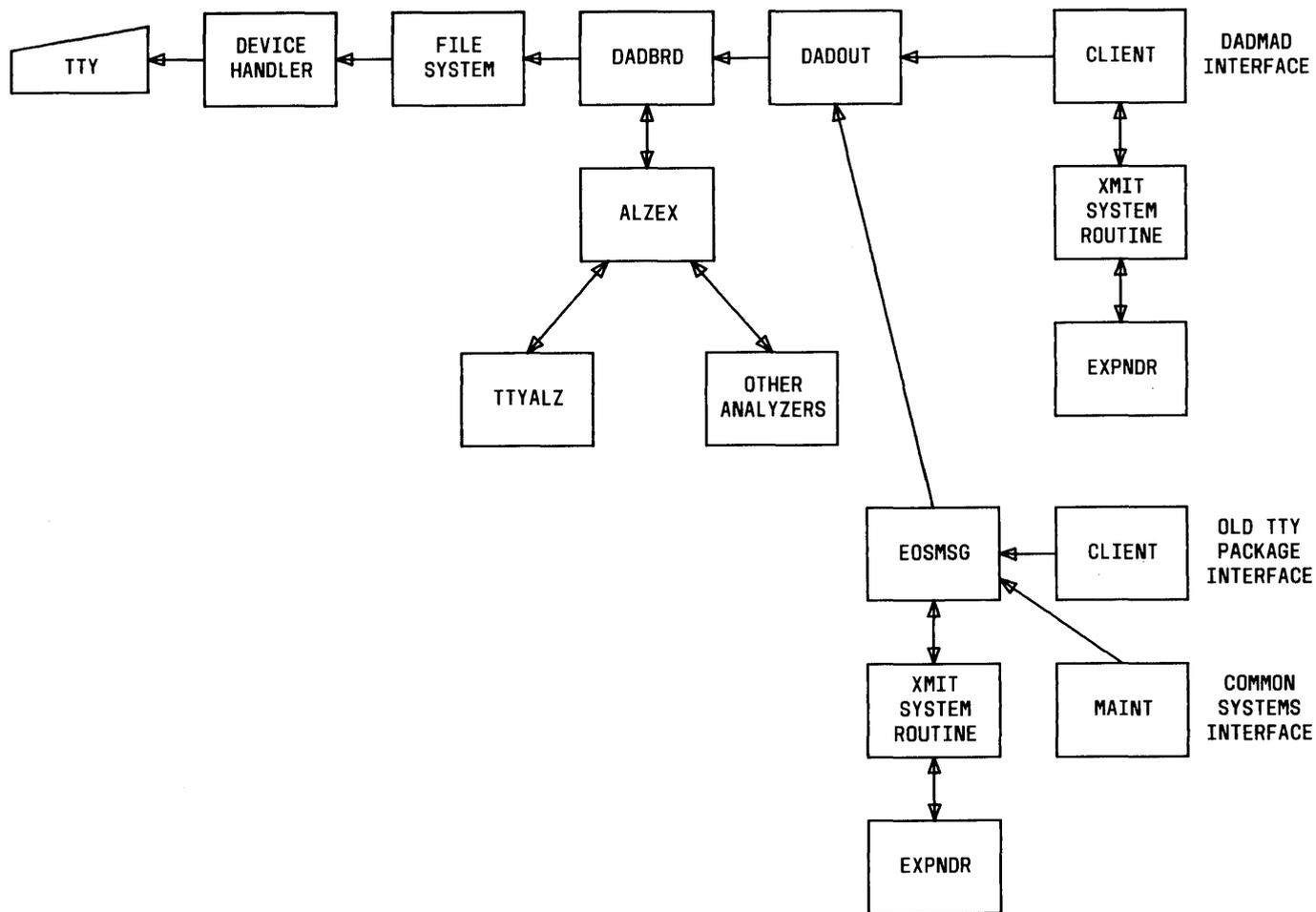
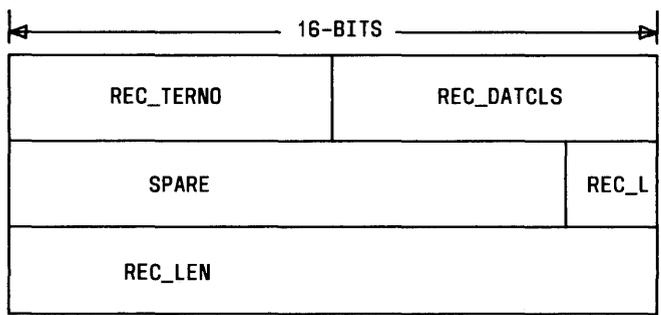


Fig. 2—Output Message Flow



REC_DATCLS - THE DATA CLASS OF THE MESSAGE
 REC_TERNO - THE TERMINAL NUMBER OF THE MESSAGE
 REC_L - THE LINK STATUS OF THE MESSAGE
 REC_LEN - THE LENGTH OF THE RECEIVED DATA (NUMBER OF WORDS)

Fig. 3—Layout of Input Message Supplementary Information

TABLE A

ACRONYMS AND ABBREVIATIONS

TERM	MEANING
ASCII	American Standard Code for Information Interchange
CIB	Client Input Buffer
COB	Client Output Buffer
DADMAD	Data Administrator
EOS	Extended Operating System
ESS	Electronic Switching System
I/O	Input/Output
IODAP	I/O Data Administration Package
IPDB	Input Data Block
OPDB	Output Data Block
PR	Program Listing
TTYC	TTY Controller

TABLE B

INPUT/OUTPUT DATA ADMINISTRATION PACKAGE PROGRAMS

NAME	NUMBER	TITLE
ALZEX	PR-4C222	Executive for Input and Output Analyzers
DADBRD	PR-4C223	Data Administrator, Writes to Terminals
DADINT	PR-4C224	Data Administrator, Initialization
DADOUT	PR-4C225	Accepts Client Output Messages
DADRED	PR-4C226	Data Administrator, Accepts Terminal Input
DADSUB	PR-4C227	Data Administrator, Common Subroutine Programs
EOSMSG	PR-4C229	Client Interface to Data Administrator
ESSALZ	PR-4C230	ESS Catalog Search Program
EXPNDR	PR-4C231	Output Message Binary to ASCII Converter
PARSER	PR-4C232	Input Message Scanner
PRSSUB	PR-4C233	Common Subroutine for TTY Package
TTYALZ	PR-4C234	Output Analyzer for TTYC Device
TTYTBL	PR-4C401	TTY Data Layout Program
USRTBL	PR-4C402	Users TTY Data Layout Program

TABLE C

I/O MESSAGE MACROS

MACRO	FUNCTION
ADD_DATCLS	Add a data class to a specified terminal.
DEL_DATCLS	Delete a data class from a specified terminal.
REC	Checks for an input message from the administrator, parses it if necessary, and copies it into the specified buffer.
RESPONSE	Generates a response to an input message. The client can either specify a canned response supported by EOS or a unique response.
XMIT	Sends data from the client buffer to the terminal(s) specified by the data class. If a format is specified, the data is passed to a translator for expansion.
XMIT_CONTROL	Specifies the priority of action and relative priority of a message, whether a full date and time line is to appear following the message, and the link status of the message.
XMIT_FORMAT	Specified format parameters to be used by the output translator in expanding the message.