

60459500

 CONTROL DATA

**TAF
VERSION 1
REFERENCE MANUAL**

**CDC® OPERATING SYSTEM:
NOS 2**

REVISION RECORD

REVISION	DESCRIPTION
<p>A (04-26-82)</p>	<p>Manual released. This manual reflects NOS 2.0 at PSR level 562. New TAF feature documented in this manual is TAF automatic recovery. Features documented in the TAF/CRM Reference Manual that are reflected in this manual are: TAF/CRM automatic recovery, TAF/CRM batch concurrency, TAF/CRM batch recovery, and TAF/CRM MIP and AK support.</p>
<p>B (01-27-83)</p>	<p>Manual updated to PSR level 580. New RMKDEF statements were added to the xxJ file for multiple index file recovery and clarifications were made of LIBTASK and recovery descriptions. Addition of the Login/Logout appendix, and revision of the Terminal Definition appendix and the Line Transmission Key appendix. This edition obsoletes all previous editions.</p>
<p>C (03-29-85)</p>	<p>Manual updated to NOS 2.4.1 at PSR level 630. This revision reflects the addition of the user statement to the TAF configuration file and of the send message task (STASK). Appendix G, Terminal Definition Commands, and appendix J, Accessing a Host, have been revised. The IBM 3270 terminal has been added to the set of supported terminals. Several technical and editorial changes have been made.</p>
<p>Publication No. 60459500</p>	

REVISION LETTERS I, O, Q, S, X AND Z ARE NOT USED.

Address comments concerning this manual to:

Control Data Corporation
 Publications and Graphics Division
 4201 North Lexington Avenue
 St. Paul, Minnesota 55112

or use Comment Sheet in the back of this manual.

© 1982, 1983, 1985
 by Control Data Corporation
 All rights reserved
 Printed in the United States of America

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Front Cover	-	2-42	B	B-9	C	J-1	C		
Inside Front Cover	C	3-1	A	B-10	C	J-2	C		
Title Page	-	4-1	C	B-11	C	J-3	C		
2	C	4-2	C	B-12	C	J-4	C		
3/4	C	4-3	C	B-13	C	J-5	C		
5	C	4-4	C	B-14	C	J-6	C		
6	C	4-4.1/4-4.2	C	B-15	C	J-7	C		
7	C	4-5	A	B-16	C	J-8	C		
8	C	4-6	B	B-17	C	J-9	C		
9	C	4-7	C	B-18	C	J-10	C		
10	C	4-8	C	B-19	C	J-11	C		
1-1	A	4-9	C	B-20	C	J-12	C		
1-2	B	5-1	A	B-21	C	J-13	C		
1-3	A	5-2	C	B-22	C	J-14	C		
1-4	C	5-3	C	B-23	C	J-15	C		
1-5	C	5-4	A	B-24	C	J-16	C		
2-1	B	6-1	C	B-25	C	J-17	C		
2-2	B	6-2	A	B-26	C	J-18	C		
2-3	B	6-3	B	B-27	C	J-19	C		
2-4	C	6-4	B	C-1	B	J-20	C		
2-5	B	6-5	A	C-2	A	Index-1	C		
2-6	B	7-1	B	C-3	B	Index-2	C		
2-7	B	7-2	B	C-4	B	Index-3	C		
2-8	A	7-3	A	C-5	C	Index-4	C		
2-9	C	7-4	B	C-6	C	Comment Sheet			
2-10	B	7-5	A	C-7	C	Back Cover	-		
2-11	B	8-1	A	C-8	C				
2-12	B	8-2	C	C-9	C				
2-13	B	8-3	C	C-10	C				
2-14	A	8-4	C	C-11	C				
2-15	A	8-5	B	C-12	C				
2-16	B	8-6	B	D-1	C				
2-17	B	9-1	C	E-1	A				
2-18	B	9-2	C	E-2	B				
2-19	C	10-1	C	F-1	A				
2-20	C	10-2	B	F-2	C				
2-21	C	10-3	B	F-3	C				
2-22	C	10-4	A	F-4	C				
2-23	C	10-5	A	F-5	C				
2-24	A	10-6	B	G-1	C				
2-25	A	10-7	A	G-2	C				
2-26	A	10-8	A	G-3	C				
2-27	A	11-1	C	G-4	C				
2-28	B	11-2	C	G-5	C				
2-29	C	12-1	A	G-6	C				
2-30	C	12-2	A	G-7	C				
2-30.1/2-30.2	C	12-3	A	G-8	C				
2-31	A	12-4	A	G-9	C				
2-32	A	12-5	C	G-10	C				
2-33	B	A-1	A	G-11	C				
2-34	B	A-2	A	G-12	C				
2-35	A	B-1	A	G-13	C				
2-36	A	B-2	C	G-14	C				
2-37	A	B-3	C	G-15	C				
2-38	B	B-4	C	G-16	C				
2-39	B	B-5	C	G-17	C				
2-40	A	B-6	C	H-1	B				
2-41	B	B-7	C	H-2	C				
		B-8	C	I-1	A				

STATE OF TEXAS

County of _____ State of Texas

No.	Name	Amount	Balance
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

PREFACE

The CONTROL DATA® Transaction Facility (TAF), Version 1.3, is a network host product. It is a network application that requires the facilities of the Network Access Method (NAM) and the 255x Series Communications Control Program (CCP). TAF runs as a subsystem of the Network Operating System (NOS), Version 2, and may be used with COMPASS, FORTRAN, or COBOL.

This manual documents the executive portion of TAF. The three data managers available for use with TAF are each documented in their respective reference manuals. These are:

- CDC® CYBER Database Control System, Version 2 (CDCS)
- CDC TAF/CRM Data Manager Version 1 (CRM)
- CDC TOTAL Data Manager Version 2

Refer to the list of related publications for full titles and publication numbers of the respective manuals.

AUDIENCE

This manual is for the data administrator and applications programmer. The data administrator (DA) is a person or a group of people responsible for installing and maintaining the transaction subsystem. This person or group sets up the data base and describes the relationships among the various elements. The DA is concerned with the internal design of the system and can modify many of the system components to suit site requirements. An applications programmer is a person who writes tasks in FORTRAN, COBOL, or COMPASS to be used by the end user (terminal operator). Using these tasks, the terminal operator can access and update the data base with little or no knowledge of TAF. New users of TAF should consult the TAF 1 User's Guide as a first source of information. It is an introductory text for inexperienced users.

The user must be familiar with basic NOS operations such as running a compilation job. The NOS 2 Reference Set, Volume 2, acquaints the new user with the fundamentals of NOS. The NOS 2 Reference Set, Volume 3, is for the high-level language applications programmer who is already familiar with NOS. The NOS 2 Reference Set, Volume 4, is for the COMPASS applications programmer.

TERMINOLOGY AND CONVENTIONS

References in this manual to the transaction subsystem imply every aspect of TAF. References to

the transaction executive imply the controlling portion of TAF, the portion described in this manual. References to FORTRAN imply either FORTRAN Extended Version 4 or FORTRAN Version 5; TS mode is not allowed (that is, the SEQ, TS, and OPT=0 parameters are not allowed on the FTN command). References to COBOL imply Version 5.

Where brackets, [], are used in task request format descriptions, they indicate optional parameters.

The term alphanumeric refers to any combination of the letters A through Z and the numbers zero through nine; special characters are not included.

Conventions for word formats are as follows:

- Crosshatching indicates a field is not used by or is not applicable to TAF.
- Reserved fields may contain information useful to TAF internal processing. The user should not expect them to contain any particular values.
- Fields labeled with mnemonics indicate that a specific parameter must be inserted (generally described after the word format).
- Fields with numeric identifiers indicate the actual value that is used or returned for a particular request.

Models 815, 825, 835, 845, and 855 of the CYBER 170 Computer Systems share many of the functional and architectural attributes of the CYBER 180 computer systems (models 810, 830, 835, 840, 845, 850, 855, and 860). The term CYBER 180-class machines describes these similar models collectively.

Extended memory for model 176 is large central memory extended (LCME). Extended memory for models 865, 875 and CYBER 180-class machines is unified extended memory (UEM). Extended memory for models 865 and 875 may also include either extended core storage (ECS) or extended semiconductor memory (ESM). Extended memory for all other NOS computer systems is either ECS or ESM. ECS and ESM are the only forms of extended memory that can be shared in a linked shared device multiframe complex and can be accessed by a distributed data path (DDP).

In this manual, extended memory refers to all forms of extended memory unless otherwise noted.

Programming information for the various forms of extended memory can be found in the COMPASS 3 Reference Manual and in the appropriate computer system hardware reference manual.

RELATED PUBLICATIONS

The following manuals contain additional information for the user.

<u>Control Data Publication</u>	<u>Publication Number</u>
COBOL Version 5 Reference Manual	60497100
Common Memory Manager Version 1 Reference Manual	60499200
COMPASS Version 3 Reference Manual	60492600
CYBER Database Control System Version 2 Data Administration Reference Manual	60485200
CYBER Database Control System Version 2 Application Programming Reference Manual	60485300
CYBER Loader Version 1 Reference Manual	60429800
FORTRAN Extended Version 4 Reference Manual	60497800
FORTRAN Version 5 Reference Manual	60481300
Network Products Network Access Method Version 1 Network Definition Language Reference Manual	60480000
Network Products Network Access Method Version 1/ Communications Control Program Version 3 Host Application Programming Reference Manual	60499500
NOS Version 2 Network Terminal User's Instant	60459380
NOS Version 2 Installation Handbook	60459320
NOS Version 2 Manual Abstracts	60485500
NOS Version 2 Operations Handbook	60459310
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300
CYBER Record Manager Basic Access Methods Version 1.5 Reference Manual	60495700
Network Products Network Access Methods Version 1/ Communications Control Program Version 3 Terminal Interfaces Reference Manual	60480600
NOS Version 2 Reference Set, Volume 1, Introduction to Interactive Usage	60459660

Control Data Publication

Publication Number

NOS Version 2 Reference Set, Volume 2, Guide to System Usage	60459670
NOS Version 2 Reference Set, Volume 3, System Commands	60459680
NOS Version 2 Reference Set, Volume 4, Program Interface	60459690
NOS Version 2 Administration Handbook	60459840
NOS Version 2 Analysis Handbook	60459300
TAF Version 1 User's Guide	60459520
TAF/CRM Data Manager Version 1 Reference Manual	60459510
TOTAL - CDC Version 2 Reference Manual	76070300

The NOS 2 Manual Abstracts is a pocket-sized manual containing brief descriptions of the contents and intended audience of all NOS and NOS product manuals. The abstracts can be useful in determining the manuals of greatest interest to a particular user.

Control Data also publishes a Software Publications Release History, publication number 60481000, of all software manuals and revisions packets it has issued. This history lists the revision level of a particular manual that corresponds to the level of software installed at the site.

These manuals are available through Control Data sales offices or Control Data Literature Distribution Services (308 North Dale, St. Paul, Minnesota 55103).

SUBMITTING COMMENTS

The last page of this manual is a comment sheet. Please use it to give your opinion on the manual's usability, to suggest specific improvements, and to report any errors. If the comment sheet has already been used, you can mail you comments to:

Control Data Corporation
Publications and Graphics Division ARH219
4201 Lexington Avenue North
St. Paul, MN 55112

Additionally, if you have access to SOLVER, an online facility for reporting problems, you can use it to submit comments about the manual. Declare your problem type as DOC and use NS2 as the product identifier.

DISCLAIMER

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.

CONTENTS

1. INTRODUCTION	1-1	RERUN Request	2-32
On-Line Transaction Processing Advantages	1-1	RPUT and RGET Requests	2-33
On-Line Transaction Processing Problems	1-1	RSECURE Request	2-34
Batch Interface	1-3	SECURE Request	2-35
Network Interface	1-3	TSTAT Request	2-37
Tasks	1-4	System Task Recovery Requests	2-38
Transaction Processing	1-4	CALLTRN Request	2-38
TAF Requests	1-4	SRRUN Request	2-40
Entering the Transaction Subsystem	1-4	TINVOKE Request	2-40
Implementing an Application	1-5	WSTAT Request	2-41
Subcontrol Points	1-5		
Multimaiframe	1-5		
Extended Memory	1-5		
		3. ACCESS AND USE OF CDCS BY TASKS	3-1
2. TASK/TRANSACTION EXECUTIVE INTERFACE	2-1		
Communication Block	2-1	4. SYSTEM FILES	4-1
Requests	2-3	xxJ File	4-1
Data Manager Requests	2-3	CRM Data Manager xxJ File	4-1
Journal File Requests	2-3	TOTAL Data Manager xxJ File	4-4
Memory Dump Requests	2-3	CDCS or No Data Manager xxJ File	4-4
System Requests	2-3	Creating an xxJ File	4-4.1
Memory Requests (COMPASS Only)	2-3	Journal Files	4-5
Message Requests	2-4	System Journal File	4-5
Task Scheduling Requests	2-4	Data Base Journal Files	4-5
CALLRTN Request	2-4	Journal File Entry Header	4-5
CALLTSK Request	2-6	JOURNL Request	4-5
NEWTRAN Request (COMPASS Only)	2-7	TAF Configuration File	4-7
TRANCHK Request	2-8	DMS Statement	4-7
Input/Output Requests	2-8	DISPLAY Statement	4-7
WAITINP Request	2-9	NETWORK Statement	4-8
LOADCB Request	2-10	RECOVER Statement	4-8
SETCHT Request	2-11	K. Statement	4-8
BLDABH Request	2-12	User Statement	4-8
GETABH Request	2-14	Communication Recovery File	4-8
TERMDEF Request	2-16		
SEND Request	2-17	5. BATCH/TRANSACTION SUBSYSTEM INTERACTION	5-1
RELSQB Request	2-23		
TSIM Request	2-23	BTRAN Request	5-1
TARO Request	2-26	SUBMT Request	5-3
IIO Request	2-27		
BWAITINP Request (COMPASS Only)	2-28		
TPSTAT Request (COMPASS Only)	2-28		
BEGIN Request	2-28		
Task Control Requests	2-28	6. MEMORY DUMPS	6-1
CEASE Request	2-28	Task Dump	6-1
ITL Request	2-29	DSDUMP Request	6-1
LOGT Request (COMPASS Only)	2-29	CMDUMP Request	6-3
KPOINT Request	2-30	K.DSDUMP Command	6-4
WAIT Request	2-30	KTSOMP Utility	6-4
Task Utility Requests	2-30.1	Task Dump Formats	6-4
LOGIN Request	2-31	TAF Dump	6-4
EXTRACT Request	2-31	K.DUMP Command	6-5
INSERT Request	2-32	K.DUMPLIM Command	6-5
Application Recovery Requests	2-32		

7. APPLICATIONS INTERFACE FOR AUTOMATIC RECOVERY	7-1	10. TASK LIBRARY	10-1
Purpose	7-1	LIBTASK Utility	10-1
Concepts	7-1	Task Residency and Type	10-1
Named-Transactions and Run Units	7-1	LIBTASK Command	10-2
Data Base Recovery	7-1	Input Directives	10-5
Transaction Recovery	7-1	Named-Transaction Input Directive	10-5
Coding	7-2	Task Input Directive	10-5
Coding Structures for Data Base Recovery Only	7-2	Scheduling Priority	10-7
Transactions Using CRM	7-2	Creating a New Task Library	10-7
TAF Transactions Using CDCS	7-2	Updating an Existing Task Library	10-7
Coding Structures for Transaction Recovery	7-3	Deleting a Task from a Task Library	10-7
Transactions Using CRM	7-3	Ignoring a Task in the Replacement File	10-8
TAF Transactions Using CDCS	7-3		
Terminal Recovery Considerations	7-4	11. REQUIRED SYSTEM TASKS AND TRANSACTIONS	11-1
CALLTSK without CEASE and RECOVERY	7-5	BTRAN Recovery Task (BTASK)	11-1
Multiple Data Manager Use	7-5	Data Base Recovery Task (CTASK)	11-1
TOTAL Data Manager Use	7-5	Initial Task (ITASK)	11-1
		K Display Task (KDIS)	11-1
		Logout Task (LOGT)	11-1
		Message Abort (MSABT)	11-1
8. DATA ADMINISTRATOR AND OPERATOR INTERFACE FOR RECOVERY	8-1	Inactive Task (OFFTASK)	11-2
Recovery Overview	8-1	CDCS Rerun Task (RCTASK)	11-2
Hardware Configuration Considerations	8-2	Run Unit Recovery Task (RTASK)	11-2
Installation	8-2	Send Message Task (STASK)	11-2
TAF K Display	8-2	System Message Task (SYSMSG)	11-2
Initial K Display	8-2	Execute Named Task (XTASK)	11-2
K.INT Command	8-2	XTRAN Transaction	11-2
K.ERO Command	8-3		
K.REC Command	8-3	12. ITASK SYSTEM TASK	12-1
Restart K Display	8-3	Macros	12-1
TCF/CRF Recovery Considerations	8-3	TIMCNT Macro	12-1
TAF Configuration File	8-3	TRAN Macro	12-1
Communication Recovery File	8-3	STRAN Macro	12-1
Error Override	8-3	ITASK Description	12-2
File Reformatting	8-3	TAF-Originated Transactions	12-3
Multiple Communication Recovery Files	8-4	Conditions Requiring TAF-Originated Transactions	12-3
Multimainframe Recovery	8-4	Time Activation	12-3
Recovery Processing Flow	8-4	K.IDLE Console Operator Command	12-3
TAF and System Failure	8-4	Terminal Login	12-4
NAM Failure	8-5	Terminal Break Conditions	12-4
Terminal Failure	8-5	Terminal Connection Broken	12-4
CDCS Failure	8-5	Network Normal Shutdown	12-4
Task Failure	8-5	Network Failure	12-4
TAF Termination	8-5	Network Immediate Shutdown	12-4
TAF Automatic Recovery Report	8-6	Terminal Inactive	12-4
		Logical Error	12-4
		Block Not Delivered	12-5
		Terminal Characteristic Change	12-5
		Terminal Input Too Large	12-5
		Terminal Type Ahead	12-5
		User Status	12-5
9. NETWORK FILES	9-1		
Terminal Network Description File	9-1		
Network File Origination	9-1		

APPENDIXES

A. CHARACTER SETS	A-1	D. SAMPLE COMMAND LIST STRUCTURES	D-1
		COMPASS Task Command List Structure	D-1
		FORTRAN Task Command List Structure	D-1
B. DIAGNOSTIC MESSAGES	B-1	COBOL Task Command List Structure	D-1
C. GLOSSARY	C-1	E. FILES USED BY TAF	E-1

F. INSTALLATION OVERVIEW	F-1	Step 1 - Gathering Information	J-1
		Step 2 - Setting Up Your Terminal	J-1
		Step 3 - Connecting Your Terminal to the Network	J-1
G. TERMINAL DEFINITION COMMANDS	G-1	Step 4 - Identifying Your Terminal to the Network	J-3
Introduction	G-1	Step 5 - Selecting a Host	J-3
Terminal Definition Command Format	G-1	Step 6 - Connecting to the Selected Host	J-3
Restrictions	G-2	Step 7 - Identifying Yourself and Your Terminal to the Host	J-4
Terminal Definition Commands	G-2	Step 8 - Selecting a Network Application	J-4
(AB) - Abort Output Block Character	G-3	Detailed Description of Access	J-4
(AR) - Automatic Character Recognition	G-3	Step 1 - Gathering Information	J-4
(BF) - Blocking Factor	G-3	Step 2 - Setting Up Your Terminal	J-7
(BR) - Break Key as User Break 1	G-3	Step 3 - Connecting Your Terminal to the Network	J-8
(BS) - Backspace Character	G-4	Step 4 - Identifying Your Terminal to the Network	J-8
(B1) - Interruption or User Break 1 Character	G-4	Procedure for Asynchronous Terminals	J-8
(B2) - Termination or User Break 2 Character	G-4	Procedure for HASP Terminals	J-9
(CH) - Display Terminal Characteristics	G-4	Procedure for Mode 4 Terminals	J-10
(CI) - Carriage Return Idle Count	G-4	Procedure for Bisynchronous Terminals	J-10
(CN) - Cancel Character	G-5	Step 5 - Selecting a Host	J-10
(CP) - Cursor Positioning after Input	G-5	Selecting a Host	J-10
(CT) - Network Control Character	G-5	Selecting a Specific Host Path	J-11
(DL) - Delimiters for Single-Message Transparent Input Mode	G-5	Controlling the Display of Host Paths	J-11
(EB) - End-of-Block Character	G-6	Step 6 - Connecting to the Selected Host	J-13
(EL) - End-of-Line Character	G-7	Step 7 - Identifying Yourself and Your Terminal to the Host	J-13
(EP) - Echoplex Mode	G-7	Automatic Login	J-14
(FA) - Full-ASCII Input Mode	G-7	Login Dialog	J-14
(HC) - Host Connection	G-8	Step 8 - Selecting a Network Application	J-15
(HD) - Display of Host Nodes	G-8	Automatic Connection	J-15
(HN) - Host Node Selection	G-9	Manual Connection	J-15
(HS) - Host Selection	G-9	Abbreviated Login and Application Connection	J-16
(IC) - Flow Control for Input Devices	G-9	Switching to a Different Application	J-16
(IN) - Input Device and Transmission Mode	G-9	Direct Switches	J-16
(LI) - Line Feed Idle Count	G-10	Indirect Switches	J-17
(LK) - Lockout of Unsolicited Messages	G-10	Dialog for Direct and Indirect Switches	J-17
(MS) - Message to Network Operator	G-10	Restarting Login Identification	J-17
(OC) - Flow Control for Output Devices	G-11	Direct Restarts	J-17
(OP) - Output Device Selection	G-11	Indirect Restarts	J-17
(PA) - Parity Processing	G-11	Secure Login Restarts	J-17
(PG) - Page Waiting	G-12	Disconnecting from a Host	J-18
(PL) - Page Length	G-12	Disconnecting with Application Commands	J-18
(PW) - Page Width	G-12	Disconnecting with Host Commands	J-18
(SE) - Special Editing Mode	G-12	Disconnecting with a Network Software Command	J-18
(TC) - Terminal Class Command	G-12	Reconnecting to a Host or Connecting to a Different Host	J-18
(TM) - Terminating a Terminal-Host Connection	G-13	Disconnecting from One Host and Connecting to a Different Host	J-18
(XL) - Multinessage Transparent Mode	G-13	Interruptions	J-19
H. DELIMITING AND TRANSMITTING TERMINAL INPUT	H-1	Communicating with the Network Operator	J-19
Physical End-of-Line (Line Feed)	H-1	Receiving Messages	J-19
Logical End-of-Line (Carriage Return)	H-1	Sending Messages	J-19
Message Transmission	H-1	Preventing Messages	J-19
I. DETECTION OF POTENTIALLY BLOCKED TASKS	I-1	Suspensions of Communication	J-19
		Communication Failures	J-19
		Application Failures	J-20
J. ACCESSING A HOST	J-1	Disconnection from an Application and/or Host	J-20
Brief Description of Access	J-1	Disconnection from the Network	J-20

INDEX

FIGURES

1-1	TAF Control Point for CDCS Data Manager	1-2	5-4	SUBMT Request	5-4
1-2	TAF Control Point for TAF/CRM Data Manager	1-2	6-1	DSDUMP Parameter Table	6-3
1-3	TAF Control Point for TOTAL Data Manager	1-3	6-2	CMDUMP Parameter Table	6-4
2-1	Communication Block Format	2-2	6-3	KTSDMP Format	6-5
2-2	CALLRTN	2-5	7-1	Recoverable TAF Transaction Using CDCS	7-4
2-3	CALLRTN (Chain)	2-5	8-1	Format of Part Three of the TAF Automatic Recovery Report	8-6
2-4	CALLRTN (Multiple Chains)	2-5	9-1	Communication Between a Terminal Status Table and a Communication Block	9-2
2-5	CALLTSK with CEASE Request	2-6	10-1	Task Library Report	10-4
2-6	CALLTSK without CEASE Request	2-7	12-1	Format of a TTOT Table Entry	12-1
2-7	NEWTRAN Parameter Table	2-7	12-2	Format of a TRANT Table Entry	12-2
2-8	TRANCHK Parameter Table	2-8	12-3	Format of a TRANS Table Entry	12-2
2-9	LOADCB Parameter Table	2-11	12-4	Communication Block for TAF- Originated Transactions	12-3
2-10	SETCHT Parameter Table	2-13	D-1	COMPASS Command List Structure	D-1
2-11	GETABH Parameter Table	2-15	D-2	FORTTRAN Command List Structure	D-1
2-12	TERMDEF Parameter Table	2-17	D-3	COBOL Command List Structure	D-1
2-13	SEND Parameter Table	2-23	J-1	Separate Acoustic Coupler	J-2
2-14	TSIM Parameter Table	2-25	J-2	Data Set Built into the Terminal	J-2
2-15	TARO Parameter Table	2-27	J-3	Data Sets with Switches and Buttons	J-2
2-16	IIO Parameter Table	2-27	J-4	Terminal with Built-in Acoustic Coupler	J-3
2-17	ITL Parameter Table	2-30	J-5	Interactive Display Console	J-5
4-1	Journal File (Buffered and Nonbuffered Structure)	4-2	J-6	Interactive Printer Console	J-5
4-2	Journal File Entry Header	4-6	J-7	Remote Batch Terminal	J-6
4-3	JOURNL Parameter Table	4-6	J-8	Host Availability Display	J-11
5-1	BTRAN Header Word	5-2	J-9	Sample Login from an Asynchronous Terminal	J-13
5-2	BTRAN Request	5-2			
5-3	SUBMT Parameter Table	5-3			

TABLES

2-1	BLDABH Request Parameters	2-13	A-1	7-Bit ASCII Code Standard Print 63/64-Character Set	A-1
2-2	Application Block Header	2-15	E-1	Files Used by TAF	E-1
2-3	COBOL SEND Request Parameters	2-18	F-1	Installation Overview	F-2
2-4	FORTTRAN SEND Request Parameters	2-20	G-1	Default Terminal Definitions	G-15
2-5	COMPASS SEND Request Parameters	2-21	G-2	Parameter Ranges for Terminal Definition Commands	G-16
2-6	COMPASS SEND Parameter Interrelationships	2-24	H-1	Default Message Delimiter and Transmission Keys	H-2
2-7	TSAT Keyword Descriptions	2-39			
2-8	WSTAT Keyword Descriptions	2-42			
5-1	BTRAN Status Values	5-2			

TAF is a network product that controls transaction processing. Transaction processing consists of taking an existing collection of information, called a data base, and correcting old data or adding new data to create an up-to-date data base. Such a correction or addition is called a transaction.

In a transaction processing system, the data base must be efficiently structured and easily accessible to the user. A data manager provides these features.

Control Data offers the following data managers for use with TAF.

- The CYBER Database Control System (CDCS).
- The CYBER Record Manager (CRM). (TAF supports only those features documented in the TAF/CRM Data Manager 1 Reference Manual.)
- The TOTAL Data Manager.

All data managers can operate concurrently. However, a single transaction can use only:

- CDCS and one other data manager,
- or
- CDCS alone,
- or
- One other data manager alone.

The main function of TAF is on-line transaction processing; that is, direct operator interface with the data base through a terminal.

ON-LINE TRANSACTION PROCESSING ADVANTAGES

Some advantages associated with on-line transaction processing are:

- The operator submitting the data through TAF can make immediate data corrections.
- The operator can verify that operations have completed successfully. TAF allows the operator to view additions or corrections to the data base to verify their accuracy.
- The data base is updated in seconds. TAF provides immediate updating capability, which allows other transactions to make use of the updated data base.

The most important feature of TAF is its efficient system resource utilization. TAF's transaction processing differs from other modes of operation in that a task only contains application code. Terminal communications are handled by one terminal communications interface. Data base communications are handled by one copy of a data base manager. When tasks need to be loaded, they are simply copied from a task library to a subcontrol point; none of the normal overhead of loading is incurred.

The TAF control point contains application tasks as subcontrol points. When a task completes, one of the following occurs:

- It may be immediately reused by other waiting transactions (no reloading is necessary).
- Another task may be loaded in its place.
- Other active tasks may be moved up in memory and the remaining memory returned to the system for use by non-TAF applications.

TAF will expand its memory to a maximum value as needed.

When there is no transaction activity, TAF size is reduced to a small amount of memory. When input is sensed, TAF rolls back in to service the request.

Figures 1-1, 1-2, and 1-3 summarize the TAF control point and its operation.

ON-LINE TRANSACTION PROCESSING PROBLEMS

Some problems associated with on-line transaction processing and the methods that TAF uses to alleviate them include the following.

- It is difficult to schedule and control multiple jobs. TAF solves the problem of multiple job control scheduling by using subcontrol points. Like the operating system that controls and schedules jobs to control points (where they are processed), TAF controls and schedules transaction processing programs to portions of its control point called subcontrol points. Subcontrol points are protected from each other by CYBER hardware memory protection.
- The terminal interface is complicated. TAF minimizes terminal programming by using simple input/output (I/O) commands that are processed by the standard network terminal interface.

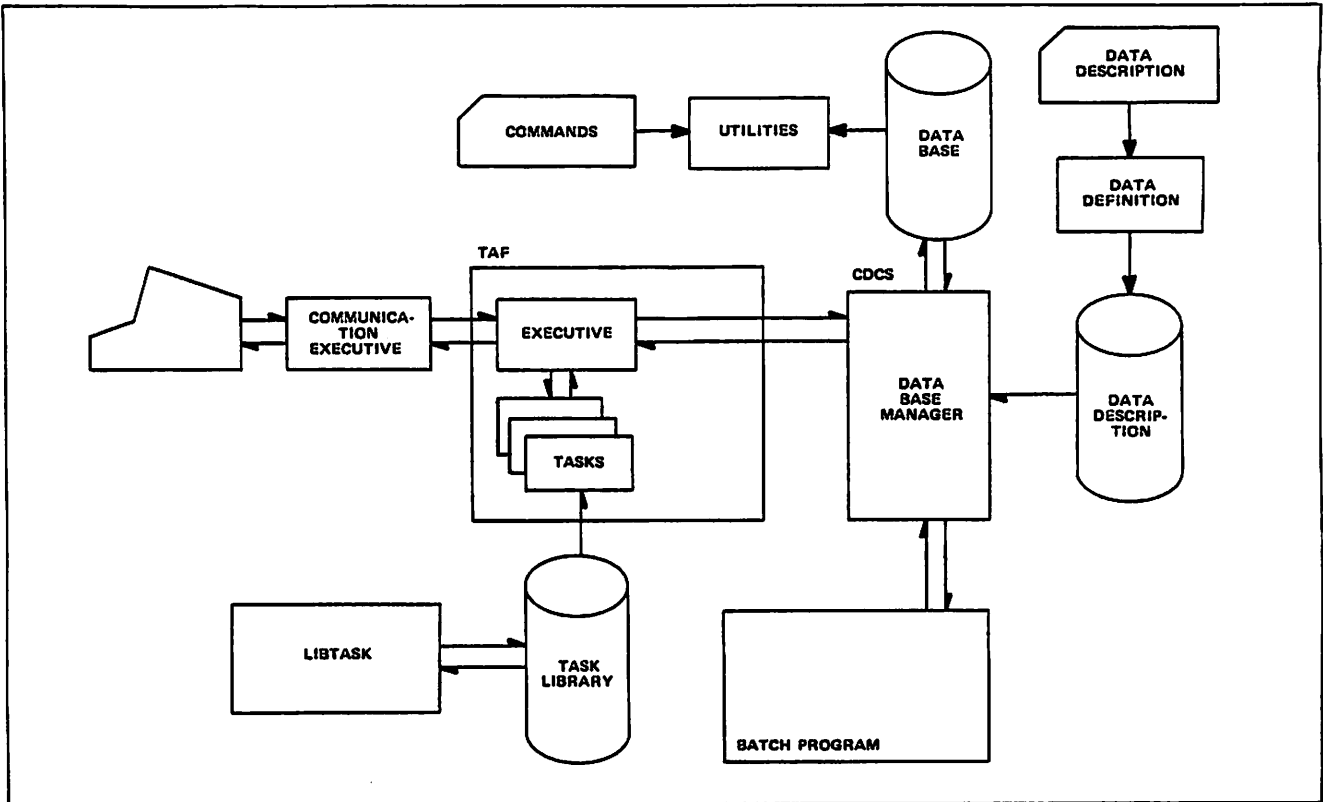


Figure 1-1. TAF Control Point for CDCS Data Manager

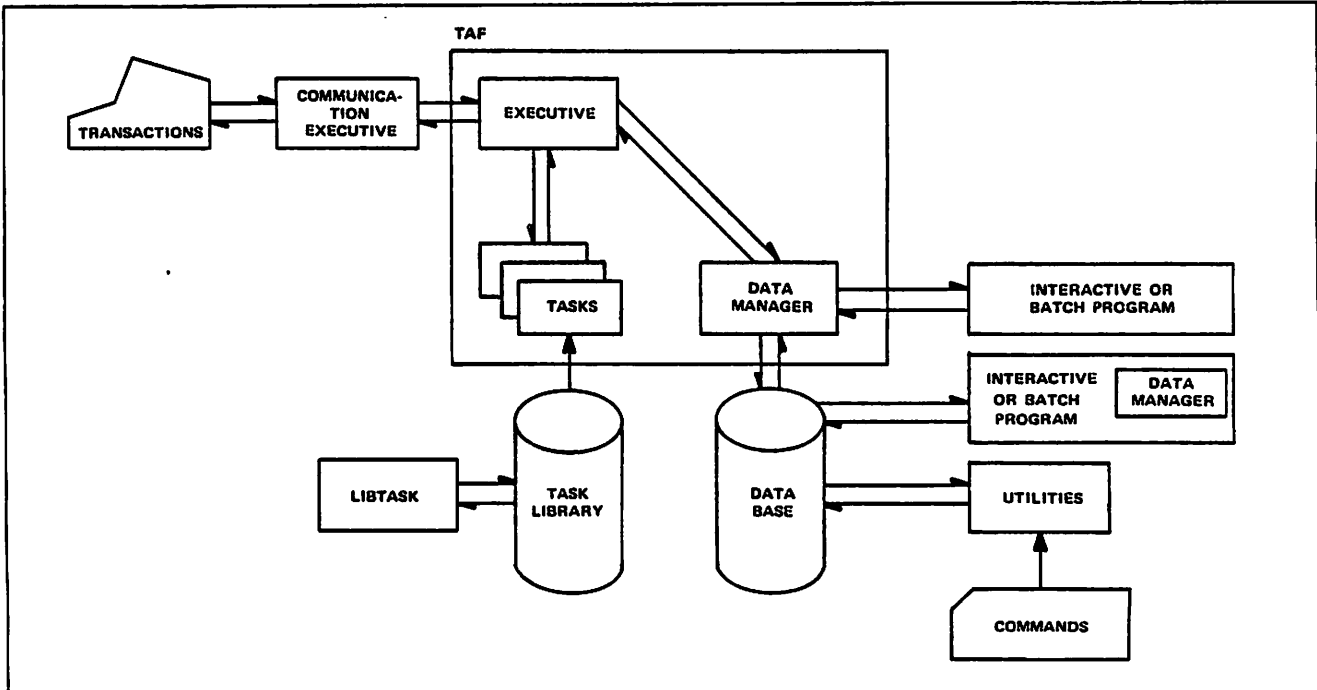


Figure 1-2. TAF Control Point for TAF/CRM Data Manager

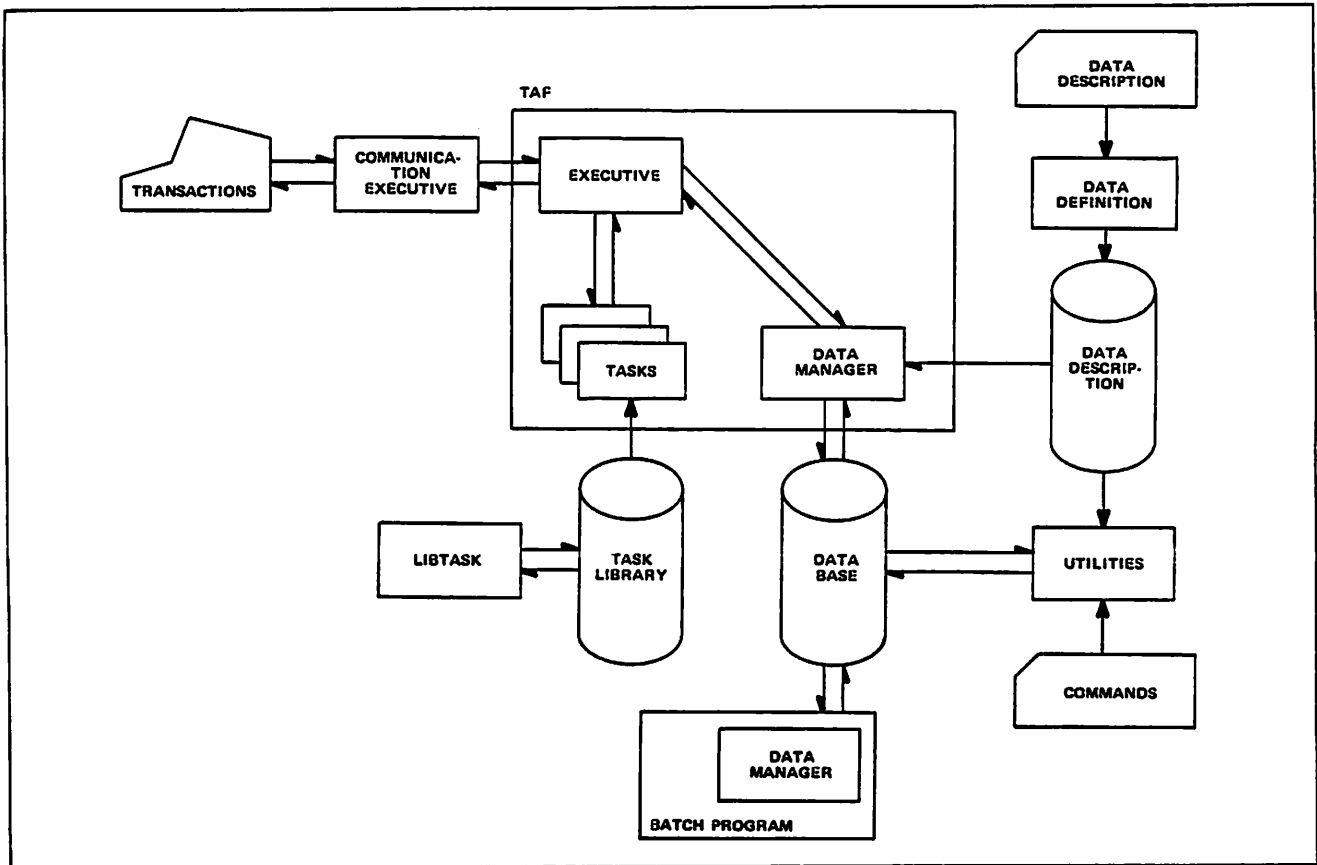


Figure 1-3. TAF Control Point for TOTAL Data Manager

BATCH INTERFACE

A batch interface with TAF is also available. The batch interface can generate large reports without tying up a terminal until the listing has been printed. The report can be printed at a high-speed central site printer. Batch mode can also be used for testing and debugging. The batch interface is described in section 5.

NETWORK INTERFACE

The network terminal interface uses NAM and 255x communications processors. This interface is used by all other network products (for example, Remote Batch Facility and Interactive Facility). If validated, terminal users may select these applications as well as TAF. The network provides the following features for transaction processing.

- Managing network protocol.
- Buffering and queuing data for regulating data flow.
- Supporting a wide variety of terminals through standardization of data formats as well as providing the ability to handle transparent data.
- Sharing network among communication-oriented services.

The network makes use of the interactive virtual terminal (IVT) concept. This concept abstracts certain functions of a variety of real terminals. The applications programmer need not be concerned with character sets and communications protocols.

- An IVT has an input/output device that sends or receives certain amounts of data termed logical lines. These logical lines are transformed into physical lines of characters of the appropriate character set for the real terminal.
- Logical lines that exceed the physical capacity of the real terminal are automatically folded into two or more physical lines.
- The spacing of logical lines of output can be further controlled by the use of format effectors.
- Output to a device may be optionally paged, so that data that would overwrite any output being displayed is not sent until the terminal user has acknowledged the preceding output.
- Both the application and the terminal user can redefine terminal characteristics of the actual terminal, which may differ from those assumed, and supply or redefine the operational controls provided.

When an application requires features not provided by the IVT but known to exist on the connected real terminal, the application may do one of the following.

- Embed appropriate control characters in the output text.
- Transfer data in transparent mode, in which case all transforms are inhibited, and the application has direct access to and responsibility for all real terminal features including the character set.

TASKS

The applications programmer implements a transaction system by writing tasks. A task is a program with a TAF interface that performs a specific function. A single task or several tasks perform a sequence of events called a transaction. Tasks can read and update information on the user's data base. They can send messages to terminals and receive input. Tasks can also schedule other tasks to assist in completing transactions.

Tasks that run under TAF can be written in COBOL, FORTRAN, or COMPASS. A task differs from a normal batch or interactive program in that it contains calls to TAF commands and data manager commands and must have a common block (referred to as the communication block) defined. The communication block is needed to receive terminal input and pass data between tasks. (The communication block is described in section 2.)

Binary copies of each task are stored in the task library. The LIBTASK utility is used to place these binary copies on the task library. Section 10 describes the task library and LIBTASK.

Tasks written in COBOL must be compiled with the TAF parameter on the COBOL command. Tasks written in FORTRAN must be compiled with the Z parameter on the FORTRAN command (FORTRAN 4) or with DB=0 on the FTNS command (FORTRAN 5).

TRANSACTION PROCESSING

After following the login procedures outlined in appendix J, the user indicates the transaction to be performed by entering a code that is assigned by the system designers. Such a code could specify that a user is entering a loan payment; another code would be used to close out a loan, and so on. TAF receives this initial input and places it in a communication block. This communication block is the means by which the data is passed from a terminal to a task or from task to task.

When a transaction is completed, the communication block is released and can be used by other transactions.

After the communication block has been constructed, TAF schedules an initial task (ITASK) to interpret the code the user entered. Using this code, ITASK requests TAF to schedule the appropriate task(s) to process the transaction. The task scheduled may also call other tasks to assist in completing the transaction.

TAF REQUESTS

Communication between terminals and tasks is accomplished by using TAF requests. Output can be sent to a transaction terminal from the task by using the SEND request. When this request is processed, control passes to TAF. TAF extracts the message from the task and passes it to the network. The message is then transmitted to the transaction terminal, and control passes back to the task for additional processing.

When a task requires additional input from the terminal, the WAITINP request is used. The task performs a SEND request to send output to the terminal, prompting the terminal operator for the needed input. The WAITINP request is then processed. Since the terminal operator takes seconds to respond to the request, the task is rolled out of central memory and the subcontrol point and communication block are released. These resources can then be used to process other transactions while the terminal operator is responding. When TAF receives the requested input, it places it into a new communication block, and the task is rolled into an available subcontrol point. The new communication block is passed to the task, and the task continues processing.

These and other requests are further described in sections 2 and 5.

ENTERING THE TRANSACTION SUBSYSTEM

The subsystem you will be accessing is the TAF subsystem. There are various procedures for accessing this transaction subsystem. These methods are discussed in detail in appendix J. Using appendix J, a coordinator of TAF activities at a particular site who knows the types of users and applications at the site can determine the most desirable access procedures.

Because of the flexibility of the network and the transaction subsystem, the procedure to access the transaction subsystem can vary substantially from site to site. Therefore, the coordinator of transaction subsystem operations at each site should provide each user, or terminal location, with a detailed login procedure for that site.

The site should also provide information concerning terminal characteristics. A variety of terminal classes is supported for use on the network with

TAF. Each of these terminal classes has characteristics, which the analyst can change by modifying the network files and which you can change by entering terminal definition commands.

A discussion of the terminal characteristics, the defaults for each terminal class, and the procedure for changing these characteristics at a terminal is included in appendix G.

IMPLEMENTING AN APPLICATION

The following steps specify the procedure for implementing an application.

1. Review this manual.
2. Design the application including:
 - Selection of the data manager (CDCS and/or CRM, or TOTAL) to allow the correct choice of file types, file structure, and file interaction.
 - Selection of recovery and backup techniques, including journalizing of files.

Refer to the CYBER Database Control System 2 Reference Manuals, the TAF/CRM Data Manager 1 Reference Manual, or the TOTAL - CDC 2 Reference Manual for more information (publication numbers are listed in the preface).

3. Describe and install the data base using the data manager. Refer to the appropriate data manager reference manual for this procedure.
4. Test load the data base using the data manager commands (usually done in batch mode).
5. Using the application design, write the application tasks to process transaction input.
6. Optionally modify ITASK to reflect the actual application. Tasks MSABT, LOGT, and SYMSG may also require modification. (Refer to section 11.)
7. Build a task library. (Refer to section 10.)
8. Build an xxJ file. (Refer to section 4.)
9. Build a network file, and use NDL to configure terminals and establish login sequences and terminal characteristics.
10. Build the TAF configuration file (TCF). (Refer to section 4.)
11. Test the system.

SUBCONTROL POINTS

The following discussion is intended for analysts and may be omitted by the applications programmer.

Tasks reside at subcontrol points within the field length of the transaction executive. Each task runs with its own exchange package, reference address (relative to the transaction executive), and field length (a subset of the field length of the transaction executive). A task, therefore, can reside in any contiguous segment within the transaction executive field length. Memory is allocated in 100g word blocks located on even 100g word boundaries (for example, 32100g to 32200g). A subcontrol point consists of one 120g word block of memory for system use, followed by sufficient blocks to contain an entire task.

The subcontrol point feature allows the transaction executive to control each task. Some of the advantages associated with the subcontrol points are:

- Isolating one subcontrol point from other subcontrol points and the transaction executive. This means that no application program can destroy the transaction executive or circumvent system security.
- Blocking RA+1 requests from a subcontrol point. No requests are allowed directly from a subcontrol point to NOS. Any such requests are intercepted by the system monitors, which return control to the transaction executive. Thus, the only RA+1 requests a task can legally issue are those processed by the transaction executive. This includes requests to the data managers.

The transaction subsystem allows a maximum of 31 subcontrol points. An installation parameter sets the number of subcontrol points that the transaction executive initializes. When the transaction executive is loaded, the operator can select a number of subcontrol points other than this default value. The number of subcontrol points must not be less than 2 nor greater than 31. Once the transaction subsystem is initialized, no change in the number of subcontrol points is allowed. The optimum number of subcontrol points is selected by the site.

Each subcontrol point requires eight words of table space. No space, other than a table entry, is allocated for a subcontrol point unless it is active. Each active subcontrol point has 111g words reserved, beginning at RA, similar to those used for batch control points.

MULTIMAINFRAME

Under a multiframe configuration, the transaction subsystem can be run on either or both mainframes. However, a data base and JOURO file (section 4) cannot be shared by the transaction subsystems. Each subsystem must have a unique data base.

EXTENDED MEMORY

TAF does not allow the use of your access to extended memory. However, the TAF procedure file can declare data files and libraries as extended memory-resident.

A task is isolated from the data manager and the remainder of the system by the transaction executive. The isolation guarantees that errors in a task do not propagate beyond the task. The following types of requests constitute the interface between the task and the transaction executive.

- Data manager requests
- Journal file requests
- Memory dump requests
- System requests
- Task scheduling requests
- Input/output requests†
- Task control requests
- Task utility requests
- Application recovery requests
- System recovery requests

Internally, a task performs a request by loading address RA+1 of its subcontrol point. This activity is not visible to the applications programmer who uses the macros or calls given in this section. The transaction executive uses the request parameters to determine what processing is required by the task. The address field and any fields in a parameter list that reference an address contained within the task are verified to ensure that the address referenced is within the task field length.

Tasks communicate with each other and accumulate input by using the communication block. The communication block is discussed next, followed by the types of requests.

COMMUNICATION BLOCK

Communication between a chain of tasks is provided by the communication block. The initial input or the input from a terminal as a result of the WAITINP request (refer to Input/Output Requests later in this section) is also stored in the communication block. The communication block can be used to pass parameters or data from task to task. The format of the communication block is shown in figure 2-1. The length of terminal input is not restricted by the length of the communication block. The LOADCB request is available to read extra input that overflows

the communication block (refer to the description of m in figure 2-1). If the communication block is passed between tasks in a transaction and overflow data is present, other tasks in the transaction can access this data with a LOADCB request.

If COBOL is used, the communication block must be described in the COMMON-STORAGE SECTION. Shown here is an example of such a description:

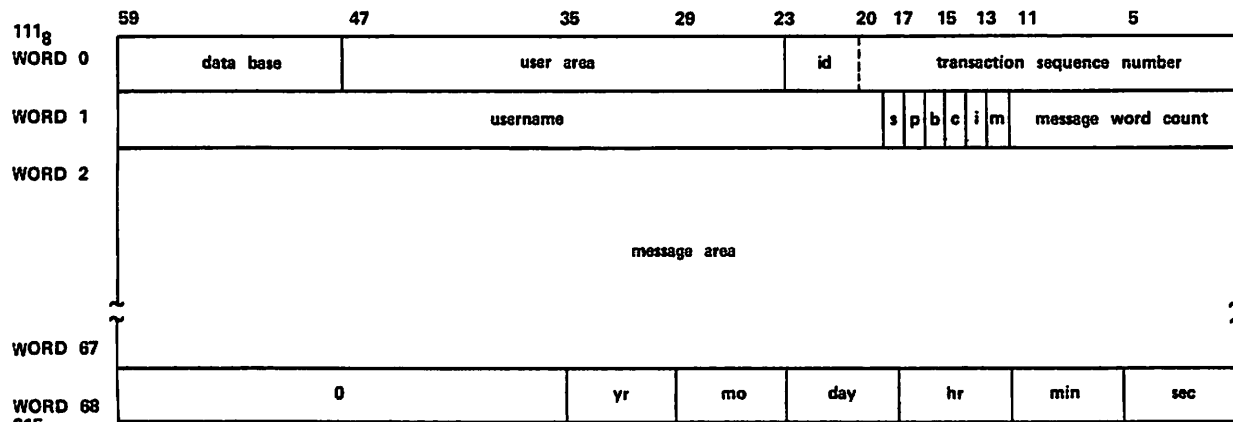
```

      .
      .
      .
DATA DIVISION.
COMMON-STORAGE SECTION.
01 COMMUNICATION-BLOCK.
   03 CB-HEADER-AREA-WORD-ZERO.
      05 CB-DATA-BASE-NAME                PIC X(2).
      05 CB-USER-TST-AREA                 COMP-4 PIC 9(6).
      05 CB-TRANS SEQ-NUMBER              COMP-4 PIC 9(6).
   03 CB-HEADER-AREA-WORD-ONE.
      05 CB-USERNAME                      PIC X(7).
      05 CB-INPUT-STATUS-BYTE             COMP-4 PIC 9.
      05 CB-MESSAGE-WORD-COUNT           COMP-4 PIC 99.
   03 CB-MESSAGE-AREA                    PIC X(570).
   03 USER-AREA                          PIC X(90).
   03 CB-TRAILER-AREA.
      05 FILLER                          PIC X(4).
      05 CB-PACKED-DATE.
          07 CB-YEAR                      COMP-4 PIC 9.
          07 CB-MONTH                    COMP-4 PIC 9.
          07 CB-DAY                      COMP-4 PIC 9.
      05 CB-PACKED-TIME.
          07 CB-HOUR                     COMP-4 PIC 9.
          07 CB-MINUTE                   COMP-4 PIC 9.
          07 CB-SECOND                   COMP-4 PIC 9.
WORKING-STORAGE SECTION.
      .
      .
      .
    
```

This will be referred to in the manual as the standard COBOL communication block because of the correspondence between it and the fields shown and described in figure 2-1. However, applications programmers may find it convenient to use a more abbreviated communication block when possible.

COMP-4 is specified for data areas intended for numerical information. Refer to the COBOL 5 Reference Manual for information on the relationship between the number of digits specified in the picture clause for COMP-4 data and the number of bits reserved.

†The standard CMR-supported, compiler-generated I/O calls from FORTRAN and COBOL are not allowed. The COBOL SORT verb is not supported.



data base Data base that the originating terminal is validated to use.

user area Image of originating terminal user area from the terminal status table (refer to section 8). Contains the recovery flag. (Refer to the TARO and TSIM requests for more information.)

transaction sequence number Assigned by the transaction executive to each newly initiated transaction. This number is generated by incrementing a counter by one and identifies the transaction, or task chain, throughout its life in the subsystem.

id The first octal digit of the transaction sequence number. It is the identifier for the network file under which the user running this transaction is defined. The value is zero to seven.

username One- to seven-character user name, left-justified with binary zero fill from the terminal status table.

s† Bit 17 of word 1 is set to one if this communication block was created by a TAF-originated transaction. Word 2, and possibly 3, of the communication block contains additional information (operating system or network-defined conditions that may be processed by TAF-originated tasks). Refer to section 11 for information concerning required system tasks.

pt Bit 16 of word 1 is set to one if this communication contains data on which a parity error occurred during terminal input.

bt Bit 15 of word 1 is set to one if this communication block was created by a batch transaction (BTRAN request).

c Bit 14 of word 1 is set to one if CDCS aborts.

i Bit 13 of word 1 is set to one if TAF is idling down.

m† Bit 12 of word 1 is set to one if additional input to be loaded by means of the LOADCB request exists.

message word count Length of the input message is words in the communication block proper.

message area A message from a terminal (as a result of initial terminal input, WAITINP, or BWAITINP requests) begins at word 2 and is terminated by 12 bits of zero in positions 11-0 of the last word of the message. All remaining message area words contain binary zero. The maximum length of a terminal message is 57 words. This area may also contain data from a BTRAN request. The maximum length of a BTRAN message is 62 words. In all cases, the length of the message is in the message word count field. The information contained in this area is available to you, and its contents is passed to any communication block generated by the processing of a CALLTSK or CALLRTN request. To retain the terminal input, you may only write in words message word count+2 through 67. If there is no need to retain the terminal input, the entire area can be used.

yr, mo, day, hr, min, sec Date and time in packed binary format from the operating system PDATE macro.

†The main purpose of these fields is to notify the system task ITASK of certain conditions. Under COBOL, the EXTRACT routine allows examination of these fields. Also, a COBOL data type, computational-4, can be used to access these fields more easily; computational-4 data types are allowed only on COBOL Version 5.2 and above. ITASK is described in section 12, and EXTRACT is described later in this section.

Figure 2-1. Communication Block Format

If FORTRAN is used, the communication block must be in the first block of labeled common.

If more data is input than can be contained in the communication block, the LOADCB request may be used to obtain the input.

REQUESTS

The following paragraphs describe the various types of requests a task can make. They describe task scheduling, input/output, task control, and task utility requests and show the formats for making these requests from tasks written in FORTRAN, COBOL, and COMPASS.

DATA MANAGER REQUESTS

A task can make data manager requests to one of the data managers, (CDCS, CRM, or TOTAL) supported by the transaction subsystem. (Refer to the preface for a list of the publications that document these data managers and the requests used.)

JOURNAL FILE REQUESTS

A task can supplement automatic journaling by writing data on a journal file. Journal files and the journal requests are discussed in section 4.

MEMORY DUMP REQUESTS

These requests are discussed in section 6.

SYSTEM REQUESTS

System requests from transaction tasks are limited to the following.

COMPASS Macros

ABORT
DATE x
ENDRUN
JDATE x
MEMORY (as described below)
MESSAGE x (to console B display, line one only)
PDATE x
RTIME x
TIME x

FORTRAN Functions and Subroutine Calls

DATE(x) or CALL DATE(x)
JDATE(x) or CALL JDATE(x)
TIME(x) or CALL TIME(x)

COBOL Statements

ACCEPT x FROM DATE.
ACCEPT x FROM TIME.
ACCEPT item FROM TTY.
DISPLAY item UPON TTY.

These requests are described in detail in Volume 4 of the NOS 2 Reference Set, FORTRAN Extended 4 Reference Manual, FORTRAN 5 Reference Manual, and the COBOL 5 Reference Manual. (Refer to the preface for publication numbers.)

MEMORY REQUESTS (COMPASS ONLY)

Tasks can interrogate the status of and change their central memory (CM) field length (FL) via the RA+1 requests MEM and RFL. The format of these calls is documented for the MEMORY macro in the NOS 2 Reference Set, Volume 4. The use of these requests in TAF is restricted as follows:

- Memory requests should be issued with the recall (r parameter); otherwise, control may be returned before the request is processed.
- There is an upper limit beyond which a task is not allowed to increase its CM space; this limit is referred to as the maximum task field length (MFL). The MFL for each task is equal to the task FL plus the value specified by the EF LIBTASK directive for the task (refer to section 10). If a task attempts to exceed this value, it may be aborted (refer to documentation of the MEMORY macro NA parameter in the NOS 2 Reference Set, Volume 4).
- Field length cannot be decreased to less than the last word address of the default communication block transfer address rounded up to the nearest 100g.
- A non-CM-resident task making a memory request may be rolled out of CM if the request cannot be satisfied initially.
- Since a task making a memory request may be rolled out, it may be desirable to free locked data manager resources prior to the request.

Tasks will also be allowed to use the common memory manager (CMM) with the following restrictions:

- The processing performed by CMM for the task must not result in CMM using the fast dynamic loader or the task will be aborted.
- The preceding restrictions on use of MEM and RFL requests apply.

Common memory manager requests are documented in detail in the Common Memory Manager 1 Reference Manual.

MESSAGE REQUESTS

COMPASS tasks can issue messages to the operator or the JOURO file by using the RA+1 request MSG. The function code determines the disposition of the message. Function code zero causes the message to be written to only the JOURO file. Function code one causes the message to be placed in line one of the B display.

FORTRAN termination messages are also written to the JOURO file.

COBOL and FORTRAN tasks can issue messages to TAF's dayfile by using the TMSG request. The tasks that issue this request must reside in the system task library. **TASKLIB**

The TMSG request issues a message to TAF's dayfile.

The format is:

COBOL

ENTER TMSG USING message[,length].

message	A 01-level data item containing the dayfile message.
length	A computational-1 data item specifying the length of the message in characters. The maximum length is 80 characters.

FORTRAN

CALL TMSG(message,length)

message	The name of the array that contains the dayfile message.
length	An integer specifying the length of the message in characters. The maximum length is 80 characters.

TASK SCHEDULING REQUESTS

A task, including ITASK (refer to section 11), can call in order a list of other tasks. This process is referred to as task scheduling. The following four requests can be used to schedule tasks.

- **CALLRTN**
Calls up to five tasks. Upon completion of the called chain of tasks, returns control to the calling task.
- **CALLTSK**
Calls up to five tasks. The task making the call either ceases or continues to run, allowing the new task chain to proceed independently.

- **NEWTRAN**
Initiates up to five new tasks (COMPASS only).
- **TRANCHK**
Checks to see if a specific task is active (COMPASS only).

NOTE

If a list has either no tasks or more than five tasks, TAF aborts the task and issues the following error message on the screen.

TASK REQUEST ARGUMENT ERROR

CALLRTN Request

The CALLRTN request is used to request that the transaction executive schedule a task or series of tasks, with control returning to the calling task upon completion of the called tasks. A CALLRTN request cannot be made while data manager requests are outstanding. The current communication block is passed to the called tasks and returned to the caller when the caller is reinitiated. Upon return, the caller begins execution immediately after the CALLRTN request. Only the information in the communication block may have changed. Return from the called tasks to the caller takes place automatically after the last called task ceases. A task waiting for return may be temporarily rolled from memory.

The format is:

COBOL

ENTER CALLRTN USING task₁ task₂...task_n.

task ₁	Parameter whose value is a one-to seven-character alphanumeric task name. The tasks are scheduled by CALLRTN in the order specified. From one to five tasks can be specified.
-------------------	---

FORTRAN

CALL CALLRTN(task₁,task₂,...,task_n)

task ₁	Parameter whose value is a one-to seven-character alphanumeric task name, left-justified with either blank or binary zero fill to a word boundary. The tasks are scheduled by CALLRTN in the order specified. From one to five tasks can be specified.
-------------------	--

COMPASS

CALLRTN address

address Address of a list of from one to five task names. Each task name must be left-justified with binary zero fill to a word boundary. The list is terminated by a word of binary zeros. Tasks are scheduled by CALLRTN in the order they appear on the list.

NOTE

If task₁, specified on a CALLRTN request, makes a CALLRTN request, task₁ + 1 through task_n, specified on the original CALLRTN request, are not scheduled. (Refer to figure 2-4.)

The following diagrams illustrate CALLRTN processing.

In figure 2-2, TASK A, at subcontrol point X, executes a CALLRTN to TASK B. TAF will first load a copy of TASK B to an available subcontrol point (unless a serially reusable copy of TASK B is already at a subcontrol point or CM-resident). TAF will then copy the contents of TASK A's communication block (CB) to TASK B's communication block area. TASK B begins execution. When TASK B has completed execution, TAF will copy the communication block, which was probably modified by TASK B, back to TASK A and allow TASK A to continue execution at the instruction immediately following the CALLRTN command.

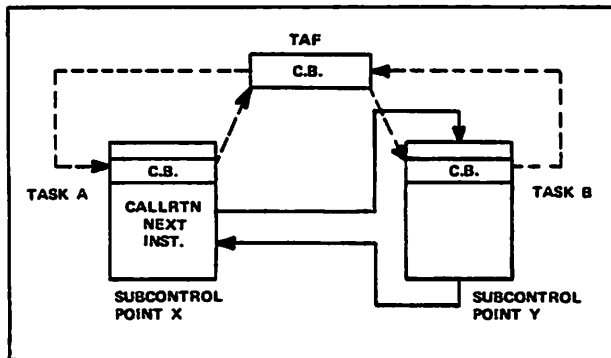


Figure 2-2. CALLRTN

The CALLRTN command allows a chain (list) of up to five tasks to be called. Figure 2-3 shows TASK A calling a chain of three tasks. As each task in the chain is completed, TAF will automatically pass the communication block to the next task in the chain and allow that task to execute. When the last task in the chain has completed, TAF will pass the

communication block back to the calling task and allow it to continue execution at the instruction immediately following the CALLRTN command.

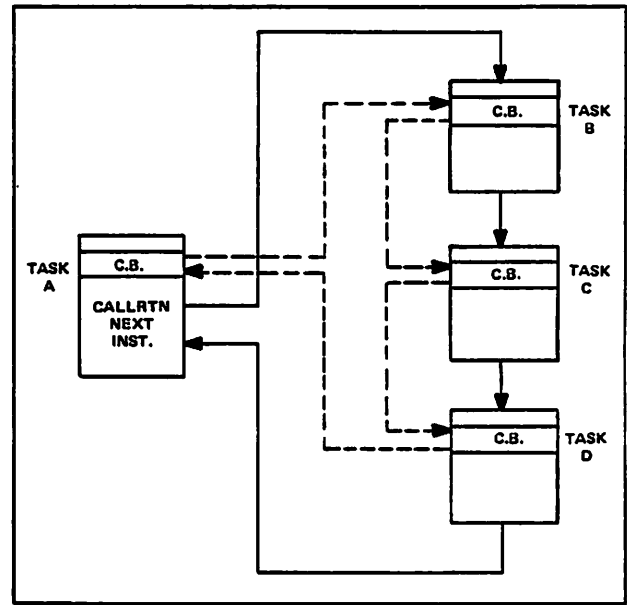


Figure 2-3. CALLRTN (Chain)

Figure 2-4 illustrates the processing referred to in the preceding note. The majority of the data inputs are handled by the TASK B-C-D chain. When an exception is found in TASK C, exception processing is invoked in the TASK E-F chain. Each new CALLRTN command cancels the previous one (TASK D is not executed), but TAF will always bring control back to the instruction immediately following the original CALLRTN command (TASK A). The communication block is passed from task to task as shown in figures 2-2 and 2-3.

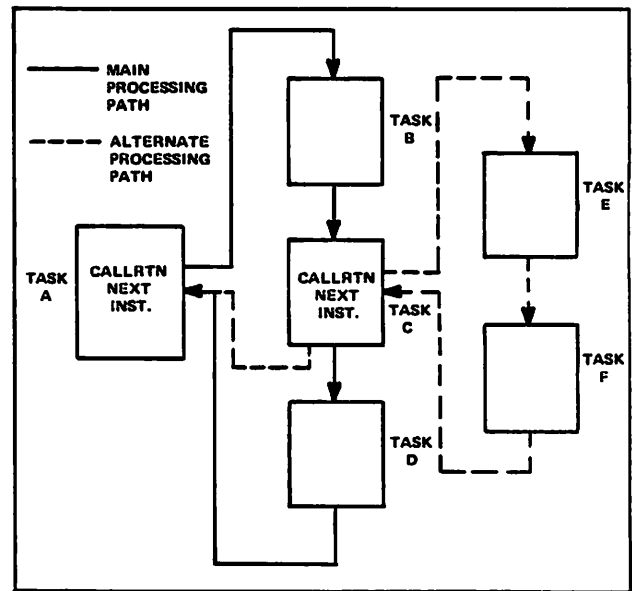


Figure 2-4. CALLRTN (Multiple Chains)

CALLTSK Request

The CALLTSK request is used to request that the transaction executive schedule a task or series of tasks (task chain). Included in this request is a parameter option that enables task branching or a CEASE function to be performed after the call is honored. If the task is not to cease on this call, a branching of the chain initiates a new chain that is independent of the original chain. The new chain is assigned a new sequence number for unique identification within the transaction subsystem (figure 2-1). The number of branches allowed by a transaction is controlled by an assembly constant MAXBW. If a CALLTSK request without a CEASE request is issued from a recoverable transaction, recovery may be affected. Refer to CALLTSK without CEASE and Recovery in section 7.

The format is:

COBOL

ENTER CALLTSK USING task code task₁...task_n.

task Parameter whose value is a one-to seven-character alphanumeric task name. This task is the next task to call in the present task chain (same sequence number) if code is set to zero; otherwise, it is the first task of a branch chain (the called task gets a different sequence number and proceeds independently of the calling task).

code Computational-1 item whose value is nonzero if the calling task is to continue after the request, initiating an independent task chain. If the value of this parameter is zero, the calling task ceases after this request. This parameter must be specified.

task₁ Parameter whose value is a one-to seven-character alphanumeric task name. From zero to four tasks can be specified. These tasks are scheduled after the one specified in the task parameter.

FORTRAN

CALL CALLTSK(task,code,task₁,...,task_n)

task Parameter whose value is a one-to seven-character alphanumeric task name, left-justified with either blank or binary zero fill to a word boundary. This task is the next task to call in the present task chain (same sequence number) if code is set to zero; otherwise, it is the first task of a branch chain (the called task gets a different sequence number and proceeds independently of the calling task).

code Integer whose value is nonzero if the calling task is to continue after the request, initiating an independent task chain. If the value of this parameter is zero, the calling task ceases after this request. If not specified, the default is zero.

task₁ Parameter whose value is a one-to seven-character alphanumeric task name, left-justified with either blank or binary zero fill to a word boundary. From zero to four tasks can be specified. These tasks are scheduled after the one specified in the task parameter.

COMPASS

CALLTSK address,cease

address Address of a list of from one to five task names. Each task name must be left-justified with binary zero fill to a word boundary. The list is terminated by a word of binary zeros. CALLTSK schedules the tasks in the order they appear on the list.

cease If specified, the calling task ceases after this request. If not specified, the calling task continues after the request. In this case the CALLTSK initiates an independent task chain.

The following figures illustrate CALLTSK processing.

Figure 2-5 shows TASK A making a CALLTSK with a CEASE request to TASK B. TAF will load a copy of TASK B to an available subcontrol point (this step is not necessary if a serially reusable copy of TASK B is already at a subcontrol point or is CM-resident). The communication block will be copied from TASK A to TASK B. TASK A's subcontrol point is released for use by other transactions, and TASK B is allowed to begin execution.

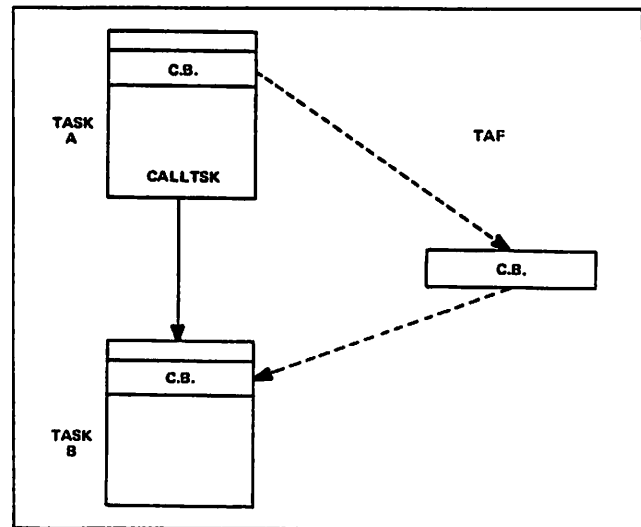


Figure 2-5. CALLTSK with CEASE Request

The CALLTSK without a CEASE request allows a new asynchronous chain to be started. In figure 2-6, TASK A uses this command to call TASK B. TAF will copy the contents of TASK A's communication block to a new communication block. A copy of TASK B will be loaded to an available subcontrol point (unless a serially reusable copy of TASK B is already at a subcontrol point or is CM-resident). The new communication block will be copied to TASK B. TASK A is allowed to continue execution and TASK B is allowed to begin execution.

NEWTRAN Request (COMPASS Only)

The NEWTRAN request initiates a new transaction chain with a unique sequence number and with the data base, user area, and user name set to zero in the communication block. The NEWTRAN request is designed for system use, primarily by ITASK, and is provided only for COMPASS.

The format is:

COMPASS

NEWTRAN address

address Address of a parameter table in the form shown in figure 2-7.

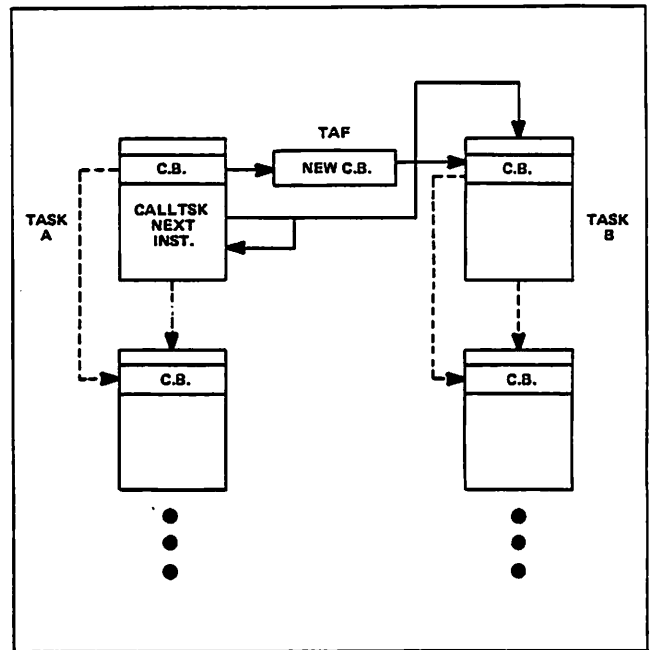


Figure 2-6. CALLTSK without CEASE Request

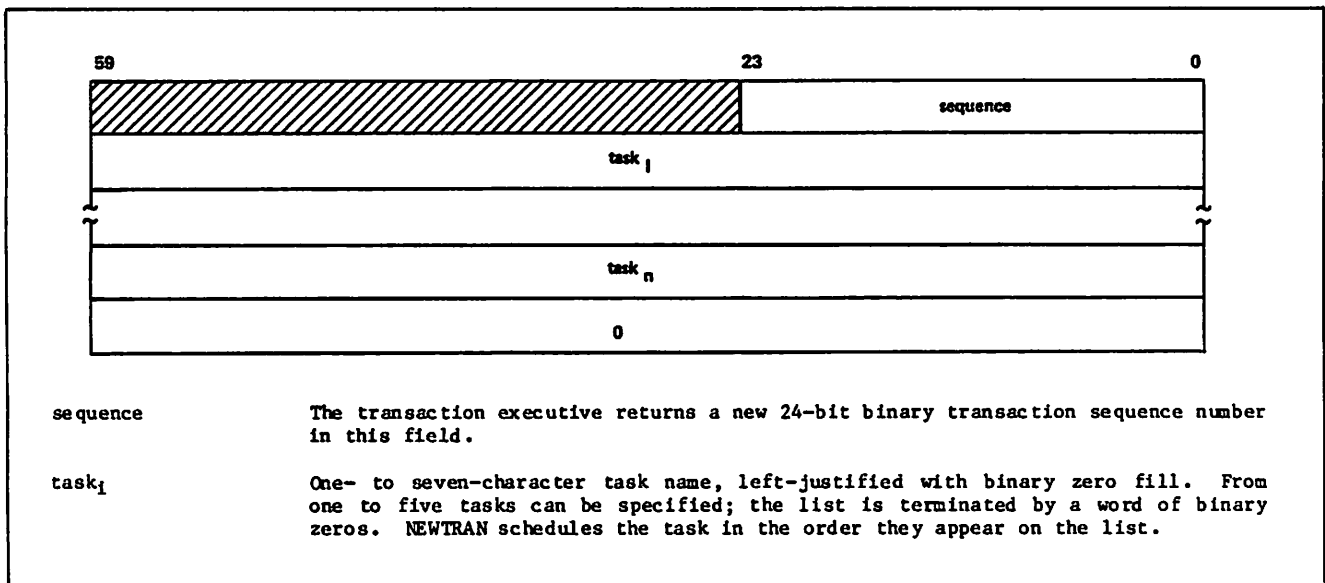


Figure 2-7. NEWTRAN Parameter Table

TRANCHK Request

The TRANCHK request determines the status of a transaction. This request is provided only for COMPASS.

The format is:

COMPASS

TRANCHK address

address Address of a parameter table in the form shown in figure 2-8.

INPUT/OUTPUT REQUESTS

The following requests are associated with input/output.

- WAITINP
Places terminal input data in the communication block. Equivalent to a read request.
- LOADCB
Reads data that overflowed the communication block.
- SETCHT
Sets character type for terminal input.

- BLDABH
Builds the application block header to use different character types, terminal format effectors, and auto input.
- GETABH
Returns the last application block header to the task.
- TERMDEF
Defines the operational controls or format of the terminal.
- SEND
Sends data to terminal. Equivalent to a write request.
- RELSCB
Releases memory used for extra communication blocks.
- TSIM
Reads terminal status table.
- TARO
Writes into terminal status table.

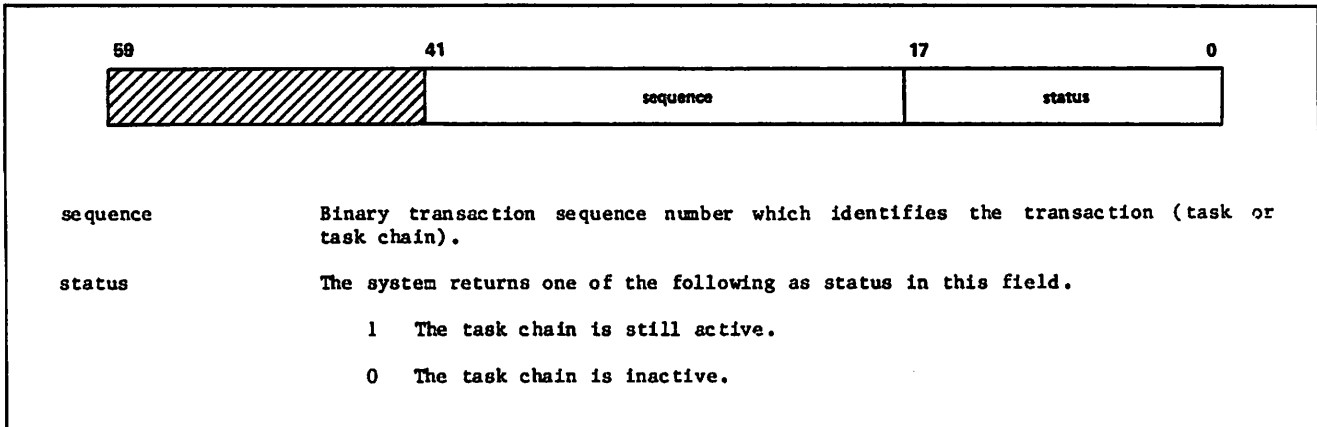


Figure 2-8. TRANCHK Parameter Table

- IIO
Limits total number of requests (RA+1 calls) a task can make.
- BWAITINP
Specifies that terminal input data is to be placed in a location in the task field length that is not reserved for the communication block.
- TPSTAT
Returns a value indicating that the active terminal subsystem is NAM.
- BEGIN
Specifies the area where the terminal transmission is to be placed in the task field length. It does not return the communication block to the first word address of the task unless the task requests.

WAITINP Request

The WAITINP request (wait for terminal input) reads data from a terminal. Internally, this request informs the transaction executive that the next input received from a terminal is to be passed directly to the requesting task and is not to initiate a new transaction. The data entered is placed in the message area of the communication block. The data held in the communication block prior to the WAITINP is lost.

The first parameter is a data item or variable whose value the system sets to indicate the reception of terminal input or an error condition. An optional second parameter of the request specifies a time limit, after which processing of the transaction is restarted even if no input has been received from the terminal.

NOTE

CM-resident tasks and tasks not associated with a terminal (batch origin) are not permitted to make a WAITINP request (task aborts).

At least one message (SEND) must be sent to the terminal before making a WAITINP request, and at least one SEND request from the original task chain must follow a message received following a WAITINP request.

No data manager requests may be active when a task makes a WAITINP request. All tasks that make WAITINP requests must be entered in the task library with the scheduling queue limit set to one. (Refer to section 10.)

WAITINP does not cause any data manager cease requests to be issued. Therefore, any resources locked or reserved by a transaction prior to a WAITINP continue to be locked or reserved during and after the WAITINP.

When a WAITINP request is processed, all communication blocks are released and the task is rolled out. All data in the communication block, except the transaction sequence number, will be lost, so the user should save any data to be preserved prior to the request. When input arrives from the terminal, a new communication block is generated, and the task is restarted. The new communication block will contain the transaction sequence number associated with the task prior to the WAITINP request. If the request does not return a normal status, the information in the communication block is meaningless.

The format is:

COBOL

ENTER WAITINP USING status time.

status Computational-1 item that the system sets to indicate the status of the WAITINP request. Upon completion of the request, status has one of the following values.

0 The request processed normally; terminal input appears in the normal input portion (begins in the third word) of the communication block.

< 0 Another task is waiting for input from the originating terminal. The WAITINP request must be reissued if input is desired.

> 0 The time delay elapsed before input was received from the terminal.

time Computational-1 item that contains the time limit, in decimal seconds, to wait for input. If not specified, the default is 480 seconds. The maximum value is 2048 seconds.

FORTTRAN

CALL WAITINP(status,time)

status Integer variable that the system sets to indicate the status of the WAITINP request. Upon completion of the request, status has one of the following values.

0 The request processed normally; terminal input appears in the normal input portion (begins in the third word) of the communication block.

< 0 Another task is waiting for input from the originating terminal. The WAITINP request must be reissued if input is desired.

> 0 The time delay elapsed before input was received from the terminal.

time Integer time limit, in decimal seconds, to wait for input. If not specified, the default is 480 seconds. The maximum value is 2048 seconds.

COMPASS

WAITINP time

time Time limit, in decimal seconds, to wait for input. If not specified, the default is 480 seconds. The maximum value is 2048 seconds.

Status of the WAITINP macro is determined by the following:

- Word 2 of the communication block contains 1 if the time delay elapsed before input was received from the terminal.
- The X6 register contains a negative value if another task is waiting for input from the originating terminal. The WAITINP request must be reissued if input is desired. (Word 2 of the communication block should be checked before X6 is checked.)
- The X6 register is set to zero if the request processed normally; terminal input appears in the normal input portion (begins in the third word) of the communication block. (Word 2 of the communication block should be checked before X6 is checked.)

LOADCB Request

If the transaction input data exceeds the message area of the communication block, bit 12 of word 1 of the communication block (the *i* field) is set. The LOADCB request loads the remainder of the input message into a user-defined buffer. This buffer is a single contiguous block of storage for all additional communication blocks. No headers or trailers appear in this area.

The maximum input size is determined by an installation parameter.

The data that remains after the communication block message area fills (referred to here as overflow data) is transferred to the user-defined buffer. If the LOADCB request is made with no data overflow (*i* field not set) or if a RELSCB request was made prior to the LOADCB request, no data is transferred and the status word is set to zero. If the overflow data consists of more words than will fit in the buffer, the buffer is filled, and the status word is set to a value greater than the number of words in the buffer. If the overflow data can be contained in the buffer, the overflow data is transferred to the buffer, and the number of words transferred is set in the status word.

The overflow data can be requested as many times during a transaction as desired. Each time, the overflow data is transmitted as specified previously.

The format is:

COBOL

ENTER LOADCB USING buffer status r.

buffer A 01-level item identifying the user-defined buffer area.

status Computational-1 item that the system sets to indicate the LOADCB request status. The significance of the value set has been described earlier in the LOADCB request description.

r Computational-1 item. If a non-zero value is entered for this parameter, any message data that overflows the LOADCB buffer specified with the buffer parameter is released (same as RELSCB).

FORTRAN

CALL LOADCB(buffer,length,status,r)

buffer An array identifying the user-defined buffer area.

length Integer length of the array specified with the buffer parameter in words.

status Variable that the system sets to indicate the LOADCB request status. The significance of the value set has been described earlier in the LOADCB request description. This value is also returned as the value of the function, if LOADCB is called as a FORTRAN function.

r Integer variable. If a nonzero value is entered for this parameter, any message data that overflows the LOADCB buffer specified with the buffer parameter is released (same as RELSCB).

As stated in the description of the status parameter, the LOADCB request can be referenced as a FORTRAN function. The format of the function is LOADCB(buffer,length,status,r); the parameters are the same as those already described. The value of the function is the value of the status parameter.

COMPASS

LOADCB address

address Address of a parameter table in the format shown in figure 2-9.

The LOADCB macro status is returned in the X6 register. The significance of the value returned has been described earlier in the LOADCB request description.

SETCHT Request

The SETCHT request allows a task to change the application character type (ACT) of terminal messages placed in the communication block.

NOTE

This is an optional request; the default character type is 6-bit display code characters.

The following character types are available.

- 7-bit ASCII, seven and one-half characters per word.
- 7-bit ASCII characters, right-justified in 12-bit bytes, five characters per word.
- 6-bit display code characters, 10 characters per word.

This request causes TAF to send a change input character type supervisory message to NAM. For more information about supervisory messages, refer to the NAM/CCP Reference Manual.

NOTE

This request does not change the data currently held in communication blocks for the task. For example, a LOADCB request gets data in the former character type, while the next WAITINP request gets data in the new character type.

If 7-bit ASCII characters are used to originate transactions, ITASK transaction tables must be set up in terms of ASCII and not display code.

The format is:

COBOL

ENTER SETCHT USING username status act.

username A 01-level item containing a one- to seven-character user name, left-justified with blank fill. The user name may also be specified with a computational-1 item with binary zero fill. If the item contains only binary zeros, the character type applies to the originating terminal. This user name must be a name appearing in an NCTF1 file. A user name consisting of the character P with binary zero fill cannot be specified.

status Computational-1 item whose value is set by TAF to indicate user name status. This value is zero if the user is not logged in and one if the request is processed normally. The task is aborted if the user name is not defined or if the data base for the specified user does not match the data base of the originating user.

act A 01-level item containing one of the following words, left-justified.

ASCII7 7-bit ASCII. Seven and one-half characters per word.

ASCII5 7-bit ASCII, right-justified in 12-bit bytes, five characters per word.

DISPLAY 6-bit display code.

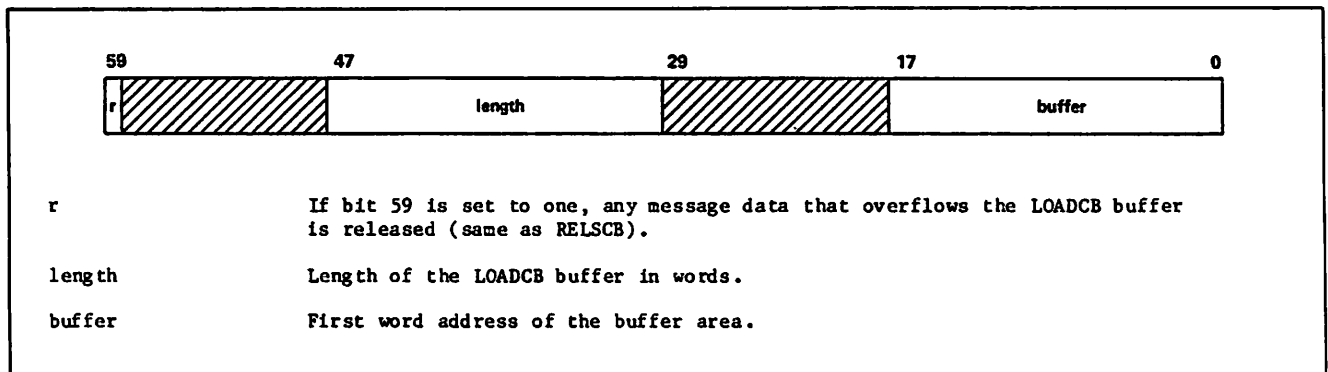


Figure 2-9. LOADCB Parameter Table

FORTRAN

CALL SETCHT(username,status,act)

username One- to seven-character user name, left-justified with zero or blank fill. If the terminal name is zero, the character type applies to the originating terminal. This user name must be a name appearing in an NCTF1 file.

status Unnormalized floating-point number whose value specifies the user name status. This value is zero if the user is not logged in and one if the request is processed normally.

The task is aborted if the user name is not defined or if the data base for the specified user does not match the data base of the originating user.

act Parameter containing one of the following words, left-justified with binary zero or blank fill.

ASCII7 7-bit ASCII, seven and one-half characters per word.

ASCII5 7-bit ASCII, right-justified in 12-bit bytes, five characters per word.

DISPLAY 6-bit display code.

COMPASS

SETCHT address

address Address of a parameter table in the form shown in figure 2-10.

BLDABH Request

The BLDABH request builds or modifies the application block header (ABH). This header accompanies every block passing between TAF and NAM, and thus is used during communication between TAF and terminals. It contains detailed information describing the block it accompanies.

NOTE

This is an optional request. The default application block header specifies 6-bit display code characters, nontransparent mode, no format effectors, and no auto input.

To use some of the NAM features, a task must use the BLDABH request to modify the ABH. Some of the features provided via the ABH are:

- Character type specification of messages sent to a terminal. Either 7-bit ASCII or 6-bit display code characters may be used.

- Transparent mode specification. Transparent mode inhibits all data transforms and allows the task direct access to the real terminal features.
- Terminal format effector specification. Format effectors control printing (or the cursor on display terminals).
- Auto-input mode specification. Auto-input mode allows a fill in the form type of transaction. This is provided since the first n characters of a message output to the terminal via the SEND request (ABT=2 only) can be used to form the first n characters of subsequent input from the terminal. If n is greater than 20, only the first 20 characters are returned, followed by the entered text.

It is not necessary to use the BLDABH request before every SEND. It is needed when a selectable parameter on the ABH for the next SEND differs from a setting on the ABH for the previous SEND. In other words, a setting of a parameter once selected by BLDABH stays in effect for that task unless changed by a subsequent BLDABH request. If a task issues a BLDABH request and then calls another task, the new task must also issue a BLDABH request because the request parameters are not passed to it. If a task is to be reusable, the task should initialize the ABH if the default values have been changed.

Default values are supplied on the ABH. If the default values are the desired values, no BLDABH request is required. Default values have been selected to remain compatible with prior versions of the transaction subsystem.

The format is:

COBOL

ENTER BLDABH USING field₁ value₁...field_n value_n.

field₁ A 01-level item containing a three-character, left-justified field for which a value is to be set. field₁ can be binary zero or blank-filled. The possible values are listed in table 2-1.

value₁ A computational-1 item whose value is set for the corresponding field₁. The field₁ and value₁ parameters must appear in pairs. The possible values and descriptions are listed in table 2-1.

FORTRAN

CALL BLDABH(field₁,value₁,...,field_n,value_n)

field₁ Variable containing a three-character, left-justified field for which a value is to be set. field₁ can be binary zero or blank-filled. The possible values are listed in table 2-1.

value₁ Integer that is set for the corresponding field₁. The field₁ and value₁ parameters must appear in pairs. The possible values and descriptions are listed in table 2-1.

Example:

COMPASS

The following request builds the ABH with ACT=4, NFE=0, and AIM=1.

CALL BLDABH(3LACT,4,3LNFE,0,3LAIM,1)

The COMPASS user specifies changes to the ABH with the SEND request (described later in this section).

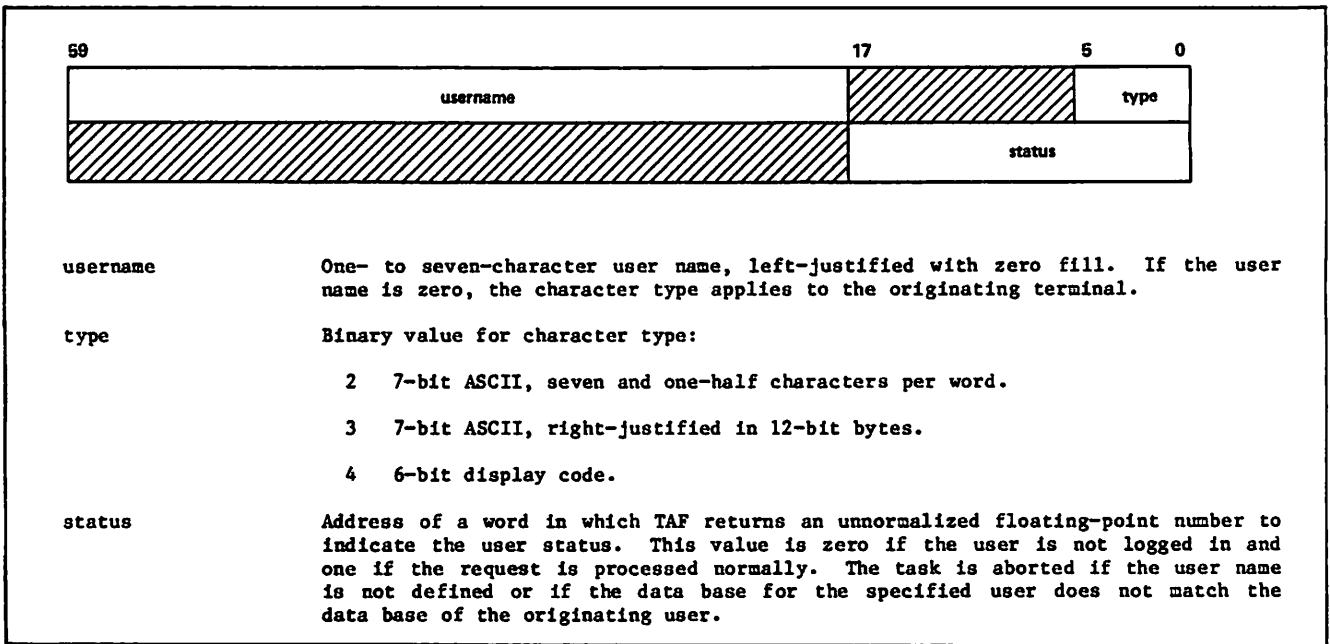


Figure 2-10. SETCHT Parameter Table

Table 2-1. BLDABH Request Parameters

field _i	value _i	Description	field _i	value _i	Description	
ACT	2	Application character type.	AIM	1†	No format effectors in the message.	
		7-bit ASCII characters, seven and one-half per word.		0	No auto-input mode.	
		7-bit ASCII characters, right-justified in 12-bit bytes, five per word.		1†	Auto-input mode selected.	
NFE	4†	6-bit display code characters, 10 per word.	XPT	0†	Transparent mode. Applies only to 7-bit ASCII character types.	
		Format effectors in message.		1	Interactive virtual terminal data transforms to console devices are performed. (This option must be selected for output to batch devices.)	
	0	Format effectors appear in the message. (Refer to the NAM/CCP Reference Manual.)			1	IVT transforms are inhibited.

†These are default values supplied by TAF, FORTRAN, and COBOL library routines for BLDABH and SEND. These are also the default values for the COMPASS SEND request when the task does not specify the ABH.

GETABH Request

The GETABH request allows a task to examine the ABH that accompanied the last message.

NOTE

This is an optional request. Terminal input and output can be done without using it.

If no SEND has been issued since the last input from the terminal, the GETABH request allows the task to obtain the ABH that accompanied the input block from the network. If a SEND has been issued, the ABH used for the SEND is obtained. This request is useful for obtaining the character type, character count, and parity error fields.

NOTE

The parity error indicators can also be obtained from the communication block, but it may be easier for the COBOL task to use GETABH to obtain this indicator.

The format is:

COBOL

ENTER GETABH USING username status field₁
value₁...field_n value_n.

username A 01-level item containing a one-to seven-character user name, left-justified with blank fill. The user name may also be specified with a computational-1 item with binary zero fill. If the item contains only binary zero, the ABH for the originating terminal is returned. This user name must be a name appearing in the NCTF1 file. A user name consisting of the character P with binary zero fill cannot be specified.

status Computational-1 item whose value is set by TAF to indicate the status of the user name. status contents is zero if the user is not logged in and is one if the

request is processed normally. The task is aborted if the user name is not defined or if the data base for the user name does not match the data base of the originating user.

field₁ A 01-level item containing a three-character field for which a value is desired, left-justified with binary zero or blank fill. The possible fields and values are described in table 2-2.

value₁ Computational-1 item in which the value for the desired field is to be returned. The field₁ and value₁ items must appear in pairs.

FORTRAN

CALL GETABH(username,status,field₁,value₁,...,
field_n,value_n)

username Parameter whose value is a one-to seven-character user name, left-justified with binary zero or blank fill. If the user name is binary zero, the ABH for the originating user is returned.

status Unnormalized floating-point number whose value is set by TAF to indicate the user name status. status contents is zero if the user is not logged in and greater than zero if the request is processed normally. The task is aborted if the user name is not defined or if the data base for the user name does not match the data base of the originating user.

field₁ Variable containing a left-justified, three-character field for which a value is desired. The possible fields and values are described in table 2-2.

value₁ Name of an unnormalized floating-point variable in which TAF returns the value of the desired field.

COMPASS

GETABH address

address Address of a parameter table in the format shown in figure 2-11.

Table 2-2. Application Block Header

field _i	Bits	value _i	Description	field _i	Bits	value _i	Description	
ABT	59-54		Application block type.	XPT	14		Transparent bit.	
		0	Null input block.			0	Nontransparent data; IVT transforms are used.	
		1	Data block; not the last (applies to SEND only).			1	If application character type is set to two or three, IVT transforms are inhibited. If ACT is set to four, this bit is ignored on output.	
		2	Data block; last or only block.					
ADR	53-42		Addressing information; TAF assigns this area for output, NAM for input.				For input, if XPT is set to one and ACT is set to four, TAF changes ACT to three. You must change ACT back to four when no more transparent input is expected.	
ABN	41-24		Application block number. This area is user- or TAF-assigned for SEND requests. You assign this area causing the block parameter on the SEND request.	BIT	12		For input, parity error flag; for output, auto-input mode flag.	
ACT	23-20		Application character type.				0	No parity error, no auto-input mode.
		2	7-bit ASCII characters, seven and one-half per word.				1	For input, a parity error was detected; for output, use of auto-input mode is specified.
		3	7-bit ASCII characters, right-justified in 12-bit bytes, five per word.					
		4	6-bit display code characters, 10 per word.	TLC	11-0		Text length in units specified by ACT. This field is assigned by TAF or by you for output. NAM assigns this field for input. The text length includes the unit separator. A unit separator is 12 through 66 bits of zero for 6-bit display code characters and US for 7-bit ASCII code characters.	
NFE	15		Format effectors.					
		0	Format effectors are present in the SEND message.					
		1	Format effectors are not present in the SEND message.					

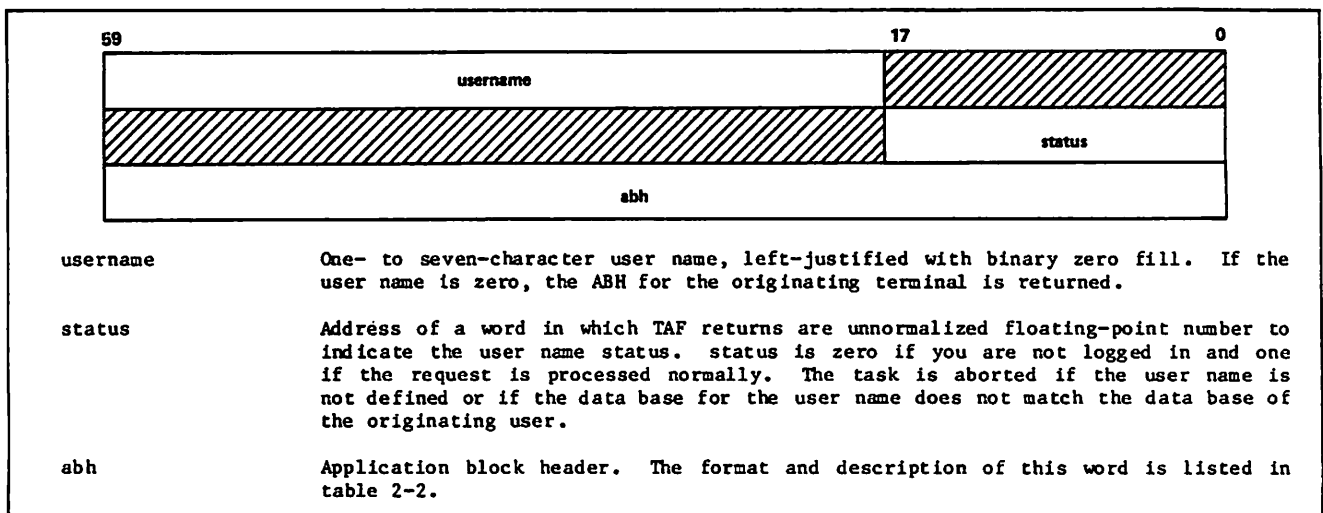


Figure 2-11. GETABH Parameter Table

TERMDEF Request

The TERMDEF request can be used by a task to define or modify terminal characteristics, where those of the connected real terminal differ from the assumptions made for the network's IVT model. This request can also be used to tailor the operational controls of the terminal.

NOTE

This is an optional request. Terminal input and output can be done without it. Each terminal that logs into the network is associated with a default set of terminal characteristics. (Refer to appendix G.)

A complete list of terminal characteristics is provided in the NAM/CCP Reference Manual and appendix G of this manual. Characteristics that may be modified include:

- Terminal physical page width.
- Number of physical lines on a terminal page.
- Specification of a character used to cancel an input logical line in progress.
- Specification of a character to be used as a break character. User breaks are sent to the application task for processing.
- Set transparent input mode. This setting exists for only one logical input line.

The format is:

COBOL

```
ENTER TERMDEF USING username status field1
field2...fieldn.
```

username A 01-level item containing a one- to seven-character user name, left-justified with blank fill. The user name may also be specified with a computational-1 item with binary zero fill. If the item contains only binary zero, the terminal definition specified applies to the originating terminal. This user name must be a name appearing in an NCTFI file. A user name consisting of the character P with binary zero fill cannot be specified.

status Computational-1 item that TAF sets to indicate user name status. status contents is zero if the user is not logged in and is one if the request is processed normally. The task is aborted if the user name is not defined or if the user's data base does not match the data base of the originating user.

field₁

A left-justified 01-level item with binary zero or blank fill, containing the characteristic (for example, PW=60, PL=35, or CN=2B). A maximum of 17 nonblank, nonbinary zero characters is allowed. The characteristics are described in the NAM/CCP Reference Manual and appendix G of this manual. If the NAM/CCP Reference Manual specifies x or h, a 6-bit display code alphanumeric representation is required (where x is a 7-bit ASCII character and h is a hexadecimal character).

FORTRAN

```
CALL TERMDEF(username,status,field1,
field2,...,fieldn)
```

username Parameter whose value is a one-to seven-character user name, left-justified with binary zero or blank fill. If the user name is zero, the terminal definition specified applies to the originating terminal. This user name must be a name appearing in an NCTFI file.

status Variable in which TAF places an unnormalized floating-point number to indicate user name status. status contents is zero if the user is not logged in and greater than zero if the request is processed normally. The task is aborted if the user name is not defined or if the data base for the user does not match the data base of the originating user.

field₁

A left-justified variable with binary zero or blank fill, containing the characteristic (for example, PW=60, PL=35, or CN=2B). A maximum of 17 nonblank, nonbinary zero characters is allowed. The characteristics are described in the NAM/CCP Reference Manual and appendix G of this manual. If the NAM/CCP Reference Manual specifies x or h, a 6-bit display code alphanumeric representation is required (where x is a 7-bit ASCII character and h is a hexadecimal character).

Example:

The following request designates the page width to be 60 characters, the page length to be 40 lines, and the plus character (ASCII representation is 2B) to be the cancel character for user USER5; it then assigns the status to variable STATUS.

```
CALL TERMDEF(5HUSER5,STATUS,5LPW=60,
5LPL=40,5LCN=2B)
```


Example:

The following statements redefine the transparent input mode text delimiter to be a slant (ASCII representation is 2F) and the character count delimiter to be 4096.

```
DATA MSG/10HDL=X2F,C40,2L96/
.
.
.
CALL TERMDEF(SHUSER5,STATUS,MSG)
.
.
.
END

COMPASS
TERMDEF address
```

address Address of a parameter table in the format shown in figure 2-12.

SEND Request

The SEND request sends a message to a user; that is, it performs a write operation. If the task does not specify the user to which the message should be sent, it is directed to the terminal that originated the transaction. At least one SEND request to the originating terminal must occur from the primary task chain for each message received. The task can also specify that the task be ended when the message is sent.

There is both a limit on the total output that one task can send (MAXTO),† and a limit on the output one task can send before it is rolled out, pending the transfer of the output to the terminal or terminals (MLIM).† Also, the maximum output length for one SEND request is determined by the MAXWS† installation parameter.

A task can specify the status parameter in COBOL and FORTRAN or the r parameter in COMPASS so that the task will wait until the network acknowledges that the message has been delivered to the user. If recall is selected, any error detected on the output message causes the task performing the SEND request to be restarted. Therefore, tasks performing SEND requests with recall should check status and compensate for errors. Use of recall can simplify recovery since the task issuing the SEND request is informed of the problem at the first address after the SEND. When recall is selected, the task is rolled out if the message length exceeds the MLIM installation parameter. Any data manager resources held by the task, such as locked records, remain assigned for the duration of the rollout. If reserving these resources for this period of time will impact the performance of the application, tasks should release all file and record locks before executing the SEND request.

If recall is not selected, ITASK is informed of the problem, rather than the task that issued the request. ITASK is informed because the issuing task may have already terminated when the message delivering problem is detected.

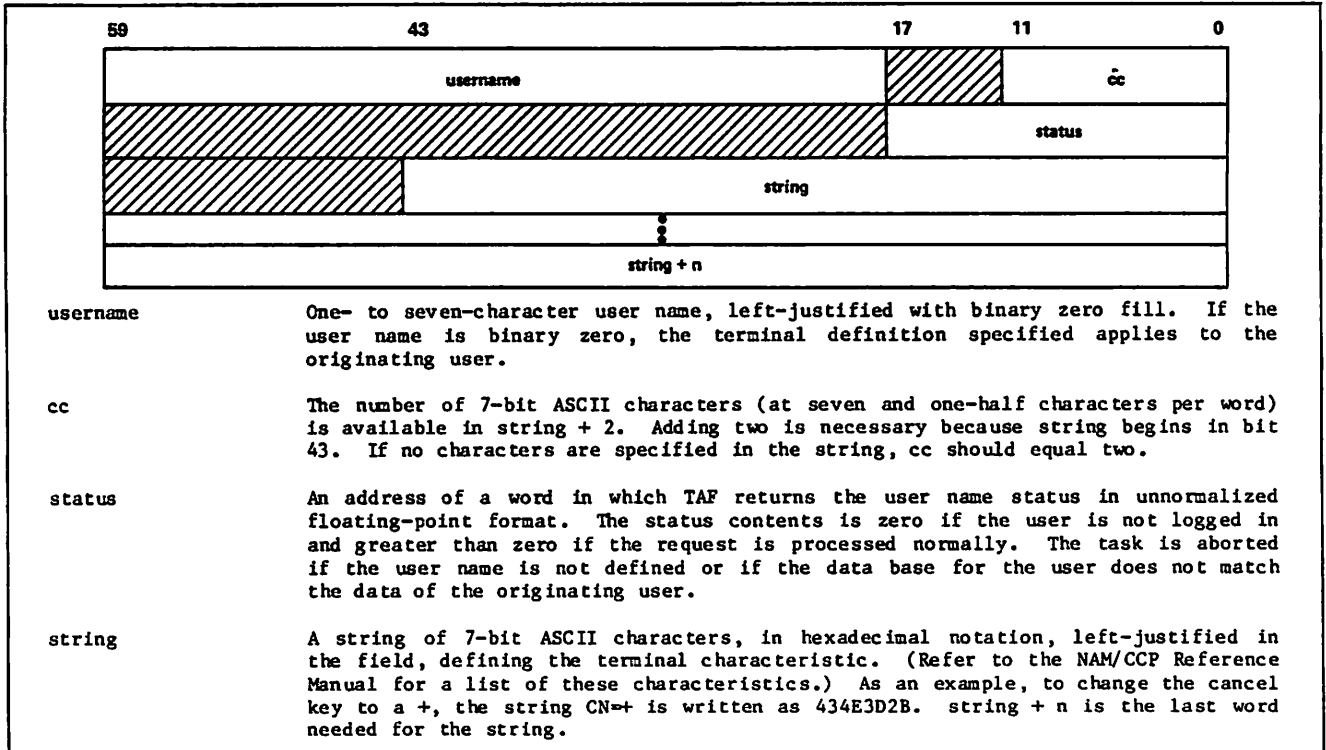


Figure 2-12. TERMDEF Parameter Table

†Refer to the NOS 2 Installation Handbook (refer to the preface for the publication number).

The format of the message on the terminal depends on the terminal class and the format effectors in the message. For example, on synchronous terminals the cursor for entering the next message to the host (upline) is always the first character of the line. On asynchronous terminals the cursor follows the last character of output; this assumes no format effectors are in the message. The NAM/CCP Reference Manual contains complete information on terminal protocols.

The format is:

COBOL

ENTER "SEND" USING messageaddress username
 cease-flag output-flag block status length.

The parameters are described in table 2-3.

NOTE

Under COBOL, SEND must be enclosed in quotation marks, because SEND is a reserved word for that compiler.

FORTRAN

CALL SEND(messageaddress,length,username,
 cease-flag,output-flag,block,status)

The parameters are described in table 2-4.

COMPASS

SEND address

address Address of a parameter table in the format shown in figure 2-13. This figure and the parameters in table 2-5 should be used with table 2-6. This table shows interrelationships between SEND parameters. Setting parameters shown in the top portion of table 2-6 results in the corresponding events shown in the bottom portion.

The entire four-word block must be present, even if the last three words are not used.

Table 2-3. COBOL SEND Request Parameters (Sheet 1 of 2)

Parameter	Data Type	Value	Description
messageaddress	01-level display item		Data item containing the message to be sent. For 7-bit ASCII character types, the message must have an end-of-line byte of the single character US for all logical lines. (Refer to the NAM/CCP Reference Manual for NAM data formats.) The TAF SEND interface supplies end-of-line bytes for 6-bit display code character types only.
username	01-level display item, or computational-1		Data item containing a one- to seven-character user name. If this parameter is omitted or specified as COMP-1 VALUE ZERO, the message is sent to the originating terminal. This user name must be a name appearing in the NCTF1 file and must be validated for the same data base as the sending task.† A user name consisting of the character P with binary zero fill cannot be specified. The user name can be binary zero or blank-filled.
cease-flag	Computational-1	0, omitted	Task end flag. The task continues after the message is sent.
		1	The task ends after the message is sent.

†Tasks associated with application identifier SY can SEND to any valid TAF user.

Table 2-3. COBOL SEND Request Parameters (Sheet 2 of 2)

Parameter	Data Type	Value	Description
output-flag	Computational-1		Application block type. Currently serves no function. Retained for compatibility with previous releases.
block	Computational-1	Nonzero	TAF uses the value contained in the data name as the ABN.
		0	TAF assigns the ABN and returns this as the value in this data name.
		Omitted	TAF assigns the ABN, and the value is not returned.
status	Computational-1		Recall flag; this parameter applies only if the cease-flag is set to zero.
		0, omitted	The task begins processing after the block is transmitted to the network.
		Nonzero	The task begins processing after the block has been delivered to the user and acknowledgment is received. Each nonzero value indicates the status of the block. The network function, subfunction, and description are given here. For all values other than 1, the block was not delivered to the user and the task must recover.
		1	Block delivered (FC/ACK).
		2	Block not delivered (INTR/USR/RC-1).
		3	User break 1 (INTR/USR/RC-1).
		4	User break 2 (INTR/USR/RC-1).
		10	Terminal not logged in (CON/END).
length	Computational-1	11	Correspondent unavailable or failed, line disconnect (CON/CB/RC-1).
			Length of the message in characters. This length should include end-of-line bytes for 7-bit ASCII character types. For 6-bit display code output, the COBOL compiler-generated length is used if the length parameter is omitted.

Table 2-4. FORTRAN SEND Request Parameters (Sheet 1 of 2)

Parameter	Data Type	Value	Description
messageaddress	Hollerith or left-justified, zero fill		Array name containing the message to be sent. For 7-bit ASCII character types, the message must have an end-of-line byte of the single character US for all logical lines. (Refer to the NAM/CCP Reference Manual for NAM data formats.) The TAF SEND interface supplies end-of-line bytes for 6-bit display code character types only.
length	Integer		Length of the message in characters. This length should include the end-of-line bytes for 7-bit ASCII character types.
username	Left-justified, zero or blank fill integer		One- to seven-character user name. This must be a name appearing in the NCTF1 file, which is validated for the same data base as the task issuing the send.† If this parameter is omitted or specified as binary zero, the message is sent to the originating terminal.
cease-flag	Integer	0, omitted	The task continues after the message is sent.
		1	The task ends after the message is sent.
output-flag	Integer		Application block type. Currently serves no function. Retained for compatibility with previous releases.
block	Integer		Application block number.
		Nonzero	TAF uses the value of this variable as the ABN.
		0	TAF assigns the ABN and returns the value in this variable. For this parameter, you should enter an integer variable assigned the value zero instead of entering zero, as follows: <pre> INTEGER ABN ABN=0 CALL SEND(messageaddress,length,username, cease-flag,output-flag,ABN,status) </pre>
status	Unnormalized floating point	Omitted	TAF assigns the ABN, and the value is not returned.
			Recall flag (applies only if cease-flag is set to zero.
		0, omitted	The task begins processing after the block is transmitted to the network.
		Nonzero	The task begins processing after the block has been delivered to the terminal and acknowledgment is received. Each nonzero value indicates the status of the block. The network function, subfunction, and description are given here. For all values other than 1, the block was not delivered to the terminal and the task must recover.
		1	Block delivered (FC/ACK).
2	Block not delivered (FC/NAK/RC=1).		
3	User break 1 (INTR/USR/RC-1).		
4	User break 2 (INTR/USR/RC-1).		

†Task associated with application identifier SY can SND to any valid TAF user.

Table 2-4. FORTRAN SEND Request Parameters (Sheet 2 of 2)

Parameter	Data Type	Value	Description
		10	Terminal not logged in (CON/END).
		11	Correspondent unavailable or failed, line disconnect (CON/CB/RC=1).

Table 2-5. COMPASS SEND Request Parameters (Sheet 1 of 2)

Parameter	Bits	Value	Description
t	59		Terminal flag.
		0	Message is sent to originating terminal.
c	58	1	Message is sent to specified user.
			Task cease flag.
o	57	0	The task cease is not performed.
		1	The task cease is performed following the SEND request.
b	56		No function. Retained for compatibility with previous releases.
			Return block flag.
r	55	0	If a is set to one, the task assigns the block number via the abn field. If a is set to zero, TAF assigns the block number, and the number is not returned in the block field.
		1	TAF assigns the block number and returns this number in the block field.
a	54		Recall flag.
		0	TAF does not wait for the block to be delivered to the terminal.
a	54	1	TAF waits for the block to be delivered to the terminal.
			Application block header flag.
a	54	0	TAF generates the application block header but does it in its own area without making use of or changing word 3 of the parameter table shown in figure 2-8. (Refer to the description of the BLDABH request for the default values TAF uses.) Because a BLDABH request is not available to COMPASS tasks, the header TAF supplies when a=0 always has default values, except for a tlc influenced by numwords. Specifically, if a is set to zero, the task can only output in 6-bit display code, cannot use format effectors, cannot use auto-input mode, and cannot preset the application block number.
		1	The task supplies the application block header. (The task supplies the ABH in the third word SEND parameter table shown in figure 2-8, with parameters shown in table 2-6.)

Table 2-5. COMPASS SEND Request Parameters (Sheet 2 of 2)

Parameter	Bits	Value	Description
messageaddress	47-30		Address of message. For 6-bit display code character type, the message must contain an end-of-line byte from 12 to 66 bits of zero. For 7-bit ASCII characters, the message must have an end-of-line byte of the single character US for all logical lines in the output block. (Refer to the NAM/CCP Host Application Programmers Reference Manual.)
numwords	17-0		Number of words in the message. If the a field is set to one, this field is not used. The tlc field must be used to specify the message length. The numwords field must contain the unit separator.
username	59-18		User name from one to seven characters with binary zero fill. If the t field is set to one, this parameter must be specified. If the t field is set to zero, this field is ignored.
block	17-0		Application block number. If the b field is set to one, TAF returns the application block number in this field.
abt†	59-54		Application block type. This parameter is ignored by TAF. It is retained for compatibility with previous systems.
		1	Indicates that this is not the last data block of the message.
		2	Indicates that this is the last or only data block of the message.
abn†	41-24		Application block number. If the b field is set to one, TAF assigns and stores a value in the block field. If the b field is set to zero, this value is stored in block.
act†	23-20		Application character type.
		2	7-bit ASCII characters, seven and one-half per word.
		3	7-bit ASCII characters, right-justified in 12-bit bytes, five per word.
		4	6-bit display code characters, 10 per word.
nfe†	15		Format effectors.
		0	Format effectors appear in the SEND message. Refer to the NAM/CCP Host Application Programmers Reference Manual for information concerning format effectors.
		1	No format effectors appear in the SEND message.
xpt†	14		Transparent bit.
		0	Nontransparent data. IVT transforms are used.
		1	If the act field is set to two or three, IVT transforms are inhibited. If the act field is set to four, this bit is ignored.
aim†	12		Auto-input mode.
		0	No auto-input mode.
		1	Auto-input mode enabled.
tlc†	11-0		Text length in units specified by the act field. If an output block contains multiple logical lines, all logical lines must contain end-of-line bytes, which must be included in the character count.
status	59-0		Status of the user name. This field is used only if the r field is set to one. The value of this field is a supervisory message returned by the network. (Refer to the NAM/CCP Host Application Programmers Reference Manual for a description of the supervisory messages.)

†These parameters apply only if the a field is set to one.

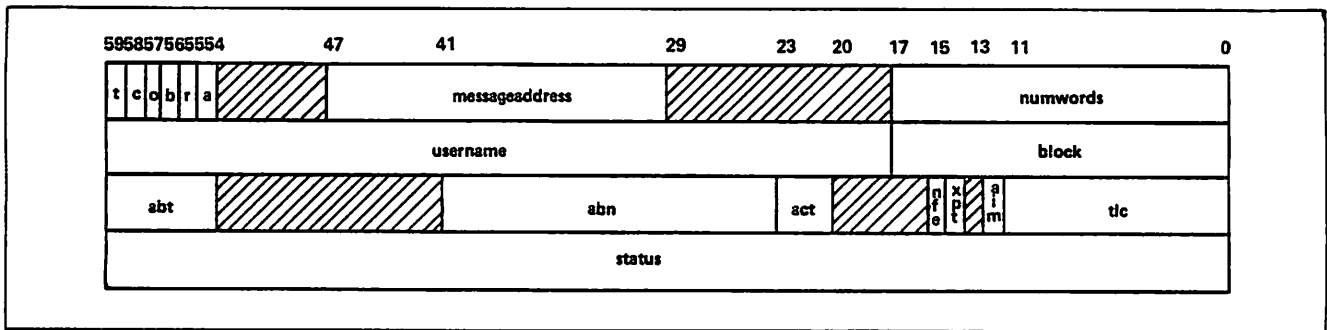


Figure 2-13. SEND Parameter Table†

RELSCB Request

The RELSCB request releases all overflow communication blocks†† to the system. After an RELSCB request is processed, subsequent LOADCB requests return a status word of zero.

The format is:

COBOL

ENTER RELSCB.

There are no parameters.

FORTRAN

CALL RELSCB

There are no parameters.

COMPASS

RELSCB

There are no parameters.

TSIM Request

The TSIM request provides limited interrogation of the terminal status table. This table, in central memory, is derived from the network file (refer to section 9). The TSIM request can search four fields in the table.

- Data base name field
- User argument field
- Communication line field
- User name field

The two-word TST entry of the users that meet a defined criterion (criterion) is returned to the user. The field specified by code is examined in each terminal table entry by taking the logical product of the field and mask to isolate a portion of the field. The logical difference of this product and criterion value is then obtained. If this result is zero, the terminal entry is placed in list. No action occurs if it is not zero; that is,

the user does not meet the specified criterion. For example, an application could use this to obtain a list of the users that are allowed to use a particular data base by selecting the data base field to be searched for a certain name.

If the task data base is not the same as the one being interrogated, then the list equals zero.

The format is:

COBOL

ENTER TSIM USING code mask criterion rlength list length.

code Computational-1 item whose value indicates which field is to be searched:

Value	Field
0	Data base name field.
1	User argument field.
2	Communication line field.
3	User name field.

mask Parameter whose value is taken as a binary mask. It is used in compiling the list of user names.

criterion Criterion value for the search.

rlength Computational-1 item whose value TAF sets to indicate the number of entries found.

list Item in which TAF lists found entries. If zero or absent, no list is returned but rlength is given as the number of found entries.

length Computational-1 item whose value indicates the number of words that list can hold. If zero or absent, no list is returned but rlength is given as the number of found entries. Only the lowest order 12 bits of this value are used.

†The parameters are described in table 2-5.
 ††Described under LOADCB Request.

Table 2-6. COMPASS SEND Parameter Interrelationships

Field	Value												
	0	1	0	1	0	0	1	0	1	0			
a t b act r	0	1	0	1	0	0	1	0	1	0			
Function	Resultant												
MSG terminator (last or only block) or BLK terminator (not the last block) is determined by:	TAF (o field is ignored)	abt field	Originating terminal	Terminal of user username	abn field	TAF	Not returned	TAF	Returned in blk field	xpt field	Transparent mode cannot be used	status field	ITASK and not the sending task
The message is sent to:													
The block number is assigned by:													
The block number assigned by TAF is:													
The message length is determined by:	numwords field	tlc field											
The character type is determined by:	Only 6-bit display code can be used	act field											
Format effector use is determined by:	Format effectors cannot be used	nfe field											
Auto input use is determined by:	Auto input cannot be used	aim field											
Transparent mode output is determined by:													
Status of the SEND is returned to:													

FORTRAN

CALL TSIM(code,mask,criterion,rlength,list, length)

code Integer whose value indicates which field is to be searched:

<u>Value</u>	<u>Field</u>
0	Data base name field.
1	User argument field.
2	Communication line field.
3	User name field.

mask Parameter whose value is taken as a binary mask. It is used in compiling the list of user names.

criterion Criterion value for the search.

rlength Integer variable whose value TAF sets to indicate the number of

list

entries found. This value is also returned as a FORTRAN function value.

Name of an array in which TAF places a list of found entries. If zero or absent, no list is returned, but rlength is given as the number of found entries.

length

Integer variable whose value indicates the number of words that list can hold. If zero or absent, no list is returned, but rlength is given as the number of found entries. Only the lowest order 12 bits of this value are used.

COMPASS

TSIM address

address

Address of a parameter table in the format shown in figure 2-14.

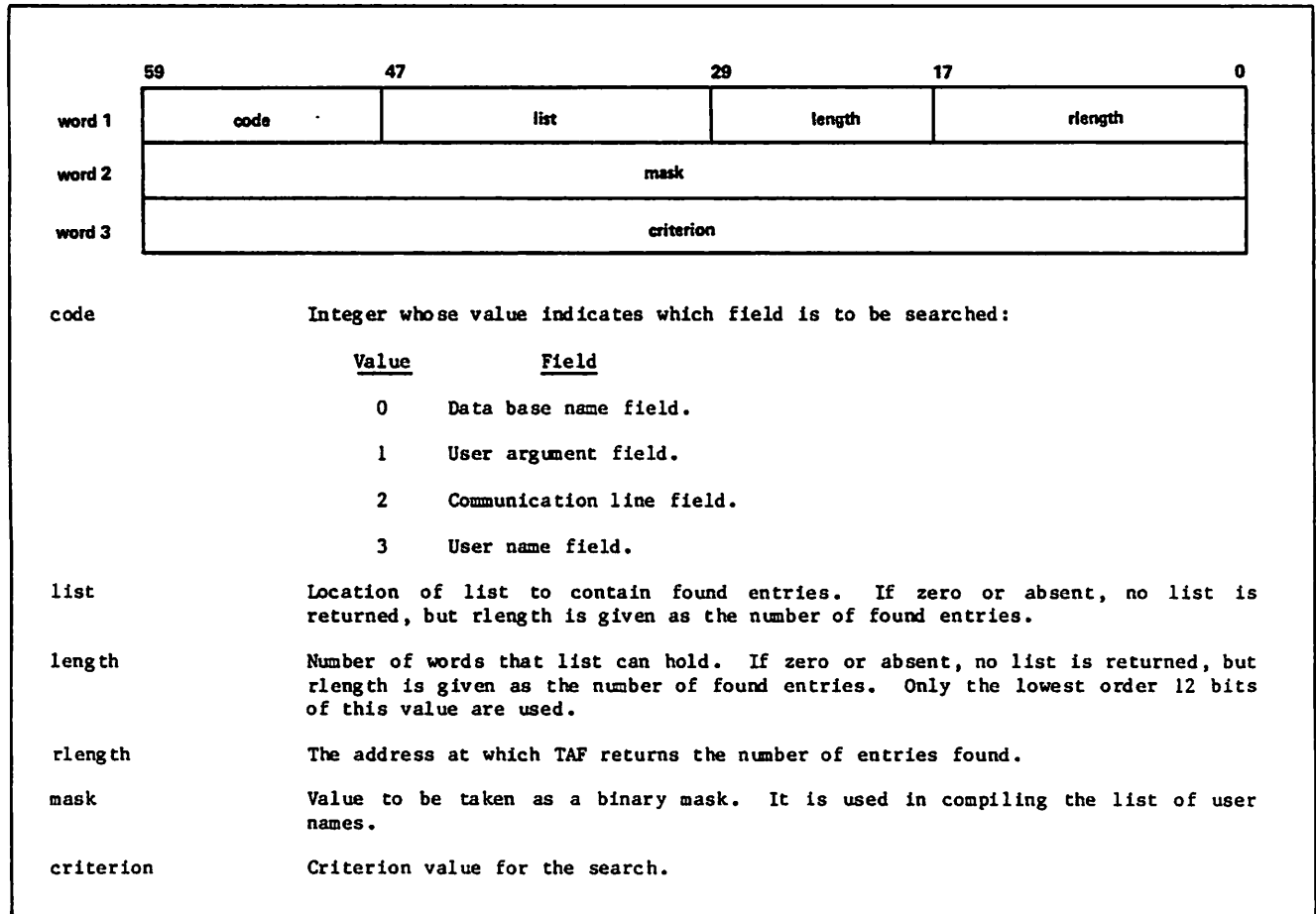


Figure 2-14. TSIM Parameter Table

TARO Request

The TARO request allows the task to manipulate the user argument field in the terminal status table entry of a particular user. (Refer to section 9 for further information.) The user argument field (24 bits) is operated on in the following manner. The logical product of the user argument field and the lower 24 bits of mask are taken; then the logical difference of this result and value is determined. This value is placed in return and replaces the old user argument field. This allows the user to test a specific field to obtain its value and to establish a particular value in the field. For example, bit 1 of the user argument field may be used to establish a terminal recovery condition. The TARO request operates in the following manner.

Case 1: The request interrogates the user argument field and determines whether the recovery bit (bit 1) is set. The value of this bit in the user area is not changed and is placed in return. (Only bits 0 through 3 are shown in this example.)

logical	010	user argument field of TST
product	<u>010</u>	mask parameter
logical	010	
difference	<u>000</u>	value parameter
	010	return value = new user argument field

Case 2: The request sets the recovery bit.

logical	000	user argument field
product	<u>010</u>	mask parameter
logical	000	
difference	<u>010</u>	value parameter
	010	return value = new user argument field

Case 3: The request clears the recovery bit.

logical	010	user argument field
product	<u>000</u>	mask parameter
logical	000	
difference	<u>000</u>	value parameter
	000	return value = new user argument field

The format is:

COBOL

ENTER TARO USING value mask username return.

value	Item whose value is a 24-bit value to be used to alter the user argument field.
mask	Parameter whose value is taken as a binary mask. Only the lowest 24 bits are used.
username	A 01-level item containing a one- to seven-character alphanumeric user name. The user name can also be a computational-1 item containing a binary zero (designating the originating user).
return	Item whose value TAF sets to indicate the resultant user argument bits.

FORTRAN

CALL TARO(value,mask,username,return)

value	Parameter whose 24-bit value is used to alter user arguments.
mask	Parameter whose value is taken as a binary mask. Only the lowest 24 bits are used.
username	Parameter whose value is a one- to seven-character alphanumeric user name, containing either blank or binary zero fill to a word boundary. If this parameter is omitted or is a binary zero, then the originating user is used.
return	Parameter in which TAF places the resultant user argument bits. This value is also optionally returned as the value of a FORTRAN function.

COMPASS

TARO address

address	Address of a parameter table in the format shown in figure 2-15.
---------	--

IIO Request

The IIO request allows a task to change its own I/O limit as measured in RA+1 calls from the task. Any request given in this manual or any data manager request is considered an RA+1 call. The default value is 320 calls.

The format of the request is:

COBOL

ENTER IIO USING nio.

nio Computational-1 item whose value specifies the new I/O limit.

FORTRAN

CALL IIO(nio)

nio Integer whose value specifies the new I/O limit.

COMPASS

IIO address

address Address of a parameter table in the format shown in figure 2-16.

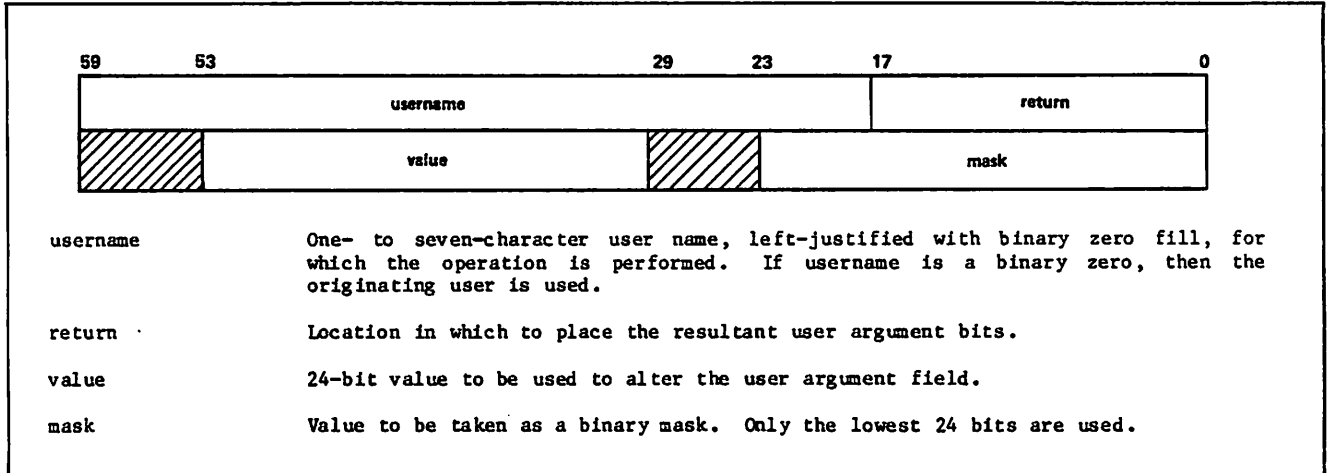


Figure 2-15. TARO Parameter Table

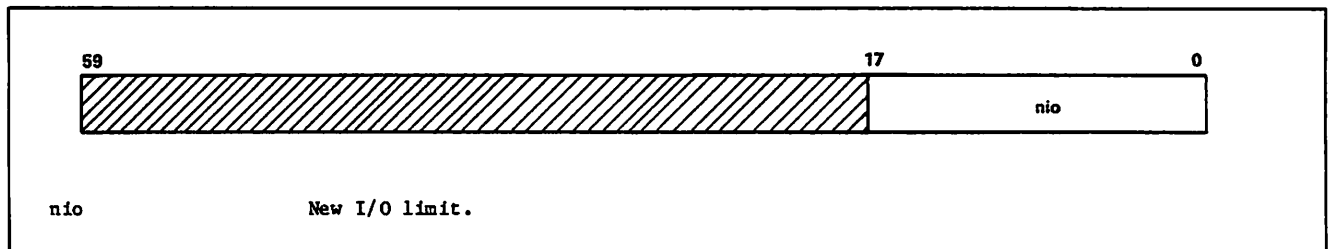


Figure 2-16. IIO Parameter Table

BWAITINP Request (COMPASS Only)

The BWAITINP request allows a task to move data from a terminal to a task field location. This location has the same format as the communication block (refer to figure 2-1). The request is processed the same as a WAITINP request, except that when the terminal data is moved to the specified location, it is moved to the area specified by the supplied parameter (address), not the first word address of the task (as in the WAITINP request). The first word address is not changed by the execution of this request.

The format is:

BWAITINP address

address Address of first word of the storage area where the terminal data is to be moved.

TAF does not pass this terminal data to tasks called with the CALLTSK and CALLRTN requests.

TPSTAT Request (COMPASS Only)

The TPSTAT request is retained for compatibility with previous releases of TAF. It was used to enable a task to determine under which of two terminal subsystems the task was running. Now, all the tasks run under NAM. Therefore, the request always returns a 1.

The format is:

TPSTAT address

address Address containing a 1 in bit 0 to indicate NAM is active.

BEGIN Request

The BEGIN request allows a task to specify the area where the terminal input data is to be placed, if an area other than the first word address of the task field length is desired. It supports programming languages that do not reserve the first word address of the task for the common block that is used for the communication block. The task must be installed in the appropriate task library with the (solicited) input directive, because the default is unsolicited. If task chains are involved, the calling task must issue a BEGIN request before resuming execution of the original task, because the calling task must be able to obtain the communication block containing the returned data from the called task. The returned communication block in solicited mode is in the same form as in unsolicited mode. Unless there is a specific need for this request, its use is not recommended because more execution overhead is required to support the secondary load request.

The format is:

COBOL

ENTER BEGIN USING buffer.

buffer A 01-level item identifying the user-defined buffer area.

FORTRAN

CALL BEGIN(buffer)

buffer An array name identifying the user-defined buffer area.

COMPASS

BEGIN address

address Address of the user-defined buffer area.

TASK CONTROL REQUESTS

These requests are used for task control.

- CEASE
End a task.
- ITL
Set a task time limit.
- LOGT
Log a terminal out of TAF.
- KPOINT
Initiate execution of K display commands.
- WAIT
Suspend processing for a given amount of time.

CEASE Request

The CEASE request provides task end processing and transaction completion processing. If a predetermined task list exists for the chain (that is, subsequent tasks have been requested for scheduling, or the CEASE is a return from a CALLRTN request), no special processing occurs on a CEASE request from a task. However, if the request is from the final task in the chain, the transaction executive releases all communication blocks and notifies the data manager that updated records should be replaced on the data base. The one exception to this procedure occurs if a task selects an abnormal CEASE by use of an optional parameter.

A task that terminates in this manner is automatically the final task in the chain. After an abnormal CEASE occurs, all communication blocks are released and records updated by the task but not yet updated on the data base are not replaced on the data base.

The format is:

COBOL

ENTER CEASE USING a.

a Computational-1 item whose value indicates the type of CEASE selected:

<u>a</u>	<u>Type of CEASE</u>
0	Normal CEASE.
1	Abnormal CEASE with-out memory dump.
-1	Normal CEASE and terminal remains locked.
217-1 > a > 1	Abnormal CEASE with memory dump.†
-1 > a > -(217-1)	Reserved for CDC.

Any other values produce undefined results and should not be used.

FORTRAN

CALL CEASE(a)

a Integer whose value indicates the type of CEASE selected:

<u>a</u>	<u>Type of CEASE</u>
0	Normal CEASE.
1	Abnormal CEASE with-out memory dump.
-1	Normal CEASE and terminal remains locked.
217-1 > a > 1	Abnormal CEASE with memory dump.†
-1 > a > -(217-1)	Reserved for CDC.

Any other values produce undefined results and should not be used.

COMPASS

CEASE a

a Integer whose value indicates the type of CEASE selected:

<u>a</u>	<u>Type of CEASE</u>
0	Normal CEASE.
1	Abnormal CEASE with-out memory dump.
-1	Normal CEASE and terminal remains locked.

†Dumps are identified by the transaction subsystem user name or by the appropriate DSDUMP parameter (refer to section 6).

a

Type of CEASE

217-1 > a > 1 Abnormal CEASE with memory dump.†

-1 > a > -(217-1) Reserved for CDC.

Any other values produce undefined results and should not be used.

ITL Request

The ITL request allows a task to set the time limit for the task to a new value. The time limit is expressed in exchange jump time units (that is, the number of times a task executed its full time slice without interruption). In addition, each call to this function decreases the CPU priority of the task by 1 until the lower limit of priority is reached to ensure good service to other tasks. The time limit default value is a product of two installation parameters. The released value is 1.12 CPU seconds. If this is exceeded, the task aborts and the message TIME LIMIT EXCEEDED is issued to the terminal.

The format is:

COBOL

ENTER ITL USING ntm.

ntm Computational-1 item whose value specifies the new time limit.

FORTRAN

CALL ITL(ntm)

ntm Integer whose value specifies the new time limit.

COMPASS

ITL address

address Address of a parameter table in the format shown in figure 2-17.

LOGT Request (COMPASS Only)

The LOGT request logs out a terminal from TAF and returns it to nontransaction mode. The terminal can then access other interactive applications. A system task is provided using this request. (Refer to System Tasks in section 11.)

The format is:

COMPASS

LOGT

There are no parameters.

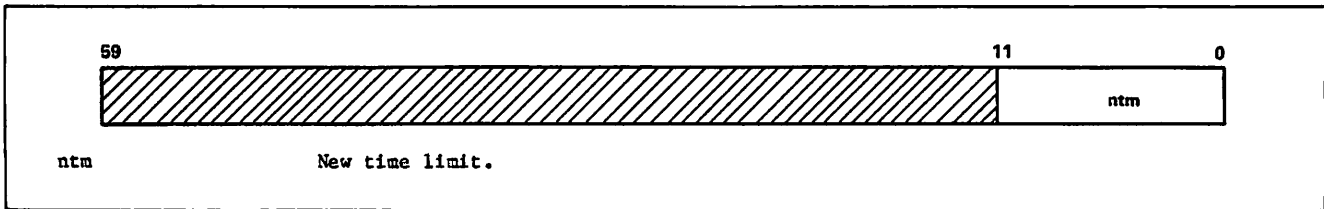


Figure 2-17. ITL Parameter Table

KPOINT Request

The KPOINT request allows a task to initiate execution of any K display command. Section 6 describes three K display commands: K.DUMP, K.DUMPLIM, and K.DSDUMP.

K display commands are normally issued from the system console to communicate with and to request specific actions of TAF. K display commands should be issued with care and then only when the results of the command are well-defined.

When a task issues a K display command, the prefix K. is omitted. For example, the entry from the console is K.DUMP., but the same command issued from a task is DUMP.

All K display commands, except K.DUMP, require that the calling task be on the system task library. Any task can issue the K.DUMP command.

The format is:

COBOL

ENTER KPOINT USING address.

address 01-level name of the character string (K display command). The length of the character string cannot exceed 80 characters. The string must begin on a word boundary and must be terminated by a period or a right parenthesis; the exception is the K.MESSAGE command, which must be terminated by 6 bits of zeros after either a period or a right parenthesis.

FORTRAN

CALL KPOINT(address)

address Address of the character string (K display command). The length of the character string cannot exceed 80 characters. The string must begin on a word boundary and must be terminated by a period or a right parenthesis; the exception is the K.MESSAGE command, which must be terminated by 6 bits of

zeros after either a period or a right parenthesis.

COMPASS

KPOINT address

address Address of the character string (K display command). The length of the character string cannot exceed 80 characters. The string must begin on a word boundary and must be terminated by a period or a right parenthesis; the exception is the K.MESSAGE command, which must be terminated by 6 bits of zeros after either a period or a right parenthesis.

WAIT Request

The WAIT request allows tasks to make TAF suspend processing for a specified number of seconds (0 to 3600). The transaction must have no outstanding data manager requests at the time a WAIT request is issued.

The format is:

COBOL

ENTER WAIT USING time.

time A computational-2 data item indicating the number of seconds for which processing is to be suspended. If unspecified or out of range, the task is aborted. Only the integer portion is significant.

FORTRAN

CALL WAIT(time)

time A real number or variable indicating the number of seconds for which processing is to be suspended. If unspecified or out of range, the task is aborted. Only the integer portion is significant.

COMPASS

WAIT address

address The address of a word specifying the number of seconds (N) for which processing is to be suspended. The word has the form 48/0,12/N. If unspecified or out of range, the task is aborted.

TASK UTILITY REQUESTS

The following task utility requests provide capabilities for the FORTRAN and COBOL user, some of which are difficult to do because they are at the bit level.

- LOGIN

Determine if a specified user is logged in.

- EXTRACT

Extract a bit string from a word.

- INSERT

Insert a bit string into a word.

1. The first part of the document discusses the importance of maintaining accurate records of all transactions.

2. It is essential to ensure that all data is entered correctly and that the system is regularly updated.

3. The second part of the document outlines the various methods used to collect and analyze data.

4. This includes the use of surveys, interviews, and focus groups.

5. The third part of the document describes the results of the data collection process.

6. The findings indicate that there is a significant correlation between the variables studied.

7. The final part of the document provides conclusions and recommendations for future research.



LOGIN Request

The LOGIN request allows a task to determine if a specified user is logged in before it issues a SEND request to that user. TAF aborts a task that issues a SEND request without recall to a user who is not logged in.

NOTE

It is possible for the receiving user to log out after the task issues a LOGIN request and before the task issues a SEND request. Therefore, it is recommended that the task issue the SEND request soon after issuing the LOGIN request.

The format is:

COBOL

ENTER LOGIN USING username status.

username A one- to seven-character alphanumeric user name, left-justified. A binary zero specifies the user associated with the transaction that is running.

status Address of the location to contain the returned status. The format of the returned status is computational-1. Values for status are:

- 0 User not logged in.
- 1 User logged in.

FORTRAN

CALL LOGIN(username,status)

username A one- to seven-character alphanumeric user name, left-justified. Possible forms are nHf and nLf (refer to the FORTRAN Extended 4 Reference Manual). A binary zero specifies the user associated with the transaction that is running.

status Address of the location to contain the returned status. The format of the returned status is integer. Values for status are:

- 0 User not logged in.
- 1 User logged in.

EXTRACT Request

The EXTRACT request allows a task to extract a bit string from a word and move it to another word, right-justified. For instance, the EXTRACT request can be used to check fields in the terminal status table or the communication block.

In the following request formats, if n is not within the specified range, if bbp plus 1 minus n is less than 0, or if there are any other errors in the parameters, TAF aborts the task and issues the following message.

TASK REQUEST ARGUMENT ERROR.

The format is:

COBOL

ENTER EXTRACT USING loc1 loc2 bbp n.

loc1 Data name of the source field. Any data type is allowed. loc1 must begin on a word boundary.

loc2 Data name of the computational-1 destination field. The bit string is right-justified in this field.

bbp Beginning bit position of field loc1; this is the leftmost bit of the field. Bits are numbered 59 through 0.

n Number of bits to extract; the range of values for n is 1 to 59, inclusive. The format of n must be computational-1.

FORTRAN

CALL EXTRACT(loc1,loc2,bbp,n)

loc1 Name of the integer source field.

loc2 Name of the integer destination field. The bit string is right-justified in this field.

bbp Beginning bit position of field loc1; this is the leftmost bit of the field. Bits are numbered 59 through 0.

n Number of bits to extract; the range of values for n is 1 to 59, inclusive. n must be integer type.

The COBOL STRING and UNSTRING statements also provide capabilities for manipulating character-aligned data. The EXTRACT request should be used only when the use of these COBOL statements is impossible or inconvenient.

INSERT Request

The INSERT request allows a task to insert a bit string into a word. For instance, the INSERT request can be used to set fields in the terminal status table.

In the following formats, if n is not within the specified range, if bbp plus 1 minus n is less than 0, or if there are any other errors in the parameters, TAF aborts the task and issues the following message.

TASK REQUEST ARGUMENT ERROR.

The format is:

COBOL

ENTER INSERT USING loc1 loc2 bbp n.

loc1 Data name of the computational-1 source field. The bit string must be right-justified in this field.

loc2 Data name of the destination field. Any data type is allowed. loc2 must begin on a word boundary.

bbp Beginning bit position of field loc2; this is the leftmost bit of the field. Bits are numbered 59 through 0.

n Number of bits to insert; the range of values for n is 1 to 59, inclusive. The format of n must be computational-1.

FORTRAN

CALL INSERT(loc1,loc2,bbp,n)

loc1 Name of the integer source field. The bit string must be right-justified in this field.

loc2 Name of the integer destination field.

bbp Beginning bit position of field loc2; this is the leftmost bit of the field. Bits are numbered 59 through 0.

n Number of bits to insert; the range of values for n is 1 to 59, inclusive. n must be integer type.

The COBOL STRING and UNSTRING statements also provide capabilities for manipulating character-aligned data. The INSERT request should be used only when the use of these COBOL statements is impossible or inconvenient.

APPLICATION RECOVERY REQUESTS

The following task requests are for use in recoverable transactions. All are for use by applications tasks although some are also used by system tasks.

- RERUN
Restarts a run unit using the initial terminal input.
- RGET
Retrieves data from the communication recovery file (CRF).
- RPUT
Places data on the communication recovery file.
- RSECURE
Retrieves a message previously saved on the communication recovery file by a SECURE request.
- SECURE
Signals the completion of a transaction's work and saves a message on the recovery file to be sent to the terminal when the run unit ceases.
- TSTAT
Returns the status of one or more specified objects.

Refer to section 7 for an overview of the recovery process from the applications programmer's perspective. Refer to the applicable data manager reference manual (CDCS, TAF/CRM, or TOTAL) for requests associated with data base recovery.

RERUN Request

The RERUN request causes the initial terminal input for a rerunnable run unit to be resubmitted to the transaction system. The input comes from the communication recovery file. The actions that occur as a result of RERUN are:

1. The recovery status is set to zero so that the recovery task will not be scheduled.
2. The restart status (TSTAT with keyword RESTART) is set to one.
3. A termination is processed for the run unit and ITASK is called to rerun the run unit.

Transactions using this request must take care to avoid entering an infinite loop if the RERUN results from an error condition.

The format is:

COBOL

ENTER RERUN USING status.

status A computational-1 data item whose value is set by TAF to indicate the status of the request. The values are:

- 0 No errors.
- 29 Recovery is not installed.
- 33 Run unit is not re-runnable.

FORTRAN

CALL RERUN (status)

status An integer variable in which TAF will return a status. The values are:

- 0 No errors.
- 29 Recovery is not installed.
- 33 Run unit is not re-runnable.

RPUT and RGET Requests

The RPUT request allows a task to put application-dependent data on the communication recovery file. The RGET request allows the task to retrieve the data placed on the recovery file by an RPUT. The length of the data in each request and the number of slots for data on the recovery file is specified at TAF initialization on the RECOVER statement in the TCF (refer to section 4). TAF maintains the data on a user name basis. TAF does not modify the user data unless the communication recovery file is initialized. These requests may be used by any task when TAF automatic recovery is installed.

The format is:

COBOL

ENTER "RPUT" USING message,index[,status,message-length,act].

ENTER "RGET" USING message,index,status,message-length[,act].

message An 01-level data item that contains the message. This is an input parameter for RPUT and an output parameter for RGET.

index A computational-1 data item indicating the index for the recovery record. Index 1 points to the first record in user area of the communication recovery file; index 2 points to the second record in user

area of the communication recovery file, and so on. This parameter is input for both RPUT and RGET. The maximum value of index is the value of the number parameter in the RECOVER statement in the TCF.

status A computational-1 data item whose value is set by TAF to indicate the status of the request. The values are:

- 0 No errors.
- 19 The index is outside of the user recovery file message area (RPUT only).
- 20 No message (RGET only).
- 21 The message does not fit in the recovery file message area (RPUT only).
- 22 The message does not fit in the user message area (RGET only).
- 23 Incorrect application character type (act).
- 28 The RGET act is not equal to the RPUT act.
- 29 TAF automatic recovery is not installed.

message-length A computational-1 data item indicating the length of the message in the message units specified by the act parameter. This parameter is required when using message units other than 6-bit display code. This is an input parameter for RPUT. For RGET, the user sets the message-length of the area to receive the message. Upon completion of RGET, TAF places the length of the message in message-length.

act A computational-1 data item indicating the application character type. The type is the same as for NAM. This is an input parameter. The values are:

- 2 7-bit ASCII characters, seven and one-half per word.
- 3 7-bit ASCII characters, right-justified in 12-bit bytes, five per word.
- 4 6-bit display code characters, 10 per word. This is the default value.

FORTTRAN

CALL RPUT(message,index,status,message-length [,act])

CALL RGET(message,index,status,message-length [,act])

message The name of an array that contains the message. This is an input parameter for RPUT and output parameter for RGET.

index An integer variable indicating the index for recovery record. Index 1 points to the first record in user area of the communication recovery file; index 2 points to the second record in user area of the communication recovery file, and so on. This parameter is input for both RPUT and RGET. The maximum value of index is the value of the number parameter in the RECOVER statement in the TCF.

status An integer variable whose value is set by TAF to indicate the status of the request. The values are:

- 0 No errors.
- 19 The index is outside of the user recovery area.
- 20 No message (RGET only).
- 21 The message does not fit in the recovery file message area (RPUT only).
- 22 The message does not fit in the user message area (RGET only).
- 23 Incorrect application character type (act).
- 28 The RGET act is not equal to the RPUT act.
- 29 TAF automatic recovery is not installed.

message-length An integer variable indicating the length of message in the message units specified by the act parameter. This is an input parameter for RPUT. For RGET, the user sets the message-length of the area to receive the message. Upon completion of the request, TAF returns the message-length as recorded on the communication recovery file.

act An integer variable indicating the application character type. The type is the same as for NAM. This is an input parameter. The values are:

- 2 7-bit ASCII characters, seven and one-half per word.
- 3 7-bit ASCII characters, right-justified in 12-bit bytes, five per word.
- 4 6-bit display code characters, 10 per word. This is the default value.

RSECURE Request

The RSECURE request allows a recovery task to obtain a message previously saved on the communication recovery file by a SECURE request. After obtaining the message, the task would normally send the message to the appropriate terminal. Any task may use this request if automatic recovery has been installed with TAF.

The format is:

COBOL

ENTER "RSECURE" USING message,status, message-length,act,format-effector, transparent-mode.

message An 01-level data item that contains the message. This is an output parameter.

status A computational-1 data item whose value is set by TAF to indicate the status of the request. The values are:

- 0 No errors.
- 20 No message.
- 22 The message does not fit in the user message area.
- 29 Recovery is not installed.

message-length A computational-1 data item indicating the length of message in the message units specified by the act parameter. Before issuing the request, the user sets the message-length area in the task. Upon completion of the request, TAF returns the message-length as recorded on the communication recovery file.

act A computational-1 data item indicating the application character type. The type is the same as for NAM. Before issuing the request, the user sets the act. Upon completion of the request, TAF returns the act as recorded on the recovery file. The values are:

- 2 7-bit ASCII characters, seven and one-half per word.
- 3 7-bit ASCII characters, right-justified in 12-bit bytes, five per word.
- 4 6-bit display code characters, 10 per word. This is the default value.

format-effector A computational-1 data item indicating format effector usage. The values are:

- 0 Format effectors appear in the message.
- 1 No format effectors appear in the message.

transparent-mode A computational-1 data item indicating transparent mode usage. Default is nontransparent mode. The values are:

- 0 Nontransparent data. Interactive virtual terminal transforms are used.
- 1 Transparent data. IVT transforms are inhibited.

FORTRAN

CALL RSECURE(message,status,message-length, act, format-effector,transparent-mode)

message The name of an array that contains the message. This is an output parameter.

status An integer variable whose value is set by TAF to indicate the status of the request. The values are:

- 0 No errors.
- 20 No message.
- 22 The message does not fit in the user message area.
- 29 Recovery is not installed.

message-length An integer variable indicating the length of the message in message units specified by the act parameter. Before issuing the request, the user sets the message-length area in the task. Upon completion of the request, TAF returns the message length as recorded on the communication recovery file.

act An integer variable indicating the application character type. The type is the same as NAM. Before issuing the request, the user sets the act. Upon completion of the request, TAF returns the act as recorded on the recovery file. The values are:

- 2 7-bit ASCII characters, seven and one-half per word.
- 3 7-bit ASCII characters, right-justified in 12-bit bytes, five per word.
- 4 6-bit display code characters, 10 per word. This is the default value.

format-effector An integer variable indicating format effector usage. The values are:

- 0 Format effectors appear in the message.
- 1 No format effectors appear in the message.

transparent-mode An integer variable indicating transparent mode usage. The values are:

- 0 Nontransparent data. Interactive virtual terminal transforms are used.
- 1 Transparent data. IVT transforms are inhibited.

SECURE Request

The SECURE request tells the transaction executive that the work of the run unit is complete. All that remains is to commit the data base and end the run unit. The SECURE request includes a terminal confirmation message. This message will be sent at the run unit end time (normal CEASE). If the commit request fails, the task may RERUN the run unit or issue a CEASE request with abnormal termination. In either case, the SECURE message will not be sent.

TAF records the message associated with SECURE on permanent storage so that the message can be delivered in spite of system or terminal failures. If no failures occur, the message will be sent from the task's SECURE message area. This means that the SECURE, data base commit, and CEASE must occur in the same task. TAF assumes that the message has been acknowledged by the terminal operator when the operator enters the next transaction input. The SECURE message may appear twice if recovery is done after a failure, but it is guaranteed to appear at least once. TAF also saves the NAM application block header with the message so that it will be sent with the correct character type and mode. Only the last SECURE message is saved per run unit. Any task may use this request if recovery has been installed with TAF; however, if the run unit is not recoverable, the SECURE message will not be issued to the terminal upon recovery (only at initial CEASE time).

For a BTRAN run unit the message is recorded on the communication recovery file but no message is sent to the terminal/username. The RECOVER statement maximumsize parameter specifies the message size that can be recorded on the communication recovery file.

The format of the message on the terminal depends on the terminal class and the format effectors in the message. Refer to the SEND request description earlier in this section for more information.

The format is:

COBOL

ENTER "SECURE" USING message,status[,message-length,act,format-effector,transparent-mode].

- message A 01-level data item that contains the message. This is an input parameter.
- status A computational-1 data item whose value is set by TAF to indicate the status of the request. The values are:
 - 0 No errors.
 - 21 The message is too large to fit in TAF's communication recovery file. Refer to the RECOVER statement maximumsize parameter.
 - 23 Incorrect application character type.
 - 24 Incorrect format-effector value.
 - 25 Incorrect transparent-mode value.
 - 26 The message exceeds the maximum length for SEND.
 - 27 The message exceeds the maximum length for a run unit.
 - 29 TAF automatic recovery is not installed.

message-length A computational-1 data item indicating the length of the message in the message units specified by the act parameter. This is an input parameter. It is required when the act parameter does not specify display.

act A computational-1 data item indicating the application character type. The type is the same as for NAM. This is an input parameter. The values are:

- 2 7-bit ASCII characters, seven and one-half per word.
- 3 7-bit ASCII characters, right-justified in 12-bit bytes, five per word.
- 4 6-bit display code characters, 10 per word. This is the default value.

format-effector A computational-1 data item indicating format effector usage. This is an optional parameter. It is necessary only if format effectors are used. The default is no format effectors. The values are:

- 0 Format effectors appear in the message.
- 1 No format effectors appear in the message.

transparent-mode A computational-1 data item indicating transparent mode usage. This is an optional parameter. It is necessary only if transparent mode is used. The default is nontransparent mode. The values are:

- 0 Nontransparent data. Interactive virtual terminal transforms are used.
- 1 Transparent data. IVT transforms are inhibited.

FORTRAN

CALL SECURE (message,status,message-length [,act,format-effector,transparent-mode])

message The name of an array that contains the message. This is an input parameter.

status An integer variable whose value is set by TAF to indicate the status of the request. The values are:

- 0 No errors.
- 21 The message is too large to fit in the communication recovery file. Refer to the RECOVER statement maximumsize parameter.
- 23 Incorrect application character type.
- 24 Incorrect format effector value.
- 25 Incorrect transparent-mode value.
- 26 The message exceeds the maximum length for SEND.
- 27 The message exceeds the maximum length for a run unit.
- 29 TAF automatic recovery is not installed.

message-length An integer variable indicating the length of the message in the message units specified by the act parameter. This is an input parameter.

act An integer variable indicating the application character type. The type is the same as for NAM. This is an input parameter. The values are:

- 2 7-bit ASCII characters, seven and one-half per word.
- 3 7-bit ASCII characters, right-justified 12-bit bytes, five per word.
- 4 6-bit display code characters, 10 per word. This is the default value.

format-effector An integer variable indicating format effector usage. This is an optional parameter. It is necessary only if format effectors are used. The default is no format effectors. The values are:

- 0 Format effectors appear in the message.
- 1 No format effectors appear in the message.

transparent-mode An integer variable indicating transparent mode usage. This is an optional parameter. It is necessary only if transparent mode is used. The default is nontransparent mode. The values are:

- 0 Nontransparent data. Interactive virtual terminal transforms are used.
- 1 Transparent data. IVT transforms are inhibited.

TSTAT Request

The TSTAT request returns information about the current user or run unit. It is used by both system tasks and applications (user) tasks.

The format is:

COBOL

ENTER "TSTAT" USING status,keyword₁,value₁,...,keyword_n,value_n.

status A computational-1 data item whose value is set by TAF to indicate the status of the request. The values are:

- 0 No errors.
- 12 The keyword specified requires TAF automatic recovery. TAF automatic recovery is not installed.
- 13 End of NEXT processing.
- 14 USER is not defined in the network file.
- 400+i keyword_i is invalid.

keyword_i An 01-level item that contains a one- to seven-character keyword for which a value is desired. The keyword must be left-justified and blank-filled. Refer to table 2-7 for keyword definitions.

value_i A data item to which TAF returns the value for the corresponding keyword. The keyword_i and value_i fields must appear in pairs. For USER, NEXT, OLDID, and NEWID usage is display. For all other TSTAT keywords, usage is computational-1.

FORTRAN

CALL TSTAT(status,keyword₁,value₁,...,keyword_n,value_n)

status An integer variable whose value is set by TAF to indicate the status of the request. The values are:

- 0 No errors.
- 12 The keyword specified requires TAF automatic recovery. TAF automatic recovery is not installed.
- 13 End of NEXT processing.
- 14 USER is not defined in the network file.
- 400+i keyword_i is invalid.

keyword_i A variable that contains a one-to seven-character left-justified keyword for which a value is desired. The keyword must be blank-filled. Refer to table 2-7 for keyword definitions.

value_i A variable field to which TAF returns the value of the desired keyword. The keyword_i and value_i must appear in pairs. For USER, NEXT, OLDID, and NEWID value_i is character data. For all other TSTAT keywords, value_i is an integer.

Some TSTAT keywords (both as COBOL and FORTRAN parameters) are meant for user tasks and others are meant for system tasks, such as BTASK, CTASK, and RTASK.

TSTAT User Task Keywords

- Keywords RESTART, OLDID, and NEWID are meant for use by application tasks that use multiple data manager begin-commit sequences to determine if the run unit is in a restarted state and, if so, where to begin processing.
- Keywords STEP, TRAN, OLDIN, NEWID, CRM, and CDCS are used by RTASK during terminal login/relogin to determine if a run unit requires rerun. The SEQ (sequence number) returned is the transaction sequence number that was first assigned to the run unit. RTASK can be either a system task or a user task.

TSTAT System Task Keywords

- Keywords NEXT, STEP, TRAN, OLDID, NEWID are used by BTASK to read sequentially the communication recovery file to find batch (BTRAN) run units that require rerunning. BTASK uses the SRERUN request with username to rerun BTRAN run units.

SYSTEM TASK RECOVERY REQUESTS

Applications programmers can skip this subsection since all the requests documented here are available only to system tasks (that is, tasks residing on the system task library). Recovery requests that are for use by both system tasks and application tasks are documented under Application Recovery Requests earlier in this section. Refer to section 8 for an overview of the recovery process from the data administrator's perspective. Refer to the applicable data manager reference manual (CDCS, TAF/CRM, or TOTAL) for requests associated with data base recovery. The following task requests are used by system tasks for transaction recovery.

- CALLTRN
Initiates a run unit.
- SRERUN
Resubmits initial terminal input for a rerunnable run unit.
- TINVOKE
Returns or specifies a system identifier for begin-commit sequences in this job.
- WSTAT
Records information needed for recovery processing.

CALLTRN Request

The CALLTRN request causes the initiation of a run unit. It is available only to the system task ITASK. The actions that occur as a result of CALLTRN are:

1. ITASK ceases. If an error condition is detected, an error message is issued to the terminal via MSABT.
2. The initial terminal input is logged on the CRF if the transaction is recoverable. The transaction will not be scheduled for execution until this circular input-output (CIO) operation is complete.
3. If the transaction initiated is recoverable, the commitment history of the previous recoverable run unit for this user is released.

Table 2-7. TSTAT Keyword Descriptions

Keyword	Value	Description
USER		Return status for the user/terminal specified in value. If USER or NEXT is not specified, then the request refers to the originating terminal/user. Only one of USER/NEXT can be specified in a single TSTAT request. Reserved for tasks on system task library.
NEXT		Return the next user from the communication recovery file(s). NEXT is available only to system tasks. When NEXT is specified, all other keywords will refer to the next user. Only one of USER/NEXT can be specified in a single TSTAT request.
SEQ		Return the transaction sequence number for specified user.
STEP		Run unit or job step.
	0	Nonrecoverable run unit or job input.
	1	CDCS down. This applies only to a recoverable run unit.
	2	Normal transaction CEASE or job end. Present only for BTRAN or TAF/CRM batch concurrency jobs.
	3	Task or job abort.
	4	Run unit is no longer rerunnable. This occurs when an I/O error is detected and the data cannot be recovered for the user.
	5	Recoverable run unit or job input.
	6	CRM begin-commit ids are logged.
	7	NAM down. This applies only to a recoverable run unit.
TRAN		Run unit type.
	1	Terminal user.
	2	BTRAN user jobs.
	3	TAF/CRM user job.
CRM		CRM usage.
	0	CRM may not be used by the transaction.
	1	CRM may be used by the transaction.
CDCS		CDCS usage.
	0	CDCS may not be used by the transaction.
	1	CDCS may be used by the transaction.
RESTART		Run unit is not in restarted state.
	1	Run unit is in restarted state. This state occurs when a RERUN request was issued.
OLDID		Return the old (previous) CRM begin-commit id from the communication recovery file.
NEWID		Return the new (current) CRM begin-commit id from the communication recovery file.

The format is:

COBOL

ENTER CALLTRN USING tname.

tname 01-level data item containing a one- to seven-character transaction name.

FORTTRAN

CALL CALLTRN(tname)

tname An integer containing a one- to seven-character transaction name, left-justified with blank fill.

SRERUN Request

The SRERUN request causes the initial terminal input for a rerunnable run unit to be resubmitted to the transaction subsystem. The input is retrieved from the communication recovery file. The actions that occur as result of SRERUN are:

- The recovery status is set to zero so that the recovery task will not be scheduled.
- The restart status (TSTAT with keyword RESTART) is set to one.
- The SRERUN is done without requesting run unit termination.

The format is:

COBOL

ENTER SRERUN USING status[,username].

status A computational-1 data item in which TAF will return a status. The values are:

- 0 No errors.
- 29 Recovery not installed.
- 31 username not in network file.
- 33 Run unit not rerunnable.

username 01-level data item containing a one- to seven-character user name. If this parameter is omitted, the rerun applies to the user name under which the current task is running.

FORTTRAN

CALL SRERUN(status[,username])

status An integer in which TAF will return a status. The values are:

- 0 No errors.
- 29 Recovery is not installed.

31 username is not in the network file.

33 Run unit is not rerunnable.

username A variable containing a one- to seven-character user name, left-justified and blank-filled. If this parameter is omitted, the rerun applies to the originating user.

TINVOKE Request

The TINVOKE request returns a system identifier to be used for all begin-commit sequences in the job or task. For TAF, this system identifier is the transaction sequence number. Optionally, for the recovery case, the job or task may supply an old system identifier. This establishes the task or job as a continuation of a previous task or job.

The format is:

COBOL

ENTER TINVOKE USING status,new-system-id,old-system-id.

status A computational-1 item in which TAF will return a status. The values are:

- 0 Request successful.
- 32 old-system-id unknown.

new-system-id A computational-1 item in which TAF will return a system identifier to be used for all begin-commit sequences in the job or task.

old-system-id A computational-1 item in which the task or job supplies the old system identifier that establishes the task or job as a continuation of the old one.

FORTTRAN

CALL TINVOKE(status,new-system-id,old-system-id)

status An integer variable in which TAF will return a status. The values are:

- 0 Request successful.
- 32 old-system-id unknown.

new-system-id An integer in which TAF will return a system identifier to be used for all begin-commit sequences in the task or job.

old-system-id An integer in which the task or job supplies the old system identifier that establishes the program as a continuation of a previous one.

WSTAT Request

The WSTAT request records information needed for recovery processing on the communication recovery file or TAF system tables. WSTAT is used by system task CTASK.

The format is:

COBOL

ENTER "WSTAT" USING status,keyword₁,value₁,...,keyword_n,value_n.

status A computational-1 item whose value is set by TAF to indicate the status of the request. The values are:

- 0 No errors.
- 29 Keyword specified requires TAF automatic recovery. TAF automatic recovery is not installed.
- 30 End of NEXT processing.
- 31 USER is not defined in the network file.
- 400+i keyword_i is invalid.

keyword_i An 01-level item containing a one- to seven-character keyword for which a value is specified. The keyword must be left-justified and blank-filled. Refer to table 2-8 for keyword definitions.

value_i A computational-1 item that contains the value for the desired keyword. The keyword_i and value_i fields must appear in pairs.

FORTRAN

CALL WSTAT(status,keyword₁,value₁,...,keyword_n,value_n)

status An integer whose value is set by TAF to give the status of the request. The values are:

- 0 No errors.
- 29 Keyword specified requires TAF automatic recovery. TAF automatic recovery is not installed.
- 30 End of NEXT processing.
- 31 USER is not defined in the network file.
- 400+i keyword_i is invalid.

keyword_i A variable containing a one- to seven-character keyword for which a value is specified. The keyword must be left-justified and blank-filled. Refer to table 2-8 for keyword definitions.

value_i Integer that contains the value of the preceding keyword. The keyword_i and value_i must appear in pairs.

The WSTAT keywords (both for COBOL and FORTRAN parameters) are given in table 2-8. All keywords necessary for recovery processing for a user name must be specified in the WSTAT request. For instance, CTASK makes a TSTAT request specifying NEXT, SEQ, STEP, TRAN, CRM, and CDCS followed by a WSTAT with SEQ, TRAN, CRM, CDCS, OLDID, NEWID, USER, and STEP. Any keyword not specified retains the value associated with it on the previous WSTAT request.

Table 2-8. WSTAT Keyword Descriptions

Keyword	Value	Description
USER		Set status for user name specified in value. If USER is not specified, then the request refers to the originating user name.
SEQ		The transaction sequence number specified in value is written for a given user name.
STEP		Transaction or job step.
	0	Nonrecoverable run unit or job input.
	1	CDCS down. Applies only to recoverable run units.
	2	Normal transaction cease or job end. Present only for BTRAN or TAF/CRM batch concurrency.
	3	Task or job abort.
	4	Run unit no longer rerunnable. This occurs when an I/O error is detected and the data cannot be recovered for the user.
	5	Recoverable run unit or job input.
	6	CRM begin-commit ids are logged.
	7	NAM down. Applies only to a recoverable run unit.
	8	Allow terminal or job input to TAF.
	9	Terminate TAF because of a recovery error.
TRAN		Transaction type.
	1	Terminal user.
	2	BTRAN user job.
	3	TAF/CRM user job.
CRM		CRM usage.
	0	CRM may not be used by transaction.
	1	CRM may be used by transaction.
CDCS		CDCS usage.
	0	CDCS may not be used by transaction.
	1	CDCS may be used by transaction.
OLDID		Set old (previous) CRM begin-commit id on the communication recovery file.
NEWID		Set new (current) CRM begin-commit id on the communication recovery file.

The CYBER Database Control System can be used alone or with one other data manager (CRM or TOTAL) in the same transaction. If a task is to use CDCS, the CDCS reference manuals listed in the preface should be consulted for information on how to set up the data base (schema, subschema, master directory, and files). This data base is associated with CDCS and the files will be at the control point of CDCS during execution. Tasks can use the CDCS verbs for data base access provided that they are compiled with the appropriate subschema relating them to a CDCS data base.

In the following, xx represents the two-character data base identifier.

A TAF application (known by the xx identifier in the TCF) is linked to a CDCS data base by means of the subschemas used in the tasks associated with the application (these are the tasks on xxTASKL). All tasks in a transaction must use the same subschema or a fatal task error (detected by CDCS) will occur.

A transaction will be considered to be connected to CDCS if it has successfully issued a request to CDCS and the terminate request to CDCS has not been completed. (A CDCS terminate request from other than the last task in a chain will be ignored.) If the last task in the chain does not issue a CDCS terminate request, one will be issued automatically when the last task completes.

A task will abort if it attempts to use CDCS when CDCS is not present in the system or if CDCS aborts while a transaction is connected to it. If a task aborts while it is connected to CDCS, TAF will issue an abnormal termination notice to CDCS for the task (unless the abort originated with CDCS).

Refer to sections 7 and 8 for information regarding recoverable transactions and CDCS/transaction aborts.

The following TAF files are associated with an application using CDCS:

TCF	}	Required.
xxJ		
xxTASKL	}	Required unless all tasks are on system task library.
xxJOR1		
xxJOR2	}	Optional.
xxJOR3		

The possibility of a deadlock involving CDCS and another data manager (TOTAL or CRM) exists. For instance, if task X locks some resources under one data manager and then tries to lock other resources under another data manager that task Y initially locked, then the two tasks may be deadlocked if task Y tries to lock any of the resources locked by task X. It is the responsibility of the installation, applications programmers, and the data administrator to prevent these situations from occurring. Such deadlocks can be prevented by locking resources in a specified order only, for example, lock in data manager A resources first, then data manager B resources, and so forth.

RESTRICT clauses cause tasks to automatically request additional central memory. All of the advice and requirements relevant to a task requesting that its FL be increased apply when RESTRICT clause processing is used (refer to Memory Requests, section 2).

CONFIDENTIAL

MEMORANDUM FOR THE DIRECTOR

Subject: [Illegible]

Reference is made to [Illegible]

It is noted that [Illegible]

On [Illegible]

The [Illegible]

It is recommended that [Illegible]

The [Illegible]

Very truly yours,

[Illegible Signature]

[Illegible Title]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]

[Illegible]



This section is intended for the data administrator and can be omitted by the applications programmer.

This section discusses four topics: the xxJ file, journal files, the TAF configuration file, and the communication recovery file.

An xxJ file is associated with each data base and has the following functions.

- Defines the user's user name, password, and optional user index.
- Defines optional journal files.
- Defines optional device residence for the task library and some files specific to the individual data managers (refer to the xxJ file descriptions that follow).

Up to three journal files can be specified optionally to supplement the system journalizing. A task can write to one of the journal files by using the JOURNAL request.

TCF, the TAF configuration file, defines the data bases associated with a data manager, defines the network files and communication recovery files to be used, and may contain K display commands.

A procedure file (TAFxxxx) is used to initialize TAF. Commands may be added to it for recovery or to establish the operating environment. For example, commands that attach journal files (xxJORN) and data base files can be included in this file. Refer to the NOS 2 Installation Handbook for more information (refer to the preface for the publication number).

xxJ FILE

An indirect access permanent file, xxJ, where xx is the data base name, is required for each data base. If additional journal files are needed, they are defined in the xxJ file. The xxJ file is saved under the transaction subsystem user name. The file is in job command format and varies depending on the data manager used.

NOTE

The transaction subsystem user name should be under the control of a data administrator. The applications programmer should not have access to this name.

CRM DATA MANAGER xxJ FILE

If data base recovery is to be used, the user specified on the USER statement must be able to read the xxJ file. This can be done by granting read permission to that user or by making the file public.

The structure of the xxJ file (when using the CRM Data Manager) is shown as follows.

xxJ	} Required.
USER(username,password, familyname)† BRF,number	
xxJOR1,devicetype,filetype. xxJOR2,devicetype,filetype.. xxJOR3,devicetype,filetype. xxTASKL,PN=packname, R=device.	} Optional.
CRM(xpfn1,type,mode,users, locks,mrl,pkl,hash,rec,fwi, packname,device) IXN(xpfxn,naky,packname, device) AKY(akn,ako,akl)	
AKY(akn,ako,akl) RMKDEF(xpfn1,rkw,rkp,kl,0, kf,ks,kg,kc,nl,ie,ch)	} One CRM statement for each file.
RMKDEF(xpfn1,rkw,rkp,kl,0, kf,ks,kg,kc,nl,ie,ch)	
•	
•	
CRM(xpfn _n ,type,mode,users, locks,mrl,pkl,hash,rec, fwi,packname,device) IXN(xpfxn,naky,packname, device) AKY(akn,ako,akl)	
AKY(akn,ako,akl) RMKDEF(xpfn _n ,rkw,rkp,kl, 0,kf,ks,kg,kc,nl,ie,ch)	
RMKDEF(xpfn _n ,rkw,rkp,kl, 0,kf,ks,kg,kc,nl,ie,ch)	
•	
•	
RMKDEF(xpfn _n ,rkw,rkp,kl, 0,kf,ks,kg,kc,nl,ie,ch)	

†This is the username, password, and familyname where all data base files and index files described in the following CRM and IXN statements are located.

The IXN and AKY statements are necessary only for multiple indexed files. The IXN statement describes the index file; the AKY statement describes an alternate key. There is one AKY statement per alternate key. The RMKDEF statement is necessary if the file has multiple indexes and is recoverable. The RMKDEF statement in the xxJ file must be identical to the RMKDEF statement used in the initial MIPGEN creation of the data base file. There must be one RMKDEF statement for each alternate key in the xxJ file. More information on RMKDEF can be found in the CRM AAM 2 Reference Manual.

The parameters for the xxJ file are:

- xx Data base name.
- password Batch password.
- number Number of before image recovery files for this data base.
- devicetype Device type:
 - MS Mass storage.
 - MT Seven-track tape.
 - NT Nine-track tape.
- filotype File type (refer to figure 4-1):
 - B Buffered write.
 - R Nonbuffered write.
- packname Pack name of the auxiliary device on which the user's task library or data base files reside. This parameter is optional on the CRM statement.
- device Type of device on which the user's task library or data base files reside. For example, DI (for an 844) or DI3 (for a multiunit 844 consisting of three devices). This parameter is optional on the CRM statement.
- xxpfn₁ Two- to seven-character file name. xx is the data base name. pfn₁ is the remaining zero to five characters in the data base file name.
- type CRM file type:
 - AK Actual key.
 - DA Direct access.
 - IS Indexed sequential.
- mode Attach mode of file xxpfn₁:
 - M Modify, append, or read.
 - R Read.
 - RM Read, with another user concurrently attaching the file in mode M.
 - W Write, modify, append, or read.

The mode parameter controls access to the file to ensure that either TAF or a batch job can write on or modify the file at a given time. However, TAF and a batch job can read the file simultaneously. Refer to the ATTACH command in the NOS 2 Reference Set, Volume 3, for access modes granted for different combinations of multiple access.

NOTE

If the TAF procedure file attaches a file that is described on a CRM statement, the mode specified in the TAF procedure file overrides the mode parameter in the xxJ file.

- users Number of transactions allowed to open file xxpfn₁ concurrently. When TAF terminates, the TAF dayfile contains a message that gives the number of OPEN requests not granted because the value of the users parameter was exceeded. Each user uses 38 words plus the pkl parameter size (in words).
- locks Number of records the transaction can lock for file xxpfn₁. When TAF terminates, the TAF dayfile contains a message that gives the number of locks not granted because the value of the locks parameter was exceeded. Each lock uses two words plus the pkl parameter size (in words).

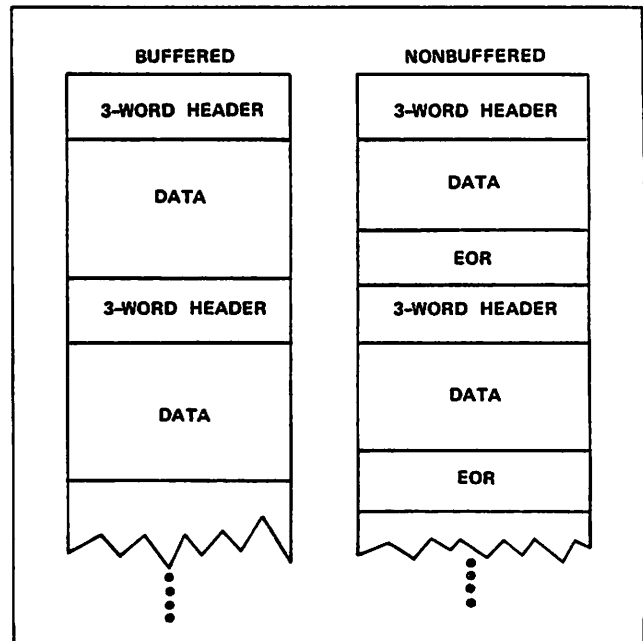


Figure 4-1. Journal File (Buffered and Nonbuffered Structure)

mrl Number specifying the maximum record length (in characters) for file `xpfn1`.
pk1 Number specifying the length in characters of the primary key for file `xpfn1`.†
hash One- to seven-character indirect access file that contains the binary code of the hashing routine for file `xpfn1`. This indirect access file must be stored under the username parameter on the USER statement. The entry point for the hashing routine must be this file name (hash). If this parameter is omitted, CRM uses the hashing routine supplied with CRM. This parameter applies only for CRM direct access files.
rec File recoverability.
 R This file is recoverable.
 N This file is not recoverable. When `rec` is not specified, `N` is assumed.

NOTE

Adding recoverable files to the data base that has a record or key length larger than any previously specified recoverable file causes recovery file validation errors during TAF initialization. Before image recovery files need to be recreated and after image recovery files need to be dumped to tape (via DMREC) and recreated. To ensure full data base recoverability, recoverable files should only be added to the data base following a normal termination of TAF.

fw1 Forced write indicator. When the forced write indicator is OFF, the modified data file buffers are written to disk when space is needed. When the forced write indicator is ON, all data file buffers that are modified in core are written to disk immediately. If `fw1` is not specified, the forced write indicator is set to OFF.
 Y Sets forced write ON for this file.
 N Sets forced write OFF for this file.
xpfxn Permanent file name of the index file; must begin with the two-letter application identifier `xx`.

naky Number of alternate keys in the index file.
akn Alternate key number. Alternate key description statements for one data base file must be ordered in a continuous sequence by alternate key number. For a detected key, this is the only parameter present in the statement.
ako Alternate key offset. The position in the record of the first character of the alternate key. This corresponds to the key offset parameter used on data manager requests accessing the record by this key. The first position in the record is character position number one (not zero).
akl Length of alternate key in characters.
rkw Relative word in the record in which the alternate key begins, counting from 0; required.
rkp Relative beginning character position within the relative key word (`rkw`), counting from 0; required.
kl Number of characters in the key, 1 to 255; required.
O Required to mark position for the reserved field.
kf Key format; required.
 0 or **S** Symbolic.
 1 or **I** Integer.
 2 or **U** Uncollated symbolic.
 3 or **P** Purge alternate key definition from the index.
ks Substructure for each primary key list in the index; optional.
 U Unique (default).
 I Indexed sequential; recommended for efficiency in processing.
 F First-in, first-out.
kg Length in characters of the repeating group in which the key resides.
kc Number of occurrences of the repeating group; zero for T-type records, and a number for a repeating group embedded within the record.

†The user can specify octal numbers and decimal numbers by using a postradix of B or D, respectively. Decimal is the default.

nl Null suppression; a null value is all spaces (symbolic key) or all zeros (integer key):

O Null values are indexed (default).

N Null values are not indexed.

ie Include/exclude sparse control character:

I Include alternate key value if the record contains a sparse control character.

E Exclude alternate key value if the record contains a sparse control character.

ch Characters that qualify as sparse control characters; up to 36 letters and digits can be specified as a character string.

n User index (obtained from site analyst).

devicetype Device type:

MS Mass storage.

MT Seven-track tape.

NT Nine-track tape.

filetype File type (refer to figure 4-1):

B Buffered write.

R Nonbuffered write.

packname Pack name of the auxiliary device on which the user's task library (xxTASKL) or data sets will reside.

device Type of device on which the user's file will reside [for example, DI (for 844) or DI3 (for a multiunit 844 consisting of three devices)].

aaaa
.
.
zzzz

} Four-character data set names.

TOTAL DATA MANAGER xxJ FILE

The following shows the structure of the xxJ file when using the TOTAL Data Manager.

```
xxJ
USER(username,password,familyname)†
UI=n.
xxJOR1,devicetype,filetype.
xxJOR2,devicetype,filetype.
xxJOR3,devicetype,filetype.
xxTASKL,PN=packname,R=device.
} Optional.

NL.
LG,PN=packname,R=device.
} Either the NL statement or the LG statement must be selected. PN and R parameters are optional. NL causes TOTAL to not log recovery data. LG causes TOTAL to log recovery data.

aaaa,PN=packname,R=device
.
.
.
zzzz,PN=packname,R=device
} PN and R parameters are optional.
```

The parameters for the xxJ file are:

xx Data base name.

password Batch password.

CDCS OR NO DATA MANAGER xxJ FILE

If an application uses CDCS and one of the other data managers (CRM or TOTAL), then the xxJ file should be structured as specified for the other data manager.

If an application accesses only the CDCS Data Manager or no data manager, the structure of the xxJ file is as follows:

```
xxJ
USER(username,password,familyname)†
xxJOR1,devicetype,filetype.
xxJOR2,devicetype,filetype.
xxJOR3,devicetype,filetype.
xxTASKL,PN=packname,R=device
} Optional.
```

The parameters are:

Parameter	Description
xx	Application (data base) name.
password	Batch password.
devicetype	Device type:
	MS Mass storage.
	MT Seven-track tape.
	NT Nine-track tape.

†TAF locates the data base using the familyname parameter in the procedure file that initializes TAF and the username and password parameters in the xxJ file. The username, password, and familyname parameters are used for validation.

filetype File type (refer to figure 4-1):
 B Buffered write.
 R Nonbuffered write.

packname Pack name of the auxiliary device on which the user's task library resides.

device Type of device on which the user's task library resides.

CREATING AN xxJ FILE

To enter an xxJ file, the data administrator can run a batch job to create and save the file under the transaction subsystem user name. This can also be done at a terminal through IAF. (Refer to the NOS 2 Reference Set, Volume 1, for IAF operating instructions.)

JOURNAL FILES

The two types of journal files are system and data base files.

SYSTEM JOURNAL FILE

The system journal file (a direct access file named JOURO that may be defined under the transaction subsystem user name or may be set up by the TAFxxxx procedure file at initialization) is not associated with any particular data base. All transaction data (input from terminals), errors, and transaction completion notification is automatically recorded on JOURO. The terminal status table is also journalized periodically. Data sent to terminals by tasks is not recorded on JOURO. JOURO will be created at initialization if not already present under the transaction facility user name.

DATA BASE JOURNAL FILES

The transaction subsystem allows up to three user-specified direct access journal files for each data base. The allowable names are xxJOR1, xxJOR2, and xxJOR3, where xx is the data base name. At initialization, TAF checks the xxJ files associated with all active applications to determine which xxJORN files are needed. It will then attach any specified xxJORN files not already at the control point and attempt to define any which it cannot find. The transaction subsystem user name must have write permission on these files. When a task makes a journal request (refer to JOURNAL request in this section), the data base for which the originating terminal is validated is used to determine the journal file on which to write. If the task is associated with a terminal validated for all data bases, the data base specified on the last data base request is used to determine the proper journal file.

JOURNAL FILE ENTRY HEADER

Each journal file entry (refer to JOURNAL request) has a header in the format shown in figure 4-2.

JOURNAL REQUEST

The JOURNAL request allows file entries to be made to supplement automatic journalizing.

The format is:

COBOL

ENTER JOURNAL USING jnum message.

jnum Computational-1 item whose value indicates the number of the journal file to which the entries are to be made. The value of this parameter must be right-justified and zero-filled.

<u>jnum</u>	<u>Journal File</u>
0	JOURO (the system journal file to which the transaction executive normally makes journal entries).
1	Data base journal file xxJOR1.
2	Data base journal file xxJOR2.
3	Data base journal file xxJOR3.

message Name of the 01-level data item containing the message to be journalized.

FORTRAN

CALL JOURNAL(jnum,message,length)

jnum Integer whose value indicates the number of the journal file to which the entries are to be made.

<u>jnum</u>	<u>Journal File</u>
0	JOURO (the system journal file to which the transaction executive normally makes journal entries).
1	Data base journal file xxJOR1.
2	Data base journal file xxJOR2.
3	Data base journal file xxJOR3.

message Name of the array containing the message to be journalized.

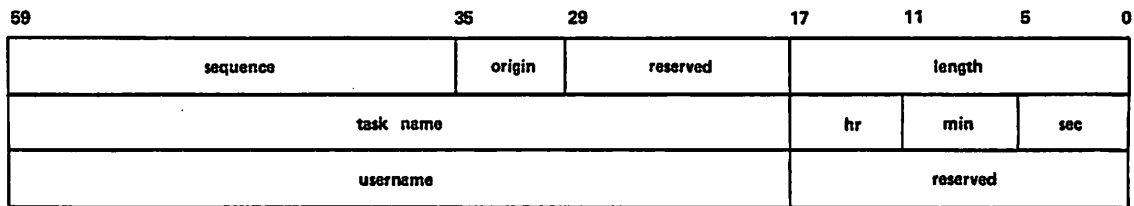
length Integer whose value specifies the length of the message in characters.

COMPASS

JOURNAL address

address Address of a parameter table in the form shown in figure 4-3.

If a journal file is defined on magnetic tape and an end-of-reel is detected, the transaction executive unloads the tape and checks for another tape on which to write. If the operator has not preassigned a unit, the journal file is placed on mass storage. If the operator enters the K.ASSIGN command, the transaction executive copies the mass storage journal file to the tape and places all subsequent entries on the tape. If the mass storage file does not fit on the tape, the message *UNABLE TO USE TAPE* is issued.



sequence Transaction sequence indicator.

origin Origin indicator (octal):

0 Task origin (JOURNAL request).

1 Transaction subsystem origin (input).

2 Data manager origin.

3 Transaction subsystem recovery/statistical data.

4 End-of-transaction indicator.

5 Incomplete block of terminal input data.

6 Terminal input for an interactive task.

7 Illegal intercontrol point transfer.

10 On-line LIBTASK update (TT option).

11 CDCS detected error (words following header contain error message).

12 MSG request with zero function code. Normal FORTRAN messages are logged with this function code.

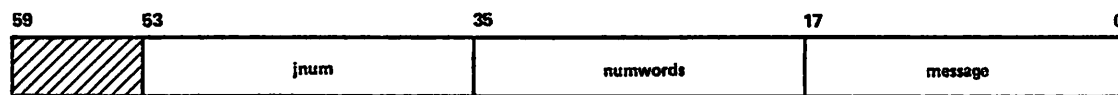
length Length of entry (equals header plus data block).

task name Name of task if task origin requested.

hr,min,sec Packed time (hour, minute, second).

username User name associated with transaction, if task origin.

Figure 4-2. Journal File Entry Header



jnum Journal file number, right-justified with binary zero fill:

0 JOURO (right-justified with binary zero fill: makes journal entries).

1 to 3 Data base journal file (xxJOR1, xxJOR2, or xxJOR3).

numwords Length of message in words.

message Address of message to be journalized.

Figure 4-3. JOURNAL Parameter Table

TAF CONFIGURATION FILE

The TAF configuration file is a required indirect access permanent file saved under the transaction facility user name. This file is used during TAF initialization and recovery. If it is not local when the TAFREC command is reached, TAFREC will attempt to get it from TAF's user index. The TCF has the following format:

<pre> USER,UN,PW,FM. DMS,datamanager,switch,xx₁, xx₂,...,xx_n. NETWORK,ID=i,FM=family, UN=username. </pre>	}	<p>At least one of each required.</p>
<pre> RECOVER,ID=i,MS=maximumsize, NM=number. </pre>	}	<p>Required or not allowed (refer to statement description).</p>
<pre> TABCON,n </pre>	}	<p>Required for batch concurrency. Refer to the TAF/CRM Data Manager Reference Manual.</p>
<pre> DISPLAY,status. K.command₁=value₁. . . . K.command_n=value_n </pre>	}	<p>Optional.</p>

DMS STATEMENT

DMS statements on the TCF are used to specify the data managers to be loaded by TAF and the applications (data bases) to be associated with each data manager.

The format is:

DMS,datamanager,switch,xx₁,xx₂,...,xx_n.

<p>data manager</p>	<p>Name of the data manager. Must be CRM, TOTAL, or OTHER. The same data manager can be specified on any number of DMS statements. Use the datamanager value OTHER for specifying an application that is either not associated with a data manager or is associated only with the CDCS Data Manager.</p>
---------------------	--

<p>switch</p>	<p>Data manager status. Must be ON or OFF.</p>
---------------	--

<p>ON</p>	<p>The specified data manager (TOTAL or CRM) will be loaded. The data bases specified on the same statement are to be made available for TAF processing. These data bases are referred to as active.</p>
-----------	--

<p>OFF</p>	<p>Data bases specified on the same statement are not made available for TAF processing.</p>
------------	--

<p>xx₁</p>	<p>Data base identifier. Must not be duplicated on the same or any other active DMS statement. For the CRM Data Manager, it is the two-letter data base name. For the TOTAL Data Manager, this parameter has the format:</p>
-----------------------	--

xyyyyy

which is the data base name (the DBMOD† file name) specified in the DATA-BASE-NAME† statement of the DBDL† statements. xx is the unique TAF data base name and yyyy is any combination of four alphanumeric characters.

A separate DMS statement for CDCS (OTHER) is not necessary if CDCS is to be used in conjunction with another data manager.

The TCF file must contain at least one DMS statement and each DMS statement with status set to ON must have at least one data base identifier or TAF will abort.

The maximum number of data bases that may be associated with one data manager is specified by the installation parameter MAXDB. This value is 25 unless it is altered by the user's site.

The maximum length of each DMS statement is 160 6-bit display code characters.

DISPLAY STATEMENT

The DISPLAY statement on the TCF controls whether or not the TAF K display will appear at the console during initialization and recovery. However, an error in a K display command causes TAFREC to issue an error message and bring up the K display for corrections regardless of the DISPLAY statement. The format is:

DISPLAY,status.

<p>status</p>	<p>A variable having one of the following values.</p>
---------------	---

<p>ON</p>	<p>K display will appear at the console during initialization and recovery to enable the operator to make changes at those times.</p>
-----------	---

<p>OFF</p>	<p>K display will not appear at the console during initialization and recovery.</p>
------------	---

† Refer to the TOTAL - CDC 2 Reference Manual for a detailed description.

NETWORK STATEMENT

The NETWORK statements on the TCF declare the network files (NCTFi) TAF uses to determine the users allowed to access TAF. There must be at least one and a maximum of eight NETWORK statements. The format is:

NETWORK, ID=i, FM=familyname, UN=username.

i An integer specifying a unique network file identifier. The range is zero to seven. This identifier is used in constructing the name of the network file (NCTFi).

familyname The family name under which the network file exists. If the familyname parameter is not specified, the default family name is used.

username The user name under which the network file exists. If the username parameter is not specified, the TAF user name is used.

RECOVER STATEMENT

The RECOVER statement on the TCF declares the set of files TAF is to use when in recovery mode. This directive is required if TAF is assembled with recovery, but it is not allowed if TAF is assembled without recovery.

The format is:

RECOVER, ID=i, MS=maximumsize, NM=number.

i An integer specifying a unique recovery file id. The range is zero to seven. This id is the id used in constructing the name of the communication recovery file (ZZCRFi). Only one id may be specified per RECOVER statement. The id must match an id specified in a NETWORK statement.

maximumsize An integer specifying the maximum size of the message, in words, that TAF will record on the communication recovery file. The default is 57. Maximum is installation parameter MAXWS. This is the maximum message size for the RPUT and SECURE requests.

number An integer specifying the maximum number of user messages that can be recorded on the communication recovery file via the RPUT request. The default is zero, the maximum is 10.

Each RECOVER statement specifies a recovery id. This id determines the communication recovery file that

TAF will use. The name of the corresponding CRF is ZZCRFi. The CRF resides under the same family and user name as the corresponding network file.

K. STATEMENT

Any K display commands that can be issued under the initial K display can appear in the TCF. Values specified in the TCF become the default values in place of the installation defaults (refer to the K display utilities in the NOS 2 Analysis Handbook for the installation default values). If the operator changes any of these values during initialization, those changes remain in effect if TAF is restarted in recovery mode.

If K display commands are used in the TCF, they must appear consecutively.

USER STATEMENT

The USER statement on the TCF specifies the user name under which TAF is to run. This user name should be defined through MODVAL to have the same user index as defined by the installation parameter TRUI. Only one USER statement is allowed in the TCF. The format is as follows:

USER, UN, PW, FM.
or
ACCOUNT, UN, PW, FM.

<u>Parameter</u>	<u>Description</u>
UN	The user name under which TAF is to run.
PW	The batch password for the user name.
FM	The family under which TAF is to run.

The USER statement should be the first statement in the TCF.

COMMUNICATION RECOVERY FILE

A communication recovery file is a file TAF uses to maintain recovery information about recoverable transactions. New CRFs are created and existing CRFs are reinitialized at initialization time by using the K.INT command (refer to section 8). There can be up to eight of these files (named CRF0, CRF1, ..., CRF7).

Each one corresponds to a network file having the same identifier (NCTF0, NCTF1, ..., NCTF7). For every user description statement in a network file, there is a user entry in the corresponding CRF. These entries are added and deleted as follows:

- New user entries are added to a CRF by adding a user description statement to the corresponding network file.
- User entries are removed from a CRF by removing the user description statement from the corresponding network file.

TRANSACTION FACILITY (TAF) K DISPLAYS

The following commands control operation of the TAF Subsystem. Initiate the TAF Subsystem by using the TAFffff command before issuing these commands.

INITIATING TAF K DISPLAY

When the transaction executive is brought to a control point, the message REQUEST *K* DISPLAY appears at the control point if a DISPLAY,ON command is specified in the TAF configuration file. Respond with the entry:

K,TAF.

Any of the following initialization commands can then be entered. If no values are to be changed, enter the command:

K.END.

Values are decimal unless otherwise indicated.

<u>Command</u>	<u>Description</u>
K.BFL=n.	Changes the starting and minimum field length allocated by TAF to CRM ($20000g \leq n \leq 100000g$). Default is 70000g.
K.CMB=n.	Changes the maximum number of communication blocks allowed to the TAF Subsystem ($7 \leq n \leq 40$). Default is 40.
K.ECS=n.	Sets the extended memory field length to be used by the transaction executive; n is octal thousands of words. Default is 0.
K.EFL=n.	Changes the maximum additional central memory field length made available to CRM for buffers and capsules ($0 \leq n \leq 100000g$). Default is 0.
K.END.	Ends input of the transaction executive initialization parameters. Initialization is completed when the TAF K display appears.

<u>Command</u>	<u>Description</u>						
K.ERO=CRF,op.	Specifies whether to override certain I/O and logic errors when processing the communication recovery files (CRF).						
	<table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left;"><u>op</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>Aborts if I/O or logic errors are encountered while processing the communication recovery files. This is the default setting.</td> </tr> <tr> <td>YES</td> <td> <p>I/O or logic errors encountered on a run unit header record result in the loss of that run unit with no indication to the terminal user. The loss is noted on the recovery report.</p> <p>I/O or logic errors encountered on a message record within a run unit result in a loss of that run unit. A status field is set in the run unit header, allowing TAF to inform the terminal user of the run unit loss.</p> <p>I/O or logic errors encountered on the CRF header record result in an unconditional abort of the TAF Subsystem.</p> </td> </tr> </tbody> </table>	<u>op</u>	<u>Description</u>	NO	Aborts if I/O or logic errors are encountered while processing the communication recovery files. This is the default setting.	YES	<p>I/O or logic errors encountered on a run unit header record result in the loss of that run unit with no indication to the terminal user. The loss is noted on the recovery report.</p> <p>I/O or logic errors encountered on a message record within a run unit result in a loss of that run unit. A status field is set in the run unit header, allowing TAF to inform the terminal user of the run unit loss.</p> <p>I/O or logic errors encountered on the CRF header record result in an unconditional abort of the TAF Subsystem.</p>
<u>op</u>	<u>Description</u>						
NO	Aborts if I/O or logic errors are encountered while processing the communication recovery files. This is the default setting.						
YES	<p>I/O or logic errors encountered on a run unit header record result in the loss of that run unit with no indication to the terminal user. The loss is noted on the recovery report.</p> <p>I/O or logic errors encountered on a message record within a run unit result in a loss of that run unit. A status field is set in the run unit header, allowing TAF to inform the terminal user of the run unit loss.</p> <p>I/O or logic errors encountered on the CRF header record result in an unconditional abort of the TAF Subsystem.</p>						

K.GO. Ends input of the transaction executive initialization parameters. Initialization is complete when the TAF K display appears.

K.INT=typ,fileid. Specifies which communication recovery files (CRF) are to be initialized. This is the only way to initialize a CRF. Files specified on a RECOVER command in the TAF configuration file and specified in this command are initialized. Files specified on a RECOVER command but not specified in this command are used for recovery.

This command also specifies whether CYBER record manager (CRM) data base recovery files are initialized or recovered. For CRM recovery files, this command is valid regardless of TAF assembly parameters.

<u>typ</u>	<u>Description</u>
CRF	Communication recovery files. This parameter is valid only if the installation parameter IPTAR equals 1.
CRM	CYBER Record Manager after-image and before-image recovery files.

<u>Command</u>	<u>Description</u>												
	<table border="1"> <thead> <tr> <th><u>fileid</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>n</td> <td>A digit from 1 to 7 that defines a CRF to be initialized. The digit corresponds to the ID parameter on the RECOVER command in the TAF configuration file. This parameter is not valid for CRM recovery files.</td> </tr> <tr> <td>ALL</td> <td>If typ is CRF, all communication recovery files defined by RECOVER commands in the TAF configuration file are initialized. If typ is CRM, all CRM recovery files are initialized.</td> </tr> <tr> <td colspan="2" style="text-align: center;">NOTE</td> </tr> <tr> <td colspan="2" style="text-align: center;">This parameter must be used with caution when typ is CRM since the CRM update history currently on the after-image recovery files is lost.</td> </tr> <tr> <td>NONE</td> <td>If typ is CRF, communication recovery files are initialized; all communication recovery files specified in RECOVER commands in the TAF configuration file are recovered. If typ is CRM, all CRM data bases are recovered based on information in the existing recovery files. No CRM recovery files are initialized. This fileid is the default for both typ=CRF and typ=CRM.</td> </tr> </tbody> </table>	<u>fileid</u>	<u>Description</u>	n	A digit from 1 to 7 that defines a CRF to be initialized. The digit corresponds to the ID parameter on the RECOVER command in the TAF configuration file. This parameter is not valid for CRM recovery files.	ALL	If typ is CRF, all communication recovery files defined by RECOVER commands in the TAF configuration file are initialized. If typ is CRM, all CRM recovery files are initialized.	NOTE		This parameter must be used with caution when typ is CRM since the CRM update history currently on the after-image recovery files is lost.		NONE	If typ is CRF, communication recovery files are initialized; all communication recovery files specified in RECOVER commands in the TAF configuration file are recovered. If typ is CRM, all CRM data bases are recovered based on information in the existing recovery files. No CRM recovery files are initialized. This fileid is the default for both typ=CRF and typ=CRM.
<u>fileid</u>	<u>Description</u>												
n	A digit from 1 to 7 that defines a CRF to be initialized. The digit corresponds to the ID parameter on the RECOVER command in the TAF configuration file. This parameter is not valid for CRM recovery files.												
ALL	If typ is CRF, all communication recovery files defined by RECOVER commands in the TAF configuration file are initialized. If typ is CRM, all CRM recovery files are initialized.												
NOTE													
This parameter must be used with caution when typ is CRM since the CRM update history currently on the after-image recovery files is lost.													
NONE	If typ is CRF, communication recovery files are initialized; all communication recovery files specified in RECOVER commands in the TAF configuration file are recovered. If typ is CRM, all CRM data bases are recovered based on information in the existing recovery files. No CRM recovery files are initialized. This fileid is the default for both typ=CRF and typ=CRM.												
K.MFL=n.	Sets the maximum field length to be used by the transaction executive ($40000 \leq n \leq 376600$). Default is 376600.												
K.REC=a.	Specifies whether to set the recovery bit in the user area of each terminal status table entry (YES or NO). If YES, the user recovery bit is set. If NO, the value of the user recovery bit is not changed from what it was before the command was issued. Default is NO.												
K.SCP=n.	Changes the number of subcontrol points ($2 \leq n \leq 31$). Default is 31.												
K.STOP.	Aborts the TAF Subsystem initialization unconditionally.												
K.TFL=n.	Changes the value used as the upper bound for TARGET. This is the amount of memory CRM uses for data and index blocks ($10000g \leq n \leq 100000g$). Default is 30000g. For more information, refer to the CYBER Record Manager Advanced Access Methods Version 2 Reference Manual.												
K.TLF=filename.	Changes the name of the system task library file (any valid file name). Default is TASKLIB.												

RESTARTING TAF K DISPLAY

On a level 3 recovery deadstart, or if the TAF Subsystem aborts, the TAF procedure file automatically restarts TAF by transferring control to the TAF automatic recovery program. The TAF automatic recovery program recovers the central memory pointers and variables defined during TAF initiation.

If a DISPLAY,ON command is in the TAF configuration file, the automatic recovery program brings up the K display on the left console screen. This display is identical to the initial K display, except that values specified in the TAF initiation replace any default values that were in the initial display.

TAF K DISPLAY

When the TAF Subsystem is executing, the TAF K display indicates the:

- Latest transaction sequence number.
- Number of words of unused memory.
- Maximum field length.
- Global task dump limit.
- Subsystem default values for memory dump parameters.

The TAF K display appears on the system console as shown in figure 8-31.

The subsystem default values are used to control memory dumps when parameters are not included in the CMDUMP or DSDUMP command. Any of these default values can be changed by specifying the corresponding parameter in the K.DSDUMP command.

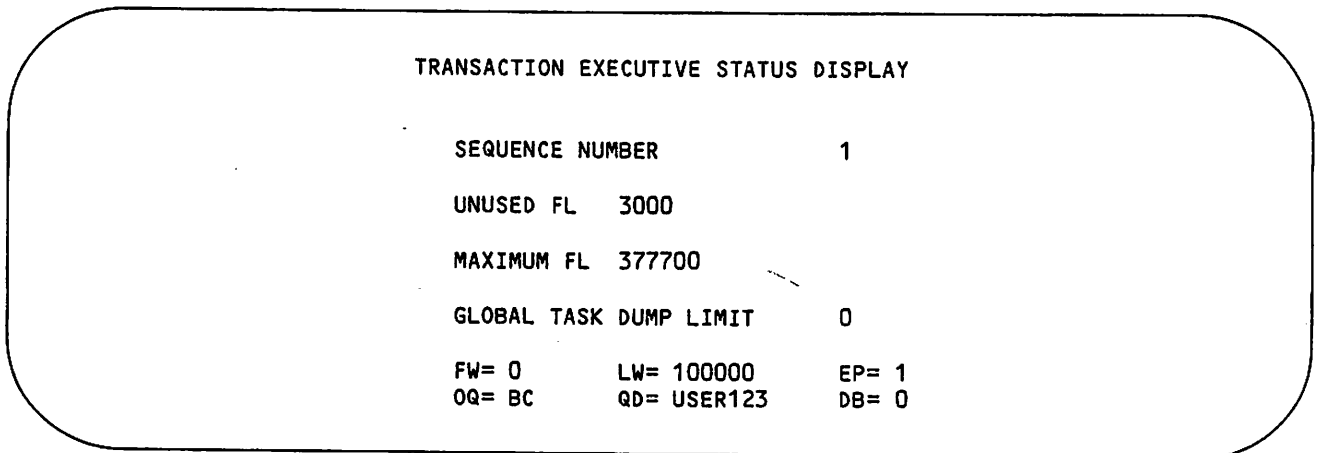


Figure 8-31. TAF K Display

The default parameter values for the CMDUMP and DSDUMP commands are given on the TAF K display shown in figure 8-31.

<u>Parameter</u>	<u>Description</u>
DB	This parameter is not used. It is retained for compatibility with previous releases of NOS.
EP	Exchange package: 0 Do not dump the exchange package. 1 Dump the exchange package.
FW	First word address of the task memory to be dumped.
LW	Last word address of the task memory to be dumped.
OQ	Output queue: BC Local batch. RB Remote batch. PF Permanent file.
QD	Queue destination: User name (if OQ=BC). Equipment identifier (if OQ=RB). Permanent file name (if OQ=PF).

TAF K-DISPLAY COMMANDS

When the transaction executive is at its control point, the following commands can be entered from the system console or submitted from tasks using the KPOINT request. Any task can issue the K.DUMP command. Only tasks that reside on the system task library can issue the other K-display commands. (Refer to the TAF Reference Manual for additional information on the KPOINT request and the system task library.)

<u>Command</u>	<u>Description</u>
K.ASSIGN,est. or K.ASSIGN,est,db,n.	Assigns a magnetic tape unit to be used for a journal file. est is the EST ordinal of the tape unit. The first form of the command makes unit est available for the transaction executive to assign to the next tape journal file that encounters end of reel. Two tape units may be preassigned. If a tape has not been preassigned in this manner, an end-of-reel on a journal file causes subsequent entries for that file to be placed on disk.

<u>Command</u>	<u>Description</u>
	The second form of the command forces journal file n (n=1, 2, or 3) for data base db, defined as a tape file, from disk to tape. The transaction executive copies the data from the disk journal file to tape est and places all subsequent entries for that file on the tape. This command is necessary after the transaction executive initialization to assign tape units to the tape journal files or after an end-of-reel on a tape journal file when no tape had been preassigned to the transaction executive. All data residing on the disk for the tape journal file must fit on the tape assigned by this command or the transaction executive will unload the tape and issue the message *UNABLE TO USE TAPE*.
K.DEBUG.	Turns on the application interface program (AIP) debug option, which logs all messages on trace file ZZZZZDN. Use this command only when TAF is installed with the DEBUG option.
K.DROP,n.	Drops an executing task at subcontrol point n.
K.DSDUMP,FW=addr, LW=addr,EP=pkg, OQ=outq,QD=qdest, DB=db.	Allows you to modify the standard system default parameters controlling memory dumps. The command does not directly cause a dump. Rather, it sets default values to be used when a subsequent CMDUMP request is received or when abort conditions occur. Refer to TAF K Display, earlier in this section, for explanations of the parameters.
K.DUMP,fwa,lwa.	Dumps all or part of the field length of the Transaction Facility from the first word address (fwa) to the last word address (lwa) of the area to be dumped. The default value for fwa is 0 and for lwa is 37777g. The default base is octal. If no parameters are specified, the entire field length is dumped. The output is routed to a printer that has an ID of 0.
	Unlike other K-display commands, the K.DUMP command can be issued from any task. Other K-display commands can be issued by tasks only if they are on the system task library (refer to the TAF Reference Manual).
	Since secure information may be contained in a dump of the Transaction Facility, the following safeguards have been set up to protect dumped information; however, the installation must take the ultimate responsibility for the protection of dumped information.
	<ul style="list-style-type: none"> • The global task dump limit (GTDL) can be set by the K.DUMPLIM command to limit the number of times the K.DUMP command can be issued from tasks. The initial value of the GTDL is 0 (zero), so the K.DUMP command is disabled from use by a task by default. • For all dumps of the Transaction Facility, whether you initiated it or a task did, a one-page header precedes the dump. This header page indicates the output is secure and should be given only to the TAF central site systems analyst. • When the Transaction Facility is dumped, the message TAF FIELD LENGTH DUMP RELEASED is issued to the system dayfile, the Transaction Facility dayfile, and line one of the control point.

<u>Command</u>	<u>Description</u>
K.DUMPLIM,n.	<p>Sets GTDL to value n ($0 \leq n \leq 9999999$). If n is not specified, the GTDL is set to 0.</p> <p>The GTDL is the number of times the K.DUMP command can be issued from tasks. This value is displayed on the TAF K display shown in figure 8-31. The initial value of the GTDL is 0. When the GTDL is 0, no dumps of the Transaction Facility can occur from tasks. Thus, the K.DUMP command is disabled from tasks by default.</p> <p>To enable the K.DUMP command for tasks, issue the K.DUMPLIM command to set the GTDL to a value greater than zero. Each time a task issues a K.DUMP command, the GTDL is decreased by one until it equals zero. When the first K.DUMP command is issued from a task with the GTDL equal to zero, the message</p> <p style="text-align: center;">GLOBAL TASK DUMP LIMIT EXHAUSTED</p> <p>is issued to the Transaction Facility dayfile, the system dayfile, and line one of the control point. Also, the message</p> <p style="text-align: center;">DUMPS LOST</p> <p>is displayed on the K display in place of the value of the GTDL. This message remains until the value of GTDL is set to a value greater than or equal to zero. The K.DUMPLIM command should be used with care in system tasks, since this might allow unauthorized users to alter the GTDL.</p>
K.IDLE.	<p>Idles down the transaction control point. Once idle down has been initiated, no new transactions will be permitted but currently executing transactions will be allowed to finish.</p>
K.JEND,db,n.	<p>Forces end-of-reel processing (writes an EOI and rewinds the file) on tape journal file n of data base db. If n is not a tape journal file, the command is ignored.</p>
K.MAXFL,n.	<p>Alters the transaction executive maximum field length. The transaction executive does not attempt to obtain more than n words of storage. This command is rejected if the value for n is more than 3766008 or less than the field length currently required for TAF.</p>
K.MESSAGE,TN=b. message.	<p>Directs the transaction executive to send message to a terminal specified by terminal name b.</p>
K.NODEBUG.	<p>Turns off the application interface program (AIP) debug option, which logs all messages on trace file ZZZZZDN. Use this command only when TAF is installed with the DEBUG option.</p>

<u>Command</u>	<u>Description</u>
K.OFFTASK,a,db.	Disables the use of task a, where a is the task name in the data base db task library directory (dbTASKL). The data base name db is not specified for tasks in the system task library.
K.ONTASK.a,db.	Reverses the effect of a previous OFFTASK command for the specified task a in the data base db task library directory (dbTASKL). The data base name db is not specified for tasks in the system task library (TASKLIB).
K.ROLLTIM,nnnnnn.	Changes the amount of time that TAF will retain its field length between communication input messages; nnnnnn is specified in units of milliseconds. Refer to the installation parameter ITRTL in the NOS 2 Installation Handbook.
K.SWITCH.	Causes the console K display to change to a display listing all allowable console commands. When K.SWITCH is entered a second time, the normal display returns. This command activates task KDIS and forces TAF to remain rolled in.
K.TBCON,nn.	Changes the number of TAF/CRM batch concurrency users; nn is less than or equal to the value specified on the TBCON command in TAF's configuration file (TCF).
K.TST,TN=a,DB=db, U=nnnn,UL=mmmm, NN=b.	Changes entries in the terminal status table for terminal a. The following entries can be changed: data base name db, user area upper 12 bits (nnnn), user area lower 12 bits (mmmm), and new terminal name b. The changes do not affect the network and simulation files. Do not use this command if the terminal is logged in.

TAF/CRM STATUS K DISPLAYS AND COMMANDS

You can use the K display to monitor the status of CRM, CRM data bases, or CRM data base permanent files while TAF/CRM is running if CRMTASK is present on the system task library. To get the CRM status K display enter:

K.DIS,CRMTASK.

After the K display is assigned to the task, the display in figure 8-32 is brought to the left screen.

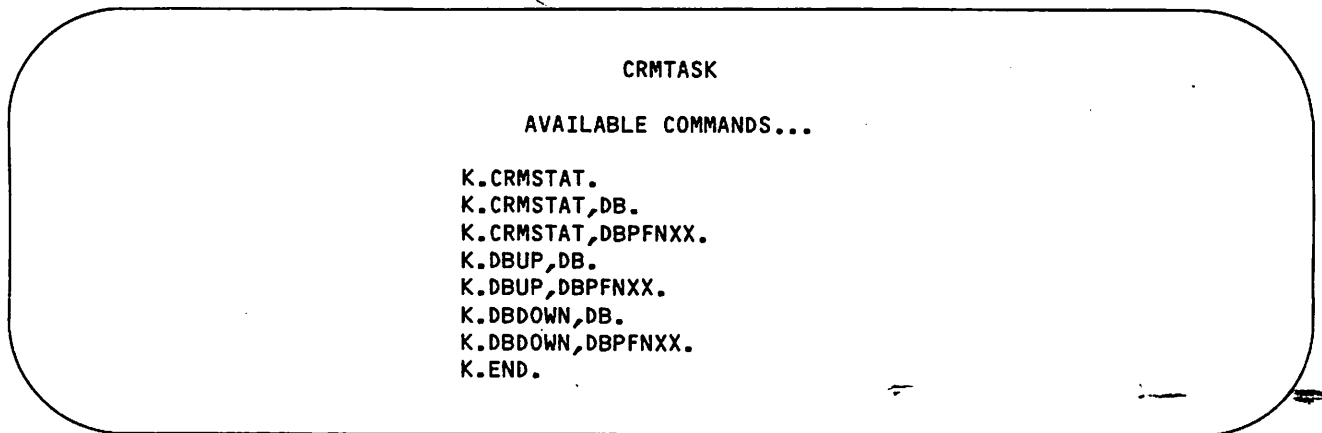


Figure 8-32. CRMTASK K Display

The following commands are available under CRMTASK.

<u>Command</u>	<u>Description</u>
K.CRMSTAT.	Selects the CRM status K display (see figure 8-33). This display shows the status of all CRM data bases.
K.CRMSTAT,db.	Selects the CRM data base status K display (see figure 8-34). This display shows the status of the specific CRM data base with identifier db. It also shows the file name and status of the before-image files.
K.CRMSTAT,dbfile.	Selects the CRM file status K display (see figure 8-35). This display shows the status of the specific permanent file in a CRM data base with file name dbfile.
K.DBDOWN,db.	Makes a specific CRM data base with identifier db unavailable for processing. When this command executes, the CRMTASK command directory appears on the K display.

<u>Command</u>	<u>Description</u>
K.DBDOWN,dbfile.	Makes a specific permanent file in a CRM data base with file name dbfile unavailable for processing. When this command executes, the CRMTASK command directory appears on the K display.
K.DBUP,db.	Makes a specific CRM data base with identifier db available for processing. When this command executes, the CRMTASK command directory appears on the K display.
K.DBUP,dbfile.	Makes a specific permanent file in a CRM data base with file name dbfile available for processing. When this command executes, the CRMTASK command directory appears on the K display.

```

** CRM STATUS **

nnn TRANSACTIONS IN INPUT QUEUE.
nn ACTIVE TRANSACTIONS.
nnn TRANSACTIONS IN OUTPUT QUEUE.

CRM DATA BASES

DB STATUS DB STATUS DB STATUS
AA UP BB IDEL CC DOWN
DD DOWN EE DOWN FF UP
GG UP

VALID COMMANDS ARE -

CRMSTAT. CRMSTAT,DB. CRMSTAT,DBPFN. END. +. -.
DBUP,DB. DBUP,DBPFN. DBDOWN,DB. DBDOWN,DBPFN.

```

Figure 8-33. CRM Status K Display

** CRM DATA BASE STATUS **

DATA BASE = db DATA BASE STATUS = UP
AFTER IMAGE FILE = arfname PRU-S REMAINING = nnn

BEFORE IMAGE FILES AND STATUSES

ZZAAB01-U ZZAAB02-U ZZAAB03-U ZZAAB04-D ZZAAB05-U
ZZAAB06-U ZZAAB07-D

FILE	STATUS	FILE	STATUS	FILE	STATUS
AAPFNO1	UP	AAPFNO2	DOWN	AAPFNO3	DOWN
AAPFNO4	IDLE	AAPFNO5	UP	AAPFNO6	UP
AAPFNO7	UP				

VALID COMMANDS ARE -

CRMSTAT. CRMSTAT,DB. CRMSTAT,DBPFN. END. +. -.
DBUP,DB. DBUP,DBPFN. DBDOWN,DB. DBDOWN,DBPFN.

Figure 8-34. CRM Data Base Status K Display

** CRM FILE STATUS **

FILE NAME = AAPFNO1
FILE STATUS = UP
RECOVERABLE = YES

PACK NAME = PACKNAM
DEVICE TYPE = DJ3
ATTACH MODE = RM

SIZE OF PRIMARY KEY = 80
NUMBER OF ALTERNATE KEYS = 3

ACTIVE USERS = 4
ACTIVE LOCKS = 8

VALID COMMANDS ARE -

CRMSTAT. CRMSTAT,DB. CRMSTAT,DBPFN. END.
DBUP,DB. DBUP,DBPFN. DBDOWN,DB. DBDOWN,DBPFN.

Figure 8-35. CRM File Status K Display

For the CRM status and CRM data base status displays, if all the information does not fit on one screen, you can bring up additional pages by entering the following command.

K.+.

To return to the first page of the display, enter the following command.

K.-.

At any time you can return to the initial CRMTASK K display (figure 8-32) by entering the following command.

K.MENU.

To end CRM status K-display processing, enter the following command.

K.END.

Each user entry in a CRF contains space for initial transaction input, RPUT messages, and a SECURE message. Two important CRF attributes are the number of messages, number, that a recoverable run unit may record on the file and the maximum size of these messages, maximumsize. The values for these parameters are given in the RECOVER statement in the TCF.

To reduce the number of user messages and/or the message size, the CRF must be reinitialized using the K.INT command (refer to section 8).

Under certain circumstances a CRF is reformatted during recovery. Refer to section 8 for details.

Faint, illegible text at the top left of the page.

Faint, illegible text at the top right of the page.



The transaction subsystem interfaces with the batch subsystem in three ways. First, batch jobs can initiate tasks by using the BTRAN request. Second, tasks can route jobs to the batch input queue by using the SUBMT request. Third, batch jobs can access TAF/CRM data files using the TAF/CRM batch concurrency feature.

The BTRAN and SUBMT requests are described in this section. The TAF/CRM batch concurrency feature is described in the TAF/CRM Data Manager 1 Reference Manual.

BTRAN REQUEST

The BTRAN request enables a batch job to start a transaction. The transaction is passed to the transaction executive through the system control point mechanism; therefore, the user must be validated for system control point communication (AW=CUCP). The user passes a transaction to the transaction executive via the header word shown in figure 5-1.

Multiple logins under one user name are not permitted in TAF. Thus, the user name for a job issuing a BTRAN request must not be in use within TAF by a transaction terminal or another batch job. Any SEND (or DISPLAY) requests in a BTRAN-initiated transaction must explicitly specify a user name different from the one under which the batch job is being run. Unlike a terminal-initiated transaction, BTRAN-initiated transactions are not required to issue any SEND request. A transaction can use the TSTAT request with the TRAN keyword to determine whether it was initiated with BTRAN.

The format is:

COBOL

ENTER BTRAN USING transaction status.

transaction	Name of the 01-level item containing the header word for the transaction to be submitted.
status	Name of the data that contains the returned status; the status is returned as a computational-1 value.

Decimal status values are shown in table 5-1.

FORTRAN

The user calls the BTRAN request in two ways. The first method is a FORTRAN function.

i=BTRAN(transaction)

i	Integer variable that contains the request status after BTRAN is called. The user must declare BTRAN as an integer.
---	---

transaction	Name of the array containing the header word for the transaction to be submitted.
-------------	---

The second method is a subroutine call.

CALL BTRAN(transaction,status)

transaction	Name of the array containing the header word for the transaction to be submitted.
-------------	---

status	Address of the variable that contains the returned status; the status is returned as an integer value.
--------	--

Decimal status values are shown in table 5-1.

COMPASS

BTRAN transaction

transaction	Address of the header word for the transaction to be submitted.
-------------	---

The status of the request is returned in register X6.

Decimal status values are shown in table 5-1.

Figure 5-2 demonstrates the use of BTRAN.

In the example shown in figure 5-2, LP3 is the name of the transaction to be processed by TAF. 50 is an item of data to be used by this transaction. STATUS=n goes to the OUTPUT file associated with the batch job. Output from the transaction cannot be sent to the user name under which the job is being run. Output to other user names (terminals) is permitted but not required.

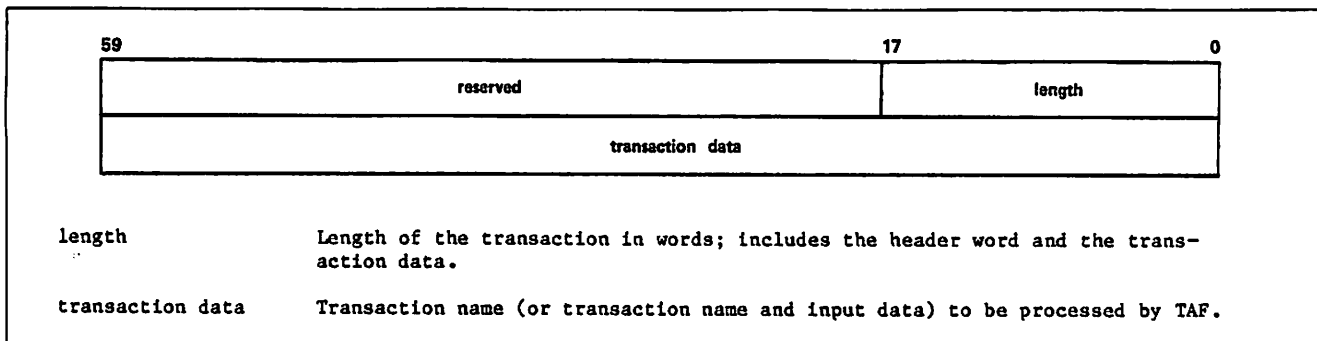


Figure 5-1. BTRAN Header Word

```

BTRANEX.
USER,XYZ1234,XYZ1.
CHARGE,*
COBOL5.
LIBLOAD (TRANLIB,BTRAN)
LGO.
--EOR--
IDENTIFICATION DIVISION.
PROGRAM-ID. BTRANEX.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. CYBER.
OBJECT-COMPUTER. CYBER.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 BATCH-TRANSACTION.
   02 RESERVED-AREA                PIC X(7).
   02 DATA-LENGTH                 VALUE 2   COMP-4   PIC 9999.
   02 TRANSACTION-DATA             VALUE IS "LP3,50" PIC X(6).
01 BTRAN-STATUS                   VALUE IS 0   COMP-1   PIC 9999.
PROCEDURE DIVISION.
START-RUN.
ENTER BTRAN USING BATCH-TRANSACTION, BTRAN-STATUS.
DISPLAY "STATUS=" BTRAN-STATUS.
STOP RUN.

```

Figure 5-2. BTRAN Request

Table 5-1. BTRAN Status Values

Value	Description	Value	Description
1	Transaction submission is complete.	15	The user name associated with the batch job issuing the BTRAN request is not validated in network file (job is aborted).
3	The transaction executive is not present.	17	The user name associated with the batch job issuing the BTRAN request is currently in use within TAF.
5	The transaction executive is busy (should not occur; inform site analyst).	19	A conflict in access types was detected. the previous transaction run under the specified user name was recoverable and not associated with a BTRAN-initiated transaction.
7	The submit block is too long; the maximum length is 62 words.	21	The transaction data is not within the field length of the batch job.
11	The transaction subsystem is not defined as a system control point (should not occur; inform site analyst).		
13	Access to TAF is denied because TAF is idling down.		

SUBMT REQUEST

The SUBMT request allows a task to route a job to the local batch input queue. The address parameter in the call points to the area in the task field length that contains the job to be routed in control word format. This format is described in conjunction with the READCW macro in the NOS 2 Reference Set, Volume 4 (refer to the preface for the publication number). Using this format, the job may consist of multiple records.

The format is:

COBOL

ENTER SUBMT USING address.

address Name of the 01-level item containing the job data to be routed. This data is in the format shown in figure 5-3.

FORTRAN

CALL SUBMT(address)

address Name of an array containing the job data to be routed. This data is in the format shown in figure 5-3.

COMPASS

SUBMT address

address Address of the first word of the job data to be routed in control word format as shown in figure 5-3.

The first word at address must be a control word in the format shown previously. The length field must specify the number of 12-bit bytes occupied by the data in the PRU (maximum of 320 bytes, equivalent to 640 characters). The level number in the trailing control word specifies whether the PRU is an end-of-record or end-of-file. If a logical record extends over more than one PRU, the level number for the first PRU of the record must be nonzero and other than 17g.

A leading and a trailing control word must be present for each PRU of data.

Typically, the data would consist of commands which would comprise one record. Additional data to be processed by the batch job could also be present and typically constitute subsequent records.

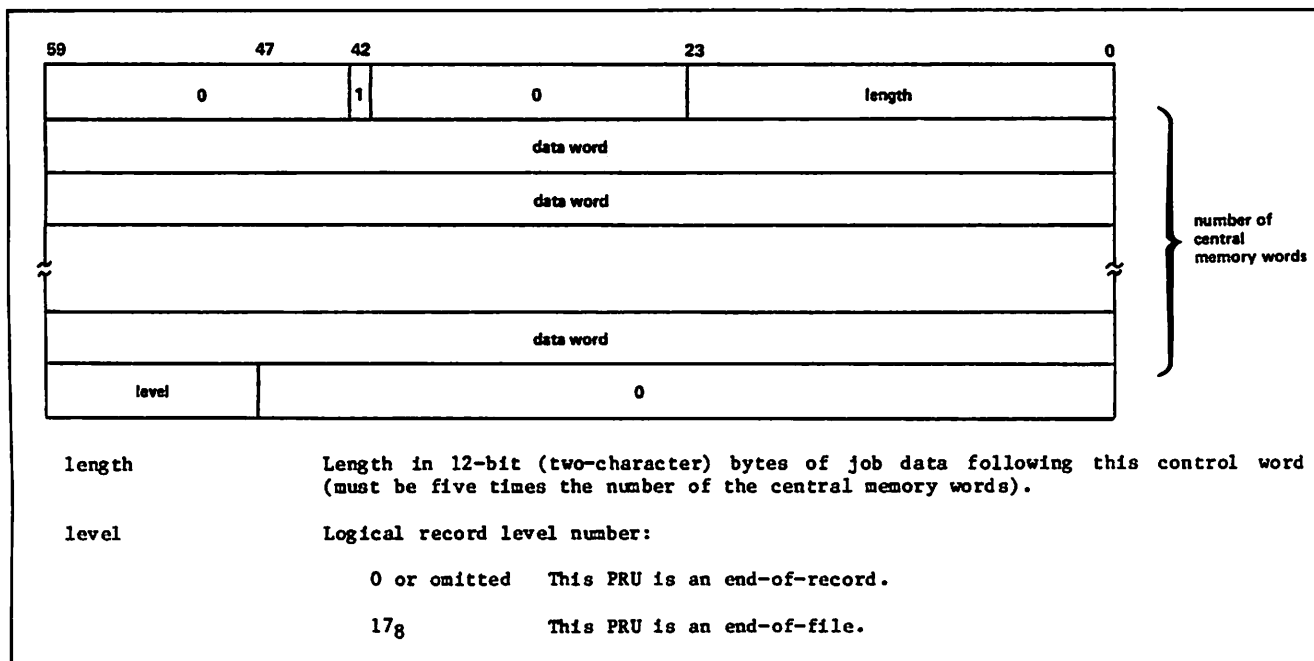


Figure 5-3. SUBMT Parameter Table

Figure 5-4 shows a task using the SUBMT request.

In the example shown in figure 5-4, the first word of JOB (00390 to 00410) sets bit 42 in the header word and gives the length (75) in 12-bit bytes of the card deck images to follow. This is five times the number of 10-character (60-bit) words in the job being submitted, in this case, statements 00420 to 00590.

Each job command must be from 1 to 80 characters long. All commands to be submitted must be padded to a multiple of 10 characters and the last 2 characters must be binary zero (character ":" in the 63/64-character set). Thus, if a statement ends in character position 9,10; 19,20; and so on, a zero word must be appended. Statements 00460 to 00530 give examples of this.

At job termination, the implicit output files of the SUBMT job are not queued by the system. This is because TAF routes the submitted job using the disposition code DC=NO. To obtain output from the job, do either of the following:

- Include ROUTE commands to place the desired files in the appropriate queues.
- Include the command SETJOB,DC=DF in the submitted job. This directs that implicit output is to be queued at the end of the job.

The example in figure 5-4 uses the first method by including a ROUTE command at line 00580.

```

00100 IDENTIFICATION DIVISION.
00110 PROGRAM-ID. SUBTEX1.
00120 ENVIRONMENT DIVISION.
00130 DATA DIVISION.
00140 COMMON-STORAGE SECTION.
      .
      .
Standard COBOL Communication Block (refer to section 2)
      .
      .
00370 WORKING-STORAGE SECTION.
00380 01 JOB.
00390 03 FILLER VALUE IS 1 COMP-4 PIC 9(5).
00400 03 FILLER VALUE IS 0 COMP-4 PIC 9(5).
00410 03 JOB-LENGTH VALUE IS 75 COMP-4 PIC 9(6).
*
* THE FOLLOWING IS THE JOB TO BE SUBMITTED
*
00420 03 JOB-CARD.
00430 05 JOB-ID VALUE IS "JOB." PIC X(4).
00440 05 FILLER VALUE IS ALL ":" PIC X(6).
00450 03 JOB-USER.
00460 05 USER-CARD VALUE IS "USER,DWG2603,ABC123." PIC X(20).
00470 05 FILLER VALUE IS ALL ":" PIC X(10).
00480 03 JOB-CHARGE.
00490 05 CHG-CARD VALUE IS "CHARGE,5927,693X451." PIC X(20).
00500 05 FILLER VALUE IS ALL ":" PIC X(10).
00510 03 JOB-DATA.
00520 05 COMMENT VALUE IS "* SUBMIT BATCH JOB." PIC X(19).
00530 05 FILLER VALUE IS ALL ":" PIC X(11).
00540 05 CTR-STATUS-1 VALUE IS "REWIND,INPUT." PIC X(13).
00550 05 FILLER VALUE IS ALL ":" PIC X(7).
00560 05 CTR-STATUS-2 VALUE IS "COPYSBF." PIC X(8).
00570 05 FILLER VALUE IS ALL ":" PIC X(2).
00580 05 CTR-STATUS-3 VALUE IS "ROUTE,OUTPUT." PIC X(13).
00590 05 FILLER VALUE IS ALL ":" PIC X(7).
*
* SUBMITTED JOB ENDS HERE
*
00600 03 INPUT-LEVEL VALUE IS 15 COMP-4 PIC 99.
00610 03 FILLER VALUE IS ALL ":" PIC X(8).
00620 PROCEDURE DIVISION.
00630 START-UP.
00640 ENTER SUBMT USING JOB.
00650 DISPLAY "JOB SUBMITTED.".
00660 STOP RUN.

```

Figure 5-4. SUBMT Request

A task can perform two types of memory dumps. The first type is a task dump, which is a dump of the task field length, exchange package, and associated data buffers. A task dump aids the applications programmer in debugging a task. The second type is a TAF dump, a dump of all or part of the TAF field length. A TAF dump contains pertinent information that is not contained in a task dump. This section describes the methods used to obtain both types of memory dumps.

TASK DUMP

The central memory dump (CMDUMP) request dumps central memory according to the parameters specified in the direct subsequent dumps (DSDUMP) request. Parameters specified on the DSDUMP request are used under three circumstances.

- When a fatal task error occurs.
- When a default value is requested in a CMDUMP request.
- When a parameter is in error in a CMDUMP request.

The purpose of DSDUMP and CMDUMP is to aid applications programmers in debugging their tasks. These dump requests allow users to dump any portion of their programs. The user specifies the first word address (fwa) and the last word address (lwa) of the area to be dumped. The exchange package may also be dumped by setting the proper parameter(s) to a nonzero value.

The DSDUMP request sets up default values for subsequent CMDUMP requests made by this task and all linked tasks in this transaction (those called by CALLTASK without CEASE). DSDUMP does not cause a dump; it only sets parameter values that can be defaulted as a result of a CMDUMP request.

A hierarchy of parameter values allows users to default any parameters they want on DSDUMP or CMDUMP calls. At system assembly time, default values are assembled for each parameter. These default values can be changed via a Kdisplay command. In addition, if users want to change any default parameter values for their tasks, they may do so by using DSDUMP requests.

This hierarchy of default values works in the following manner. The CMDUMP request causes a dump according to the parameters it specifies. If a default option is requested, the system determines if a DSDUMP request has been made for this transaction. If a DSDUMP request has been made and if the corresponding parameter value has been supplied, this value is used. If DSDUMP requests have not been specified, a new default value, the system default value for this parameter, is used.

All task dumps, whether due to an aborted task, an abnormal CEASE request, or a CMDUMP request, are handled in the same manner. A job file is created containing the information to be dumped in binary format, as well as a command record. The transaction executive issues a ROUTE request releasing the job file to the input queue for batch processing. A program called KTSAMP is used, either to generate listable output from the binary file, or to transfer it to a direct access permanent file.

When dumping to an output queue, a ROUTE command is included in the command record to direct the output produced by KTSAMP to the proper queue.

A USER command is always included, thus enabling KTSAMP to attach the appropriate file if the dump is destined for a user's file. In this case, KTSAMP appends the binary information to the end of the permanent file. This method of handling dumps relieves the transaction executive of time-consuming processing that is better performed by a utility.

DSDUMP REQUEST

The DSDUMP request allows the applications programmer to change any of the default values of the CMDUMP request. The DSDUMP does not cause a dump in itself, except when the system detects an error (that is, aborts). Any default values on the DSDUMP are satisfied from the general default values. All six parameters are required. If an error occurs on any DSDUMP parameter, the task is aborted.

The format is:

COBOL

ENTER DSDUMP USING fwa lwa ep db oq qd.

fwa Computational-1 item that specifies beginning address of task memory to be dumped; legal values are 0<fwa<FL (field length).

If fwa is negative, the default value is used.

lwa Computational-1 item that specifies last word of task memory to be dumped; it must be greater than fwa. If zero is specified, no dump of task memory occurs. If lwa equals FL, the dump extends to the end of the field length. If lwa is negative, the default value is used.

ep Computational-1 item that specifies exchange package option.

0 Do not dump exchange package.

> 0 Dump exchange package.

< 0 Use default value.

db Performs no function. Retained for compatibility with previous releases.

oq Computational-1 item that specifies output queue option.

< 0 Use default value (refer to following note).

0 Dump to local batch output queue.

1 Dump to remote batch output queue.

2 Dump to user permanent file defined under data base user number.

qd Queue destination option; value depends on oq selection.

oq qd

0 Printer id, specified as a computational-1 number. If negative, use default value (refer to following note).

1 User number

2 Direct access permanent file name

A valid name specified in display-coded, left-justified characters with either blank or binary zero fill.

NOTE

If either oq or qd is negative, the default value will be used for both, regardless of the option selected for the other.

FORTRAN

CALL DSDUMP(fwa,lwa,ep,db,oq,qd)

fwa Octal integer that specifies the beginning address of task memory to be dumped; legal values are zero<fwa<FL (field length).

If fwa is negative, the default value is used.

lwa Octal integer that specifies the last word of task memory to be dumped; must be greater than fwa.

If zero is specified, no dump of task memory occurs. If lwa equals FL, the dump extends to the end of the field length. If lwa is negative, the default value is used.

ep Integer that specifies the exchange package option.

0 Do not dump exchange package.

> 0 Dump exchange package.

< 0 Use default value.

db Performs no function. Retained for compatibility with previous releases.

oq Output queue option.

< 0 Use default value (refer to following note).

0 Dump to local batch output queue.

1 Dump to remote batch queue.

2 Dump to user permanent file defined under data base user number.

qd Queue destination option; value depends on oq selection.

oq qd

0 Integer specifying printer id. If negative, use default value (refer to following note).

1 User name

2 Direct access permanent file name

A valid name specified in display-coded, left-justified characters with either blank or binary zero fill.

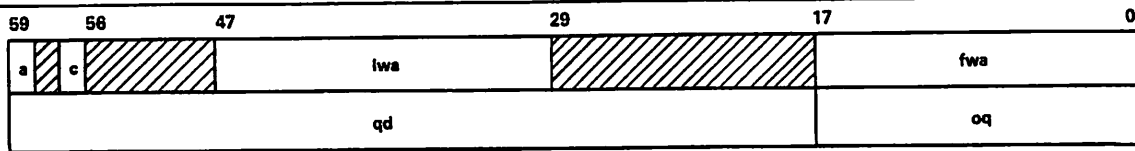
NOTE

If either oq or qd is negative, the default value will be used for both, regardless of the option selected for the other.

COMPASS

DSDUMP address

address Address of a parameter table. The table has the form shown in figure 6-1.



- a Set to one for exchange package dump.
- c Set to one for exchange package default value.
- lwa Last word of task memory to be dumped; must be greater than fwa. If zero is specified, no dump of task memory occurs. If lwa equals FL, the dump extends to the end of the field length. If lwa is negative, the default value is used.
- fwa Beginning address of task memory to be dumped; legal values are zero through lwa-1.
If fwa is negative, the default value is used.†
- qd Queue destination option; the value depends on the oq selection as shown following the oq description.
- oq Output queue option.
 - 0 Dump to local batch output queue.
 - 1 Dump to remote batch output queue.
 - 2 Dump to user permanent file defined under data base user name.

Interdependency between oq and qd is defined as follows:

<u>oq</u>	<u>qd</u>
0	Printer id, specified as binary number, right-justified within word; if negative, use default value.
1	User name.
2	Direct access permanent file name.

} A valid name specified in display-coded, left-justified characters with either blank or binary zero fill.

† Default values are specified by setting the sign bit.

Figure 6-1. DSDUMP Parameter Table

CMDUMP REQUEST

The CMDUMP request allows the applications programmer to dump memory in the task's field length to a selected output queue (and logical device). The first six parameters are required. If there is an error on or a default of any parameter specified by CMDUMP, the parameter value of the DSDUMP call is used.

The format is:

COBOL

ENTER CMDUMP USING fwa lwa ep db oq qd.

The descriptions of the first six parameters are the same as those for the DSDUMP request.

FORTRAN

CALL CMDUMP(fwa,lwa,ep,db,oq,qd)

The descriptions of the first six parameters are the same as those for the DSDUMP request.

COMPASS

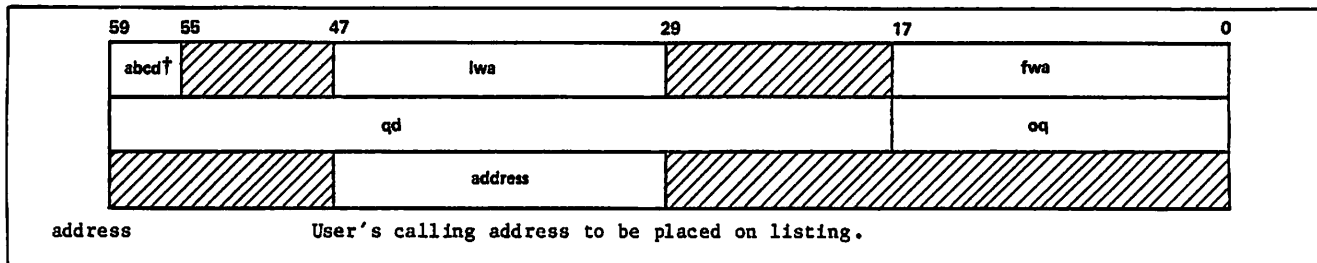
CMDUMP address

address Address of a parameter table in the form shown in figure 6-2.

NOTE

The manner in which the hierarchy of default values works was previously described. However, in the special case of output queue and queue destination (oq and qd parameters), if either parameter is defaulted, the next pair of output queue and queue destination parameters is obtained from the next level of the default hierarchy.

There is also a special case concerning errors in a CMDUMP request. If the parameter in error is the output queue value, the system is unable to route the dump as the user wants. If this error is on a CMDUMP request, the system attempts to dump according to the corresponding DSDUMP output queue and queue destination parameters. If that is not possible, the general system values are used.



†The descriptions of the first five parameters are the same as those shown in figure 6-1.

Figure 6-2. CMDUMP Parameter Table

K.DSDUMP COMMAND

The K.DSDUMP command can be used under the K display for the transaction executive control point to reset default dump parameters. The parameters can be given in any order and only those parameters to be changed need to be given.

The format when entered from the console is:

K.DSDUMP,FW=fwa,LW=lwa,EP=ep,OQ=oq,QD=qd.

The values of these parameters are as specified for the DSDUMP request made by a task. These system dump parameters can be examined by checking the K display for control points.

KTSDUMP UTILITY

KTSDMP can be called via a command to form listable memory dumps. The input files can be created by the transaction executive or by any other program. The following items can be listed.

- Central memory of a task or program.
- Exchange package and control point area of a task.
- Communication block of a task (always present).

The format is:

KTSDMP (lfn₁,lfn₂,P,O)

- | | |
|------------------|---|
| lfn ₁ | Input file name; if absent, default is INPUT. |
| lfn ₂ | Output file name; if absent, default is OUTPUT. |
| P | If P is specified, append file lfn ₁ on direct access permanent file lfn ₂ . The job must be running with the proper user name and file lfn ₂ must be defined. End-of-file is not copied to lfn ₂ . |
| O | If O is specified, the dump is in octal format. If O is omitted, the dump is in octal and 6-bit display code format, with the 6-bit display code format to the right of the octal format. |

If the P option is used, an installation may keep a running stack of memory dumps to be listed selectively at some later time. To list all dumps on the permanent dump file, the file should be attached and KTSDMP executed. All dump records on the file are processed. To list only selected dumps, the permanent file should be attached and the selected dump records extracted using the GTR command (refer to the MOS 2 Reference Set, Volume 3). KTSDMP should be executed to process the output file from the GTR command.

The user must specify the O option when issuing the KTSDMP command from a terminal; if the O option is not specified, execution is terminated, and KTSDMP issues the following message.

DISPLAY DUMP NOT ALLOWED TO TERMINAL.

TASK DUMP FORMATS

Figure 6-3 contains the formats of the dumps provided. The leftmost date and time in the header are the date and time of creation of the KTSDMP input file. The current date and time follow. The task control point area dump appears only when the exchange package dump is requested. The exchange package dump, task control point area dump, and the communication block dump all appear on the same page when the exchange package dump is requested.

TAF DUMP

A TAF dump is a dump of all or part of the TAF field length. The user initiates a TAF dump using the K.DUMP command. A TAF dump assists the systems analyst in detecting system errors that do not abort TAF. When a task detects an error in TAF, a TAF dump might be needed because there is significant information outside the task field length that is not contained in a task dump.

A task initiates a TAF dump using the KPOINT request (refer to KPOINT Request in section 2) in conjunction with the K.DUMP command. The K.DUMPLIM command sets a global limit on the number of K.DUMP commands that tasks can issue. A task uses the KPOINT request in conjunction with the K.DUMPLIM command to set the global task dump limit (GTDL). The remainder of this section describes the K.DUMP and K.DUMPLIM commands.

KTSDMP-taskname	yy///mm/dd.hh.mm.ss.	yy/mm/dd.hh.mm.ss.
EXCHANGE PACKAGE.		
Exchange package dump		
TASK CONTROL POINT AREA.		
Task control point area dump		
COMMUNICATION BLOCK DUMP FROM fwa TO lwa		
PROGRAM taskname SEQUENCE number ADDRESS xxxx		
Communication block dump		
KTSDMP-taskname	yy/mm/dd.hh.mm.ss.	yy/mm/dd.hh.mm.ss.
DUMP FROM fwa TO lwa		
Memory dump		
KTSDMP-taskname	yy/mm/dd.hh.mm.ss.	yy/mm/dd.hh.mm.ss.
DUMP FROM fwa TO lwa		
Memory dump		
KTSDMP-taskname	yy/mm/dd.hh.mm.ss.	yy/mm/dd.hh.mm.ss.

Figure 6-3. KTSDMP Format

K.DUMP COMMAND

The K.DUMP command initiates a dump of all or part of TAF. The user can specify the first word address (fwa) and the last word address (lwa) of the memory to be dumped.

The command formats when issued from a task are:

DUMP,fwa,lwa.

or

DUMP,lwa.

or

DUMP.

fwa Octal address of the first word of memory to be dumped; legal values are 0<fwa<lwa. The default is zero. This parameter cannot be used unless the lwa parameter is also used.

lwa Octal address of the last word of memory to be dumped; legal values are fwa<lwa<377777. The default is 377777.

If both parameters are omitted, the entire field length is dumped. The output of the dump is routed

to a printer that has an identifier of zero. A header page precedes the dump; it lists the date and time of the dump and the name of the task that issued the dump. Also listed are the instructions that the dump be given to the TAF central site systems analyst.

K.DUMPLIM COMMAND

The K.DUMPLIM command sets the GTDL. The GTDL specifies the number of K.DUMP commands that tasks can issue.

The command format when issued from a task is:

DUMPLIM,n.

n The decimal number of TAF dumps that tasks can issue. The default is zero. The maximum value is 9999999. The user specifies octal by using a postradix of B.

The initial value for the GTDL is zero. When the GTDL is zero, no TAF dumps from tasks are allowed. The GTDL must be set to a value greater than zero for TAF dumps from tasks to occur. Each TAF dump (where the task does not abort) causes the GTDL to be decreased by one. This command does not affect the execution of task dumps.

This section describes automatic recovery as it affects task writing. Recovery-related information applicable to the applications programmer is covered. Additional information of interest to the system installation and maintenance personnel can be found in section 8 of this manual.

PURPOSE

The purpose of the TAF automatic recovery feature is to enable site personnel to design TAF transactions that can recover quickly and predictably from system failures, and maintain the data base in a consistent state across failures. Terminal operators will be able to determine the status of transactions upon recovery so the possibility of issuing a double update to the data base or of missing a data base update is greatly reduced. The recovery will be automatic in the sense that TAF and the data manager will determine which data base updates to roll back, and which transactions to restart upon initialization after a failure. Site personnel need only select recovery mode when bringing TAF back up.

CONCEPTS

There are two basic components to automatic recovery: data base recovery and transaction recovery. Data base recovery ensures that the data base is always in a consistent and known state, even if a failure occurs during a run unit. Transaction recovery automatically reruns recoverable run units that were active at the time of a failure. It also guarantees that the terminal operator will receive a confirmation message when the run unit completes.

NAMED-TRANSACTIONS AND RUN UNITS

A transaction is a task or a series of linked tasks. The linking can be internal to the tasks using the CALLTSK and/or CALLRTN requests, or external to the tasks using the LIBTASK /tname directive. Refer to section 10 for a complete description of this directive. Transactions defined using the /tname directive are referred to as named-transactions. In most discussions in this section and throughout this manual, it is not necessary to distinguish between the coding that comprises a transaction and the execution of that code. However, when the distinction is important, the execution of a transaction is referred to as a run unit. The terms executing transaction, running transaction, and run unit can be used interchangeably. A run unit is recoverable only if it is the execution of a named-transaction and the RC parameter was specified when the named-transaction was defined. There are differences in terminology between CDCS and TAF. What is referred to as a transaction in TAF is always called a run unit in CDCS. The processing bounded by the begin statement (DB\$BEG or BEGTRAN) and the commit statement (DB\$CMT or COMMITRAN) is referred to as a transaction in CDCS. Each of these transactions has an id supplied within the application.

DATA BASE RECOVERY

Automatic data base recovery is available to transactions using the CDCS and/or the TAF/CRM Data Managers. Data base recovery is based on the concept of a begin-commit sequence. A begin-commit sequence is a portion of a transaction, designated by the programmer, that contains one or more data base updates that are to be treated as a unit. Should a failure occur during a recoverable run unit, data base recovery ensures that all the updates within a begin-commit sequence are performed or that none of them are. It also provides a means by which the transaction can determine which begin-commit sequences had successfully completed prior to failure so that on restart, the transaction can avoid performing those updates a second time.

Data base begin-commit sequences are identified in a transaction by placing a data base BEGIN request at the beginning of the sequence and a data base COMMIT request at the end of the sequence. (The BEGIN and COMMIT requests have different forms depending on the data manager and source language being used.) The data base BEGIN request includes a parameter for assigning an identifier to the begin-commit sequence. This identifier (selected by the programmer) is preserved across failures and can be retrieved upon restart to enable the transaction to determine how far processing had gone prior to failure. The data base COMMIT request tells the data manager that the updates are to be made permanent. A failure before a data base COMMIT causes all updates for this sequence to be rolled back. This means that the data base is returned to the state it was in at the time the BEGIN for this sequence was issued. If a failure occurs after a data base COMMIT, all updates for the committed sequence are retained. A transaction can terminate an update sequence without committing the updates by issuing a data base FREE request.

TRANSACTION RECOVERY

Transaction recovery restarts recoverable run units that were active at the time of a system or terminal failure. Transaction recovery also uses the concept of a begin-commit sequence. With transaction recovery, the BEGIN is implicit in the initiation of any recoverable run unit. The transaction COMMIT is specified with a SECURE request indicating that the work of the transaction is complete. The SECURE request should immediately precede the last data base commit request in the transaction.

When a recoverable run unit is initiated from a terminal or a batch job (via a BTRAN request), TAF logs the initial input on the communication recovery file. If a failure occurs during the run unit, automatic recovery takes place when TAF is subsequently brought back up. At that time the data base recovery routine is called in to restore the data base to a consistent state.

Then, the transaction recovery routine examines all run units that were active at the time of failure to determine which should be rerun. For run units initiated from a terminal, a flag is set in the terminal status table and the decision is postponed until the terminal logs back in. For run units initiated by BTRAN, the decision and restart, if necessary, are made immediately. This decision process is summarized in the following steps.

1. Is this transaction designated as recoverable?

YES NO

Go to step 2. Do not rerun.

2. Does this transaction use CDCS?

YES NO

Rerun. Go to step 3.

3. Was the transaction SECURE at the time of failure?

YES NO

Go to step 4. Rerun.

4. Did an incomplete begin-commit sequence exist at the time of failure?

YES NO

Rerun. Send SECURE message. Do not rerun.

CODING

Automatic recovery requires a transaction to contain a specific set of requests in a predetermined order. The following outlines show some suggested structures for recoverable transactions.

The OPEN, CLOSE, and LOCK requests need not appear in the places shown. The restrictions placed on these requests by TAF/CRM and CDCS automatic recovery are as follows:

- OPEN

An OPEN request cannot occur within a begin-commit sequence in a transaction using CDCS.

- CLOSE

A CLOSE request cannot appear within a begin-commit sequence in any transaction.

- LOCK

A record lock conflict within a begin-commit sequence causes all updates in the sequence to be rolled back. If this happens, the

begin request must be reissued and the update process restarted. Therefore, whenever possible, TAF/CRM transactions should request the record locks before making the DBEGIN request. For transactions using CDCS, the lock request is implicit in the record READ.

CODING STRUCTURES FOR DATA BASE RECOVERY ONLY

Transactions Using CRM

A transaction that uses the CRM Data Manager may be structured as follows. In addition, the recovery parameter must be specified on the xxJ file CRM statement for any file to be updated by this transaction. Although only one begin-commit sequence is shown, actual transactions can contain as many separate (nonnested) begin-commit sequences as desired. Multiple data base updates are permitted within a begin-commit sequence.

```

OPEN request(s).
Record lock request(s).
DBEGIN request.
.
.
(data base processing/updating)
.
.
DBC COMMIT or DBFREE request.
CLOSE request.

```

TAF Transactions Using CDCS

TAF transactions using the CDCS data management system may be structured as follows to take advantage of data base recovery. Although only one begin-commit sequence is shown, actual TAF transactions can contain as many separate (nonnested) begin-commit sequences as desired. Multiple data base updates are permitted within a begin-commit sequence.

In all the CDCS outlines in this section where a FORTRAN request name is different from the equivalent COBOL request name, the FORTRAN name is given in parentheses.

```

OPEN file request(s).
DB$BEG (BEGINTRAN) request.
.
.
(data base processing/updating)
.
.
DB$CMT (COMMITTRAN) or DB$DROP (DROPTRAN) request.
CLOSE file request(s).

```

CODING STRUCTURES FOR TRANSACTION RECOVERY

Transactions Using CRM

Recoverable transactions using the CRM Data Manager can be divided into two categories: single begin-commit sequence transactions and multiple begin-commit sequence transactions. A single begin-commit sequence transaction needs only to include a SECURE request immediately before the COMMIT request. If a failure occurs before the COMMIT request is successfully executed, the data base will automatically be rolled back and the transaction rerun at recovery time. A multiple begin-commit sequence transaction must first determine if recovery is in progress. If it is, the transaction must then discover which was the last successfully committed begin-commit sequence and begin processing with the following begin-commit sequence. For this purpose, the data base BEGIN requests require a user-supplied identifier to be included as part of the request. All identifiers used within a single transaction must be unique for the restart point to be unambiguous.

Single Begin-Commit Sequence Transaction

```
OPEN files.
Record lock request(s).
DBEGIN request.
.
.
.
(data base processing)
.
.
.
SECURE request.
DBCOMIT request.
CLOSE request.
transaction CEASE -- SECURE message is sent to
terminal.
```

In the preceding transaction, if a failure interrupts processing any time prior to the DBCOMIT request, TAF automatic recovery reruns the transaction from the beginning. If a failure interrupts processing after the DBCOMIT request, automatic recovery only sends the SECURE message to the terminal.

Multiple Begin-Commit Sequence Transaction

```
OPEN files.
TSTAT request using keywords RESTART, OLDID.
If RESTART value is zero or OLDID value is
blank, begin processing at SEQN1.
If OLDID is SEQN1
begin processing at SEQN2.
.
.
.
If OLDID IS SEQN(n-1)
begin processing at SEQNn.

Processing Module(s)
.
.
.
```

(data base processing)

```
.
.
.
SEQN1.
Record lock request(s).
DBEGIN request, begin-commit id = SEQN1.
Perform processing module(s).
DBCOMIT request.
SEQN2.
Record lock request(s).
DBEGIN request, begin-commit id = SEQN2.
Perform processing module(s).
DBCOMIT request.
SEQN3.
.
.
.
SEQNn.
Record lock request(s).
DBEGIN request, begin-commit id = SEQNn.
Perform processing module(s).
SECURE request.
DECOMIT request.
CLOSE files.
transaction CEASE -- SECURE message is sent to
terminal.
```

TAF Transactions Using CDCS

A recoverable TAF transaction using CDCS must first determine if recovery is in progress. If it is, the TAF transaction must determine the last successfully committed begin-commit sequence. This is necessary even for single begin-commit sequence transactions because all recoverable run units using CDCS are rerun during recovery regardless of whether the last begin-commit sequence was successfully committed. The failure may have occurred after the COMMIT request and before the transaction CEASE. Rerunning from the start would cause a double update for the transaction. The transaction must begin processing at the first begin-commit sequence following the last successful COMMIT. If the final begin-commit sequence was successfully committed prior to failure, the transaction need only retrieve and send the SECURE message to the terminal.

Figure 7-1 shows the structure of a recoverable, multiple begin-commit sequence transaction that uses CDCS. The decision process that determines where to begin during recovery is shown in expanded form for clarity. The use of subscripted variables for begin-commit ids would allow for condensed code in the case of a large number of begin-commit sequences.

CDCS applications programmers should be aware of some terminology differences between TAF automatic recovery documentation and CDCS automatic recovery documentation. The term transaction in CDCS documentation refers to what TAF documentation calls a begin-commit sequence; that is, all processing starting with a data base begin request and ending with the corresponding data base commit request. Similarly, CDCS documentation uses the term transaction-id to refer to what TAF documentation calls the begin-commit id.

```

OPEN file(s).
TSTAT request, Keyword RESTART.
If RESTART value is zero
    DB$CTID (ASSIGNID) request
    RPUT request
    begin processing at SEQN1.
ELSE
    RGET request.
DB$ASK (FINDTRAN) request.
If the begin-commit id is blank
    begin processing at SEQN1.
If the begin-commit id is SEQN1
    begin processing at SEQN2.
IF the begin-commit id is SEQN2
    .
    .
    .
If the begin-commit id is SEQN(n-1)
    begin processing at SEQNn.
If the begin-commit id is SEQNn or *****
    begin processing at WRAP-UP.

Processing Module(s)
    .
    .
    (data base processing/updating)
    .
    .

SEQN1.
    DB$BEG (BEGINTRAN) request, begin-commit id = SEQN1.
    Perform processing module(s).
    DB$CMT (COMMITRAN) request.
SEQN2.
    .
    .
    .

SEQNn.
    DB$BEG (BEGINTRAN) request, begin-commit id = SEQNn.
    Perform processing module(s).
    SECURE request.
    DB$CMT (COMMITRAN) request.
    RESTART-value equal to zero.
WRAP-UP.
    If RESTART value is not zero
        RSECURE request
        SEND request
        CLOSE request(s).
        Transaction cease.

```

(If not being recovered.)
 (Get new RESTART-id.)
 (Save new RESTART-id on CRF.)
 (Skip to first begin-commit sequence.)

 (Retrieve old RESTART-id from CRF)
 (Get id of last committed begin-commit
 sequence. This also assigns the old
 RESTART-id to this transaction.)

 (Retrieve SECURE message from CRF.)
 (Send SECURE message to terminal.)

Figure 7-1. Recoverable TAF Transaction Using CDCS

TERMINAL RECOVERY CONSIDERATIONS

If a communication subsystem, TAF, or system failure occurs, an application cannot assume that the terminal contains any particular set of data. The last message may not have been transferred to the terminal, or the terminal operator may not have noticed the message. Also, login procedures or terminal failures may destroy the screen.

Terminal recovery is also dependent upon the type of terminal and the form used on the terminal. On hard copy terminals, the terminal may not need all the data resent. In fact resending the data may spoil the form. On cathode ray tubes (CRTs), the data

entered by the operator may need to be sent after recovery to indicate what still needs specification.

Applications must design their terminal dialog so that the operator has no doubts about the state of the run unit by sending a final message via SECURE indicating unmistakably that the run unit was successfully completed. The transaction system guarantees that the operator will see this message in all cases when the run unit completed successfully; otherwise, the operator might reenter the transaction causing a double update of the data base.

Applications that require many terminal interactions must reconstruct the screen from the data base upon recovery.

CALLTSK WITHOUT CEASE AND RECOVERY

A CALLTSK without CEASE starts a new run unit. This new run unit is not recoverable. It will be restarted only if the calling task is part of a recoverable run unit and the task makes the CALLTSK request again upon restart.

Care is necessary when using SECURE, RSECURE, RPUT, and RGET from run units initiated by CALLTSK without CEASE. Such run units may write or read the same recovery data being written or read by the originating run unit.

MULTIPLE DATA MANAGER USE

Correct recovery cannot be assured for nested or overlapping begin-commit sequences using both CDCS and CRM. A transaction may use both CDCS and CRM

provided that begin-commit sequences are neither nested nor overlapping.

TOTAL DATA MANAGER USE

The TOTAL Data Manager does not use the concept of begin-commit for data base recovery. Applications using this data manager, however, may implement data base recovery using current TOTAL features and TAF automatic recovery. If TAF automatic recovery is installed, such tasks may use any of the requests described in section 2 of this manual, including the application recovery requests.

During recovery, transactions that do not use CRM or CDCS and that were active at the time of failure are rerun unless the SECURE request data resides on the communication recovery file. If the SECURE message is present, the transaction is not rerun but the SECURE message is sent to the terminal.

Faint, illegible text in the top left section of the page.

Faint, illegible text in the top right section of the page.



This section contains recovery information of interest to persons responsible for the installation and maintenance of the transaction subsystem.

RECOVERY OVERVIEW

The TAF recovery package consists of two major components: transaction recovery and data base recovery. Data base recovery can be further subdivided into the automatic component and the batch component. The integrated system works like this:

A transaction is declared to be recoverable when it is added to the application task library. The data managers (CRM and CDCS) provide methods of declaring data files recoverable. Upon initiation of a recoverable transaction, ITASK schedules the transaction via a CALLTRN request. Then the original input (that is, the transaction name and all unsolicited data) is logged to the communication recovery file before the transaction begins execution. The transaction must indicate to the data manager that it wishes to open a begin-commit sequence by making a data base BEGIN request. When the transaction requests to update a data file, the following occurs.

- The record as read from the data file (the record in this state is called the before image) is logged to a before image recovery file, also known as the before image log file.
- The record is processed by the transaction.
- The updated record (called the after image) is logged to an after image recovery file when the program makes a write, rewrite, or delete request. The updated record is also written to the data file at this time.

After an update or a series of updates, the transaction indicates to the data manager that it wishes to close the begin-commit sequence by making a data base COMMIT request. If the transaction does not intend to open another begin-commit sequence, it indicates to TAF that its work is done by issuing a SECURE request just prior to the COMMIT request. TAF logs a message associated with the SECURE request to the CRF and sends this message to the terminal when the final transaction CEASE request is processed.

If a failure occurs prior to the BEGIN request, the data base is unaffected and the transaction is restarted upon recovery using the original data logged on the CRF.

If a failure occurs between the BEGIN and COMMIT requests, the data manager rolls back the data base by applying the before images from the before image recovery file (BRF) to the affected data files. The data base is then in the same state as it was at the beginning of the begin-commit sequence. The transaction is restarted upon recovery using the data on the CRF.

If a failure occurs after the COMMIT request, the data base is not rolled back, all updates become permanent when the COMMIT is processed. Whether the transaction is restarted depends upon the data manager being used. CDCS transactions are restarted if a failure occurs any time prior to the final transaction CEASE request. CRM transactions are rerun unless both a COMMIT and SECURE request have been successfully issued. The applications tasks must be designed to handle cases where restart can occur after a COMMIT has been successfully executed. Refer to section 7 for some suggested transaction structures.

A BRF file is used repeatedly by the application. Only the before images for the current begin-commit sequence need to be saved so each new begin-commit sequence can overwrite the previous one.

An after image recovery file (ARF) is continuous. New after images are written sequentially until the file is full. At that time, logging is switched to a second ARF while the after images from the first ARF are dumped to tape.

The ARF files are not used for automatic recovery. They provide a backup in case a CRF, BRF, or data file becomes damaged. If this happens, the data administrator can use batch recovery to reload the data file(s) from the dump tape and apply after images from an ARF dump, thus restoring the data base to a correct state as of any desired point in time.

The function performed by each of the recovery components are as follows.

- Transaction recovery
 - Logs the initial input to the CRF.
 - Logs and sends the SECURE message.
 - Restarts recoverable transactions after a failure.
- Data base recovery
 - Automatic component
 - . Rolls back updates in uncommitted begin-commit sequences.
 - . Logs before images to the BRF.
 - . Logs after images to the ARF.
 - . Switches after image logging when an ARF is full and submits a batch job to dump the full ARF to tape.
 - Batch component
 - . Dumps full ARFs to tape.
 - . Dumps data files to tape upon request.

- Restores data base files from data file and ARF tapes.
- Maintains a directory of dumped files.

The nomenclature used here is specific to the TAF/CRM Data Manager. CDCS automatic recovery uses similar concepts but different terminology. The details associated with data base recovery are documented in the respective data manager reference manuals. The remainder of this section covers the details of transaction recovery.

HARDWARE CONFIGURATION CONSIDERATIONS

Recovery cannot succeed if failures affect more than one component of the recovery system. For example, if a single hardware failure destroys both the recovery files and the data base, there is no recovery other than to restore the data base from a backup copy (file dump). For this reason, the data base files, the communication recovery files, the before image recovery files, and the after image recovery files should be assigned to physically separate devices. Also, the backup directory file for batch recovery should not reside on the same device as the ARFs.

The dump tape storage area should be physically separate from the area housing the disk drives.

Automatic recovery also depends on the integrity of single input/output operations. An on-line motor-generator is essential to ensure that disk operations underway at the time of a power failure have time to complete correctly.

Performance improvements through the use of multiple CRFs and BRFs (TAF/CRM only) will be realized only if each additional file resides on a separate device.

INSTALLATION

A procedure file for TAF initialization and recovery is presented in the NOS 2 Installation Handbook.

Transaction recovery is enabled or disabled by the installation parameter IPTAR. If IPTAR is set to one, transaction recovery is enabled; if IPTAR is set to zero, transaction recovery is disabled.

If recovery is disabled, these requests will not be honored in recovery mode:

CALLTRN	For logging of input on CRF.
RERUN	
RGET	
RPUT	
RSECURE	
SECURE	
TINVOKE	
TSTAT	Except for keywords USER and NEXT.
WSTAT	Except for keywords STEP (=8 or =9) and USER.

A number of other installation parameters have an effect on recovery. Refer to the NOS 2 Installation Handbook for details.

TAF K DISPLAY

INITIAL K DISPLAY

Three K display commands are available to perform recovery-related functions: K.INT, K.ERO, and K.REC.

K.INT Command

The K.INT command specifies which recovery file(s) are to be initialized. This is the only means of initializing a CRF. Files listed on a RECOVER statement in the TCF and on this command will be initialized. Files listed on a RECOVER statement but not on this command will be used for recovery. The INT command also specifies whether CRM data base recovery files (ARFs and BRFs) are to be initialized or recovered. When used for CRM recovery files, this command is valid regardless of TAF installation options.

The format is:

K.INT=type,fileid

type File type. The options are:

CRF Command relates to the communication recovery file. This option valid only if installation parameter IPTAR = 1.

CRM Command relates to CRM recovery files.

fileid File identifier. The options are:

digits Each digit in the string specifies a CRF to be initialized. The digit corresponds to the id parameter on the RECOVER statement in the TCF. This option is not valid if type is CRM.

ALL If type is CRF, all CRFs listed in TCF RECOVER statements are to be initialized. If type is CRM, all CRM after image recovery files are to be initialized.

NOTE

This option must be used with caution as the CRM data base update history currently residing on the recovery files will be lost.

NONE If type is CRF, no CRF is to be initialized (all CRFs listed in TCF RECOVER statements are to be recovered). If type is CRM, no CRM recovery files will be initialized. NONE is the default for both values of type.

K.ERO Command

The K.ERO command specifies whether to override certain I/O or logical errors encountered during processing of the communication recovery file(s). The format is:

K.ERO=CRF,option.

option Error override option. Must be NO or YES.

NO Abort if I/O or logical errors are encountered when processing the communication recovery file(s). This is the default.

YES I/O or logical errors encountered on a run unit header record will result in the loss of that run unit with no indication to the terminal user. The loss will be noted on the recovery report.

I/O or logical errors encountered on a message record within a run unit will result in a loss of that run unit. However, a status indicator would be set in the run unit header that would enable the terminal user to be informed.

I/O or logical errors encountered on the CRF header record will result in unconditional abort of TAF.

K.REC Command

The K.REC command sets the user recovery bit in the terminal status table. The format is:

K.REC=nnn

nnn Recovery option. Must be YES or NO.

YES Set the user recovery bit in the terminal status table.

NO Do not change the present setting of the user recovery bit.

RESTART K DISPLAY

The K display that TAF formats at initialization will be displayed only if a DISPLAY statement with the ON parameter selected appears in the TCF. Refer to section 4 for details on the DISPLAY statement. If the K display is brought up, the operator makes any desired changes and types K.GO or K.END to resume processing. K.STOP will abort TAFREC.

On a level 3 deadstart, or if TAF aborts, the TAF procedure file automatically restarts TAF by transferring control to TAFREC. TAFREC recovers the low CM pointers and variables that were defined at TAF initialization via the initial K display.

If DISPLAY,ON is selected in the TCF, the K display is issued. This display is identical to the initial K display except that if any values were changed at initialization, the changed values are retained on the restarted K display. This does not apply to K.INT and K.ERO commands. Changed values for these commands are not retained after TAFREC completes. (This prevents inadvertent initialization of recovery files in a recovery situation).

TCF/CRF RECOVERY CONSIDERATIONS

TAF CONFIGURATION FILE

The TCF contains several recovery-related items.

- The RECOVER statement indicates the CRF identifier (zero to seven) and the size and number of messages that can be written to the CRF.
- Default K display values can be changed by placing K display commands in the TCF.
- The DISPLAY statement provides the option of having the K display brought up at initialization and recovery to allow the operator to override any of the default values.

Details on these statements are given with the TCF description in section 4.

The use of multiple CRFs is indicated by multiple RECOVER statements in the TCF. Each RECOVER statement corresponds to a separate CRF. For each RECOVER statement there must be a corresponding NETWORK statement. Refer to Use of Multiple Communication Recovery Files in this section and Communication Recovery File in section 4.

COMMUNICATION RECOVERY FILE

Error Override

Certain errors on the CRF file(s) can be overridden by using the K.ERO K display command (refer to Initial K Display in this section). Input/output errors on the CRF encountered during the running of transactions cause TAF to abort and TAFREC to attempt recovery of all readable data from the CRF.

File Reformatting

TAF recovery processing reformats the CRF under the following circumstances.

- If the number of user messages per recovery unit is increased in the number parameter on the RECOVER statement.
- If the size of user messages per recovery unit is increased in the maximumsize parameter on the RECOVER statement.
- If a new user is added to or deleted from the corresponding network file.

The RECOVER statement cannot be used to reduce the number or size of the user messages. The only way to reduce either of these is to reinitialize the CRF using the K.INT command.

To reformat ZZCRF*i*, TAF creates a permanent file named ZZCRFA*i* and writes the reformatted CRF to it. When the reformat is complete, the old CRF (ZZCRF*i*) is purged and the new CRF (ZZCRFA*i*) has its permanent file name changed to ZZCRF*i*.

Multiple Communication Recovery Files

Up to eight CRFs may be active simultaneously. Each CRF must have a unique identifier. Once a CRF has been initialized with a given id, it must be brought up with the same id. If this is not done, recovery will fail. Advantages of using multiple CRFs are:

- The input/output associated with the CRFs is spread out across several disks. This improves performance.
- Multiframe recovery becomes possible. Refer to Multiframe Recovery in this section.

MULTIFRAME RECOVERY

The guidelines in this subsection pertain to sites running TAF on several mainframes. These guidelines are applicable if the ability to recover transaction processing for a downed mainframe on an alternate mainframe is desired.

- Make certain that the network (NCTF*i*) and CRF (CRF*i*) file identifiers (*i*) on each mainframe are unique among all mainframes.
- Allocate unique data base names to all applications across all mainframes.
- Allocate recovery-related files to shared mass storage.
- Define unique terminal user names across all mainframes.
- When a mainframe goes down, IDLE TAF on the mainframe that is to run the downed applications.†
- Restart TAF with a TCF file that contains RECOVER statements for the recovery ids of both mainframes.
- When the downed mainframe is repaired, IDLE TAF on the recovery mainframe. Then bring TAF up independently on the two mainframes with the original TCFs.

RECOVERY PROCESSING FLOW

TAF AND SYSTEM FAILURE

Following a TAF or operating system failure, the TAF initialization procedure brings TAF up in recovery mode. During recovery mode initialization, TAFREC and TAF perform the following functions:

†If CDCS has been running on both mainframes and interfacing with TAF, CDCS must be dropped on the functional mainframe or recovery of CDCS data bases will not be possible.

- Sets the K display values to those in effect prior to failure.
- Identifies the users that were active at the time of failure. Active users will have the recovery flag (word 1, bits 0 and 53) set in the terminal status table (TST).
- Reformats the CRF if necessary (refer to File Reformatting under Communication Recovery File in this section).
- Prepares the TAF automatic recovery report.
- Initializes the transaction sequence number count.
- Schedules CTASK to move begin-commit ids from BR*F* to CRF*i*.

BTASK is scheduled to determine which users active at the time of failure were running recoverable BTRAN-initiated transactions. BTASK calls CTASK to recover these transactions.

Recovery for transactions being run from interactive terminals is postponed until the terminal logs in. When a terminal logs in, TAF checks the recovery flag (word 1, bit 53) in the terminal status table. If the recovery flag is set, RTASK is scheduled instead of ITASK. RTASK performs these functions:

- If the run unit is not recoverable, RTASK sends READY to the terminal and ceases.
- If a recoverable transaction is no longer rerunnable, RTASK sends a message telling the terminal operator
 1. that recovery has occurred,
 2. that the recoverable transaction will not be rerun automatically, and
 3. to enter the next input.

This situation occurs when an input/output error occurred on the communication recovery file. Terminal operators should be instructed to inform the data administrator if this occurs since batch recovery will be necessary to restore the data base.

- If the transaction SECURE request and a TAF/CRM COMMIT request have been successfully executed, RTASK sends the terminal confirmation message, and ceases. The terminal confirmation message was stored on the CRF as a result of the transaction SECURE request.
- If none of the preceding is true, RTASK calls CTASK.

CTASK requests a new transaction sequence number and returns the begin-commit history to TAF/CRM. CTASK then makes a RERUN request. The RERUN request turns off the recovery status, retrieves terminal input from the CRF, and schedules ITASK using the new sequence number and original terminal input. ITASK processes the input as if it had come from the terminal.

NAM FAILURE

If NAM fails, TAF continues running. The rollout table is searched to suspend transactions that were initiated by an interactive terminal and are waiting for input (the wait time in the rollout table is set to zero). All other active transactions continue to run. When a running transaction issues a network request, it is suspended if recoverable and the request is issued to the original terminal (without recall in the case of SEND). If the transaction is not recoverable, it is terminated. Suspended transactions are processed as follows:

- A terminal-login-delay flag is set in the terminal status table to prevent terminal login until the next step is completed.
- CTASK is scheduled to recover the data base associated with the transaction and to issue a WSTAT request to clear the delay flag.
- At the time the terminal logs in, RTASK is scheduled either to send the SECURE message or to rerun the transaction.

TAF will periodically attempt reconnection to NAM. When NAM becomes available, terminal operators will have to go through the login dialog required for the terminal. Upon terminal reconnection to TAF, TAF issues an appropriate message to tell the terminal operator the current state of the terminal work.

TERMINAL FAILURE

Processing for an individual terminal failure is the same as described for NAM abort.

CDCS FAILURE

If CDCS fails, TAF continues running. When TAF detects a CDCS abort, it does the following to run units using CDCS:

- If the run unit is recoverable, its processing is suspended and an informative message is issued. The terminal operator may wait for the run unit to be automatically rerun when CDCS becomes available, or may enter a new transaction. In the latter case, the run unit will not be rerun when CDCS becomes available.
- If the run unit is not recoverable, it is aborted and an error message is issued to the originating terminal.

TAF sets a flag in the CRF for suspended run units waiting to use CDCS.

TAF, on a periodic basis, will check to see if CDCS is running and ready for transaction processing.

When CDCS is again accepting requests, TAF will rerun those recoverable run units whose processing was suspended, as previously described.

TASK FAILURE

TAF does the following when a task fails:

- TAF sends a failure or end run unit message to the data manager. The data manager rolls back updates in any incomplete being-commit sequences.
- TAF logs the reason for the failure in the journal file.
- TAF calls system task MSABT. The Control Data-furnished version of MSABT sends a message explaining the error condition to the terminal. Sites may modify MSABT to meet special requirements.
- TAF reloads the task if the task has other run units queued on it or is central memory-resident.
- Run units aborted because task failures are not rerun. Failure of the logged in terminal, NAM, or CDCS does not cause a task failure for a recoverable run unit.

TAF TERMINATION

TAF goes through termination processing when:

- The computer operator or terminal operator forces execution of a K.IDLE request.
- The computer operator enters IDLE,TAF.
- TAF (including any data manager in its field length) fails.
- A level 3 deadstart occurs.

When TAF receives a K.IDLE or IDLE,TAF command, TAF tells ITASK with an idle message. ITASK then:

- Tells terminal users entering a new transaction that shutdown is occurring.
- Prohibits new transactions from starting.

When no more transactions are running (no communication blocks are reserved), TAF does an ENDRUN to give control to TAF2. TAF2 tells the data manager(s) about shutdown and records a shutdown status on the CRF. When TAF is subsequently brought up, no recovery takes place since all users were inactive at termination.

TAF AUTOMATIC RECOVERY REPORT

This report is produced by TAFREC and placed on local file ZZIRRF. A site may save the report on permanent storage and/or route the report to a printer by including appropriate statements in the TAF procedure file. The report consists of four parts.

- Part one is a copy of the TAF configuration file. Each directive is printed. If a directive error occurs, a message is printed on the following line.
- Part two is an image of the initial K display. If the same command is used repeatedly, only the last image of that command is listed.
- Part three is a report on each CRF processed. The format of this part is shown in figure 8-1.
- Part four is a summary consisting of the total number of terminals recovered, initialized, and deleted, and the number of concurrent batch and BTRAN requests recovered.

TERMINAL	TSEQ	TYPE	STEP	DATE	TIME	COMMENT
TERM1	1234	TERM	REC	82/04/26.	08.20.1.	RECOVERED
TERM2	1240	BATCH	END	82/04/26.	08.20.11.	RECOVERED
TERM3				82/04/26.	08.25.20.	INITIALIZED
TERM4	1251	BTRAN	REC	82/04/26.	08.20.11.	RECOVERED

terminal Terminal/user name.
tseq Transaction sequence number of the last run unit.
type Transaction type (refer to TSTAT request).
step Transaction step (refer to TSTAT request).
date Date of last recorded run unit.
time Time of last recorded run unit.

Figure 8-1. Format of Part Three of the TAF Automatic Recovery Report

Applications programmers should determine whether an analyst is preparing the network files. If so, applications programmers can skip this section.

Before any transaction processing can be done, the terminal configuration must be defined to the transaction executive. This is done by the terminal network description file (network file).

The NAM local configuration file is also needed to describe the terminal configuration. This file is described in the Network Products NAM 1 Network Definition Language Reference Manual.

TERMINAL NETWORK DESCRIPTION FILE

A terminal network description file consists of directives that describe users and their data bases. The directives needed consist of the //DIAL directive and user directives.

The format is:

```
//DIAL.
```

User directives must follow the //DIAL directive. User directives have the form:

```
/username,parameter-string.
```

username	A user name that is used to log into the network and TAF (or is specified in the local configuration file for automatic login). User names must be unique across all network files, even if they reside on different families.
----------	--

The possible elements of the parameter string, separated by commas, are:

TT=*ID	Transaction terminal type.
DB=aa	Two-character name of the data base to be accessed by the user. Only one data base can be associated with a user name.
UA=n	Contents of the user argument area in the communication block. n ranges from 0 through 16777215. Default is zero. User area contents can be examined and changed by tasks. The appropriate initial value is site dependent. Bit 0 is the default task recovery bit.

A simple set of terminal network file directives follows:

```
//DIAL.
/AJOHNSN,TT=*ID,DB=BC,
UA=16117711B.
/BJOHNSEN,TT=*ID,DB=BC,
UA=00007700B.
/XYZ1,TT=*ID,DB=BC.
```

NETWORK FILE ORIGINATION

When an error-free network description is ready, an analyst should create a system file called NCTFi (where i is the network file identifier) from the terminal network description file.

A typical way to originate an NCTFi file follows. First, the user establishes a set of network description statements as a local file.

```
NCTF.
USER,username,password,familyname.
CHARGE,chargenumber,projectnumber. or (*)
COPYBR,INPUT,NET.
SAVE,NET.
end-of-record
Network description statements
end-of-file
```

Then the user enters the following at the system console.

```
X.DIS.
USER,username,password,familyname.
CHARGE,chargenumber,projectnumber.
GET,NET.
SUI,377777.
DEFINE,NCTFi/CT=PU.
COPY,NET,NCTFi,V.
RETURN,NCTFi,NET.
```

This creates a direct access public file called NCTFi under the system user index 377777. NCTFi is used to generate terminal status table entries.

Figure 9-1 illustrates a terminal status table entry. Each line gives the user name and its associated data base, read and update securities, and user area value.

When a run unit originates from a terminal, the transaction executive uses the appropriate entry in the terminal status table to construct words 0 and 1 of the communication block.

After the network file is originated, it must be updated to reflect changes, and the transaction executive must be reinitialized to recognize the new information.

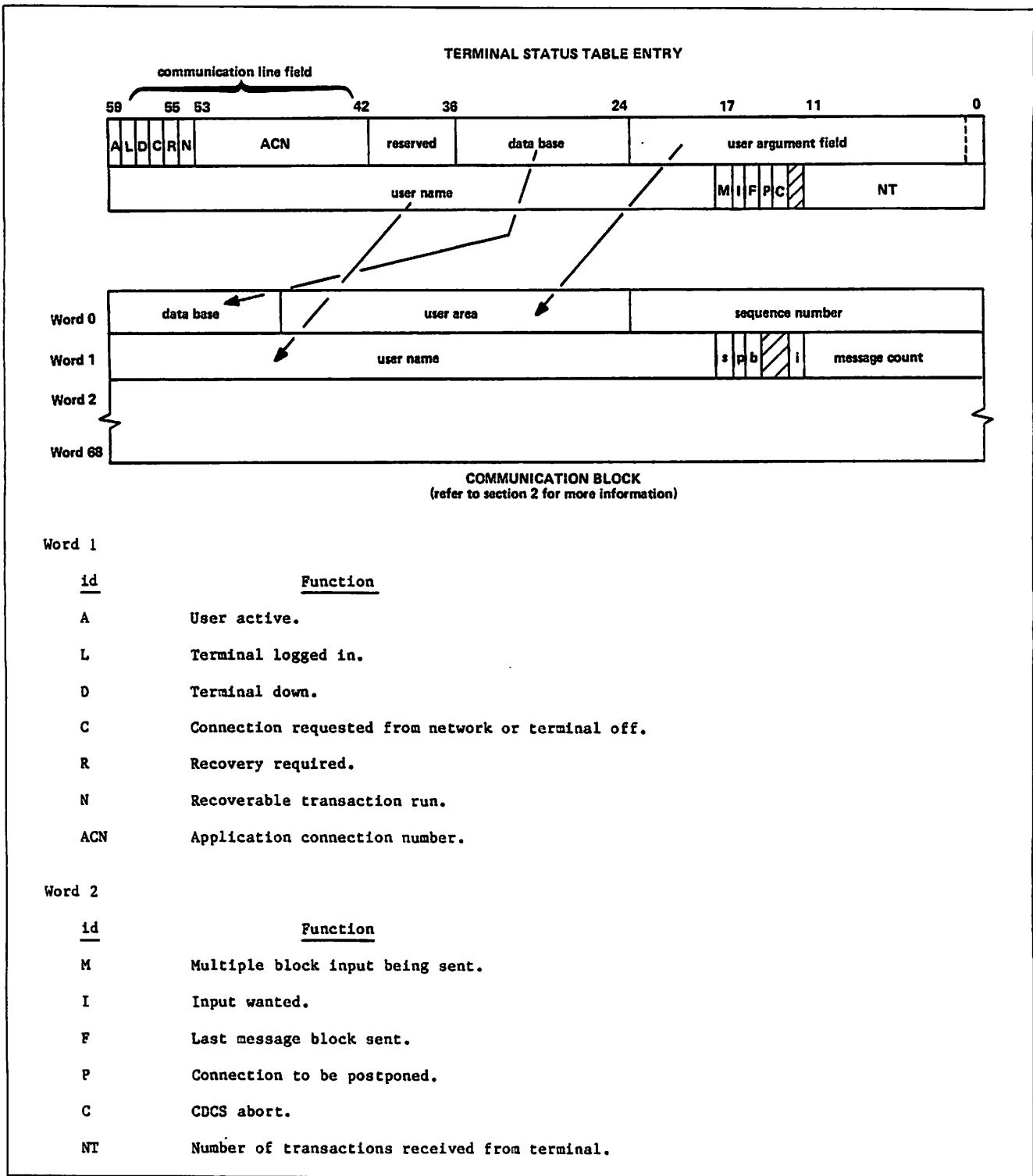


Figure 9-1. Communication Between a Terminal Status Table and a Communication Block

A task library contains all of the named-transactions and tasks that may be requested by a user group within the transaction subsystem. The task library is created by the LIBTASK utility.

A task library file is a direct access permanent file with no password that must be created with a DEFINE command before a library creation run occurs. However, it does not have to be attached by the user unless the DA option is specified (refer to LIBTASK command). The transaction subsystem simultaneously supports one system task library and, optionally, one task library for each data base.

The system task library, called TASKLIB by default, may be renamed at the site's option. It is defined under the transaction subsystem user name and given write permission to that user name. The name TASKLIB may also be changed using the K display command K.TLF. (Refer to the NOS 2 Operator/Analyst Handbook.)

A user task library must be named xxTASKL where xx is the name of the data base with which the library is associated. It must be defined under the data base user name, and explicit write permission must be granted to the transaction subsystem user name to access the file.

During initialization, TAF attempts to attach a library for each initialized data base. The xxJ file is accessed to determine the user name needed to attach the library. When a scheduling request is received, the transaction subsystem searches for the named-transaction or task first in the associated data base library, if one exists, and then in the system library. Transaction and task names, therefore, need be unique only within a given task library. The only exception to this procedure is the initial task, ITASK, which is always loaded from the system library.

LIBTASK UTILITY

The LIBTASK utility creates a new library, adds new named-transactions and tasks to an existing library, deletes named-transactions and tasks from an existing library, replaces tasks in an existing library, and compacts an existing library by removing inactive records. Named-transactions and tasks may be added or deleted while the transaction subsystem is running, because tasks are added as a new file at the end of the library. The library is a multifile file. The transaction executive attempts to read a new library directory if the LIBTASK command contains the TT option. Thus, on-line update of any task library is allowed.

If the library is modified while the transaction subsystem is running and the user wants the new directory used, the user number and password of the LIBTASK user are passed to the transaction subsystem for validation. This is done by intercontrol point communication, which is also used to check if the transaction subsystem is running if the CR or PR option is selected. Therefore, the first USER command in the job must have permission to do intercontrol point communication. (Refer to the NOS 2 System Maintenance Reference Manual for the use of MODVAL to alter permissions.) The LIBTASK run issues an error message and does not update the library if the validation fails. The task binaries to be added or replaced are in ABS or OVL format produced by the CYBER Loader.

TASK RESIDENCY AND TYPE

An important option of LIBTASK enables the user to specify the residency and type of tasks on the task library. A task can reside on central memory, extended memory, or mass storage. This can be specified with the C and E options on LIBTASK directives.

A task can be of either reusable or nonreusable type. A reusable task must be written so that it is reinitializing. A reinitializing task (program) does not modify program areas that the program considers constant; if it does, it resets (reinitializes) these areas upon reentry. Multiple requests for a reusable task can be queued together; this is partly under the control of the person who sets up the task library and partly under the control of the transaction executive. The queuing of multiple requests occurs in an internal list, the requested task list (RTL). Each entry in this list represents a scheduling request for a copy of a task. There may be multiple entries in the RTL list for a given task. The LIBTASK queue limit input directive can be used to specify the maximum number of transactions that can be queued against one entry. When the transaction executive schedules a copy of a task from the RTL list, it picks an RTL entry and forces all of the multiple requests queued for that RTL entry to use the same copy of the task.

It is generally possible to have multiple copies of a given task in execution at the same time (note that this does not apply to CM-resident tasks). Additional application control of task scheduling is provided by the Q LIBTASK option in the input directives.

A nonreusable task is not a reinitializing task; a new copy of the task must be loaded from the task library for each transaction.

The task type can be specified with the D option on LIBTASK directives. The queue limit can be specified with the QL option.

If a task is specified as central memory-resident and reusable, it is loaded to a subcontrol point when TAF is initialized and, unless the task aborts, remains at that subcontrol point to service transactions as long as TAF is running. If the task aborts, another copy of it will be loaded. It is a single copy task. No additional copies of this task are loaded to process requests after queue limit has been reached.

LIBTASK COMMAND

The LIBTASK command has two formats. The first format allows specification of the file containing input directive statements; this format is:

```
LIBTASK(p1,p2,...,pn)
```

The second format specifies that input directive statements are included in this LIBTASK command. This format is:

```
LIBTASK(p1,p2,...,Z,...,pn);d1;d2;...;dn
```

; Unique delimiter that does not occur in any d_i input directives. This delimiter does not have to be a semicolon.

d_i Input directive statement (refer to Input Directives in this section).

The input directives on a LIBTASK command remain in effect for a task in an existing task library until the task is specified in a subsequent LIBTASK command; that is, the most recent input directives (including default input directives) for a task take precedence over earlier input directives. However, this is not true for a task that is not in a task library; in this case, the first input directives (including default input directives) take precedence.

For example, you specify the following command:

```
LIBTASK(p1,p2,p3,...,pn);*T1 DL.
```

This command logically deletes task T1. If you specify task T1 in a subsequent LIBTASK command and do not specify DL, the task is activated (by default).

The LIBTASK command cannot exceed 80 characters.

The parameters (p_i) for either format can be in any order and are:

Parameter	Description
I	Specifies the file containing input directives. The file is not rewound before processing. All directives that do not refer to a task name present on the B file or P file are ignored. Options are: I File is COMPILE.

Parameter	Description
I=lfn	File is specified by you.
I=0	No directive file exists.
Omitted	Default file is INPUT.
B	Specifies the file containing task binaries to be added to the library. These may be tasks already existing in the library or they may be new ones. The file is rewound before processing. Options are: B File is LGO. B=lfn File is specified by you. B=0 No binary file exists. Omitted Default file is LGOB.
L	Specifies the file on which output is written. Options are: L File is OUTPUT. L=lfn File is specified by you. L=0 No output is written. Omitted Default file is OUTPUT.
P	Specifies the name of a direct access permanent file to be used as the library file. The file is required. Options are: P File is TASKLIB. P=lfn File is specified by you. P=0 No old library file exists. Omitted Default file is TASKLIB.
N	Specifies a newly defined direct access permanent file to be written as the new library file. Use of this option eliminates the complete copy of the P file to a scratch file on the PR option, thus significantly reducing the purge time for long libraries. The P file is not purged. Options are: N File is TASKLIB. N=lfn File is specified by you. Omitted Default file is the file specified by P.

Parameter

Description

Parameter

Description

LO Produces a report of tasks and their attributes, named-transactions with attributes, and names of member tasks. It is not necessary to specify the TT option when using LO if I=0 and B=0. Figure 10-1 shows the format of the report. The options are:

- LO Same as LO=A.
- LO=A Report all tasks and transactions.
- LO=T Report only tasks.
- LO=X Report only transactions.
- omitted No report is produced.

TN Specifies the name of a task or transaction. This parameter is valid only when the LO parameter is also specified. If LO=T, TN names a task; if LO=X, TN names a transaction. The options are:

- TN=tname The task or transaction (tname) is the only one reported on.
- omitted All tasks or all transactions are reported on.

DA Disables all attaches and returns of the library files in the LIBTASK program. With this option selected, the N file and the file specified by P must be attached prior to the LIBTASK call. If a library is attached in the proper mode from another user name, the DA option must be selected so that LIBTASK does not try to attach the appropriate libraries under the user name from which the update is being run.

CR Creates a new task library. LIBTASK reads one task at a time from the B file, checks for an input directive for the task, constructs a directory entry, and inserts the entry in alphabetical order in the library directory. The directory is written at the end of the library. This option applies only if LIBTASK was able to attach the task library (specified by the N option) in write mode.

PR Removes inactive task binaries and old directories from the task library. LIBTASK processes the B file and the input directives normally and, if N is not specified, copies the entire library to a scratch file. The current binary for each task in the library is then transferred from the scratch file to the library file, and the directory is updated and written at the end.

Therefore, only one copy of each task and one directory remain in the library. If N is specified, no scratch file is written, and P is not purged. The new library file contains only the latest copy of each active task and one directory.

This option applies only if LIBTASK was able to attach the task library (specified by the N option) in write mode.

TT Instructs the transaction executive to update the TAF-resident copy of the task library directory while TAF is running. If not specified, TAF will not become aware of the update until the next time it is initialized.

The installation parameters TLDMN and TRDMN are used to specify the amount of space to be reserved for adding tasks and transactions for each of the TAF-resident task library directories. The default allows 10 additions for each. Although further additions can be made to your task library, they will not be recognized by TAF until TAF is reinitialized. There is no limit to the number of times existing tasks can be replaced.

The user name from which LIBTASK is being run must be the same as the user number in the xxJ file associated with the library being modified, and TAF must be able to attach the library in write mode in order for the update of TAF's copy of the TLD to take place. You must be validated for intercontrol point communication (the CSTP bit in user's access word must be set) to use the TT option. Each time you use the TT parameter with the LIBTASK command, a file is added to your task library.

Z Specifies that input directives are on the LIBTASK command. This option overrides the I option.

Part 1. Task listing. Generated if LO=A or LO=T.

TASK NAME	PRIORITY BEG MAX	QL	FIELD LENGTH	EXTRA FL	ATTR DCEORQ
task ₁	bp mp	ql	fl	efl	attrib
task ₂	bp mp	ql	fl	efl	attrib
.					
task _n	bp mp	ql	fl	efl	attrib

task₁ Task name.
 bp Beginning priority.
 mp Maximum priority.
 ql Queue limit.
 fl Field length required by the task.
 efl Extra field length the task may request.
 attrib Task attribute list. The attribute is in effect if the letter appears.

- D Destructive code.
- C Central memory-resident.
- E Extended memory-resident.
- O Task is OFF.
- R Solicited communication block.
- Q Queueing enforced.

Refer to Task Input Directives,
 later in this section, for information about these attributes.

Part 2. Transaction listing. Generated if LO=A or LO=X.

TRANSACTION UNIT	1ST	TASKS 2ND	3RD	4TH	5TH	D.M. CD	RECOVER
tname ₁	task ₁	task ₂	task ₃	task ₄	task ₅	datamanager	rc
tname ₂	task ₁	task ₂	task ₃	task ₄	task ₅	datamanager	rc
tname ₃	task ₁	task ₂	task ₃	task ₄	task ₅	datamanager	rc

tname₁ Transaction name.
 task₁ Task names. If any task does not exist in this library, its name is enclosed in parentheses.
 data-manager The data managers used by the named-transaction. May be either or both:
 C Transaction uses CRM.
 D Transaction uses CDCS.
 Use of the indicated data manager is allowed if the letter appears.
 rc Recovery indicator. For recoverable transactions, RC appears in this position.

Figure 10-1. Task Library Report

INPUT DIRECTIVES

NAMED-TRANSACTION INPUT DIRECTIVE

The named-transaction input directive is used to create and delete named-transactions.

The format is:

```
/tname,T1=task1,T2=task2,T3=task3,T4=task4,  
T5=task5,RC,DM=datamanager,DL.
```

tname	Transaction name. Must be one to seven alphanumeric characters.
taski	Name of the task(s) in the named-transaction, tname. The tasks will be executed in the order listed. If a task issues a CALLTSK with CEASE, the remaining tasks in the original transaction will not be executed.
RC	Recovery indicator. When this parameter is specified, this transaction is recoverable. If not specified, this transaction is not recoverable.
datamanager	The data manager(s) that may be used by this named-transaction. If a recoverable transaction attempts to use CDCS or CRM and its usage was not declared via this parameter, the run unit will be aborted. This parameter is valid only if the RC parameter is also specified. The options are: C The transaction uses CRM. D The transaction uses CDCS.
DL	Causes the deletion of transaction tname from the task library. If specified, this parameter must be the only parameter in addition to the tname parameter on the directive.

Additional tasks can be made part of the execution of a named-transaction if they are initiated via a CALLRTN or CALLTSK with CEASE. A CALLTSK with CEASE, however, will prevent any remaining tasks in the original transaction from being executed.

A task may be part of more than one named-transaction.

Declaring the named-transaction recoverable is necessary but not sufficient to ensure correct recovery from failures. The transaction must also be constructed to support recovery as specified in section 7 of this manual.

Tasks specified in this directive do not have to exist in the data base task library. This allows system tasks to be included in the named-transaction definition. Named-transactions with tasks that do not exist in the data base task library are enclosed by parentheses in the report generated by using the LIBTASK LO parameter.

Adding or deleting tasks within a named-transaction or altering other attributes is accomplished by respesifying the directive in full.

TASK INPUT DIRECTIVE

To alter task characteristics, directive statements may be entered in the directive file. Only one statement should appear for each task. These characteristics are kept in the directory for use by the transaction executive when the task is to be executed.

The directive statement format is:

```
*name P1,P2,...,Pn.
```

name is the task name and p_i is any of the following input directives in any order.

<u>Directive</u>	<u>Description</u>
BP=nn	Beginning scheduling priority of task.
BP	Beginning priority defaults to 25g.
BP omitted	Beginning priority defaults to 20g.
C	Specifies a memory-resident task. The default is a disk-resident task.
D	Specifies a destructive code (non-reusable) task that is removed from the subcontrol point when the task is completed. The default is a reusable task. This option is recommended when debugging a task.
DL	Specifies that a task (name) is logically deleted. All logically deleted tasks are physically deleted from the task library when the LIBTASK command contains the PR option.
E	Specifies an extended memory-resident library copy of the task. The default is mass storage-resident.

If tasks are specified as extended memory-resident, the transaction executive loads as many as will fit into its available extended memory during initialization and leaves the remaining tasks on the mass storage file. If no extended memory is available, all tasks are loaded from mass storage.

If a LIBTASK command specifying the TI option is issued while the transaction executive is running, no tasks are replaced or added to extended memory, and one of the following situations occurs.

- Old copy of task in extended memory and new copy in extended memory. The new copy is ignored, and the old copy remains on the library in extended memory.

Directive

Description

- Old copy not in extended memory and new copy in extended memory. The new copy is used but is read from mass storage.
- Old copy in extended memory and new copy not in extended memory. The new copy is loaded from mass storage.

EF=eFl Specifies the maximum amount by which a task will be allowed to increase its initial field length. efl is assumed to be octal unless it contains an 8 or 9 or is followed by a trailing D. The default value is zero. EF will be increased to the next multiple of 100g if it is not already such a multiple. The maximum field length allowed for the task is (initial task FL) + efl. The value of efl can range from 0 to 300000g.

IG Specifies that a task (name) in the file specified with the B option is ignored.

MP=mn Maximum scheduling priority of task (for future use in aging tasks).

MP Maximum priority defaults to 50g.

MP omitted Maximum priority defaults to 40g.

O Sets task to OFF; requests for the task are routed to the OFFTASK routine (refer to OFFTASK in section 11). The default is ON.

Q Specifies that queuing is to be forced, described as follows. This option alters the normal scheduling priority of the task (refer to Scheduling Priority) as follows:

1. When there are no active copies of the task, the first copy of the task is scheduled as usual (refer to Scheduling Priority).
2. While there are from one to six active copies of the task in TAF, another copy of the task is not scheduled until the number of transactions queued against its RTL entries equals or exceeds the queue limit specified by the QL directive. When an RTL entry is used to schedule a task, it is released and made available for requesting the scheduling of other tasks.
3. When the number of active copies of the task equals seven, additional copies of the task will not be scheduled. This applies even if the queue limit is 1.

Directive

Description

An active task is a copy of a task which is both:

- At a subcontrol point or rolled out.
- In the process of handling a transaction or has transactions queued against it (the copy).

If a LIBTASK function specifying the TT parameter is issued while the transaction executive is running and an attempt is made to change the Q attribute of a task on a library in use by TAF, then the Q attribute change will not affect TAF scheduling until the executive is reinitialized.

This parameter is not applicable to CM-resident tasks.

QL=n Specifies the number of requests, up to a maximum of five, that can be queued for an RTL entry for a task. If a task uses a large amount of memory and executes rapidly, performance is not degraded if requests are queued. In this case, multiple copies of the task require too much memory. If a task does a large amount of I/O, then performance is degraded if requests are queued. In this case, multiple copies of the task enhance performance.

QL Queue entry limit defaults to five.

QL omitted Queue entry limit defaults to three.

R Specifies that the FL assigned to a CM-resident task is to be reduced to the initial load value each time the transaction is run. The default is no reduce. Non-CM-resident tasks are always started with their initial load value.

SC Sets the task communication block load status to solicited. This means you must request the communication block and supply a valid address (using the BEGIN request) after the task is initiated. The default is unsolicited.

All directives for one task must appear on the same line. The following situation will result in only P1 being recognized as an input directive for TASK1.

```
*TASK1 P1.
*TASK1 P2.
.
.
.
*TASK1 Pn.
} Improper way to specify
  input directives.
```

SCHEDULING PRIORITY

When the tasks are queued, the scheduler will attempt to load the task with the highest current priority. If there are several tasks with the highest priority, the task with the lowest field length requirement will be selected for loading. When a task enters the queue, the beginning priority (BP) is used as the task's current priority. The current priority is then aged (incremented by one) every second up to the maximum priority (MP). Tasks in the queue having identical current priorities and field length requirements normally will be scheduled in the order they entered the queue.

With these directives, classes of tasks can be created. For example, if two classes were created (X and Y) their priorities could be:

```
XBP = 25g      XMP = 40g
YBP = 30g      YMP = 45g
```

In this case, no tasks in X could be scheduled until all tasks in Y having a current priority greater than 40g had been scheduled.

CREATING A NEW TASK LIBRARY

The following examples show how you can create a new task library. An explanation is provided after each LIBTASK command.

```
LIBTASK(P=0,N=lfn1,CR,B=lfn2,I)
```

LIBTASK copies file lf_{n2} to file lf_{n1} according to input directives in the COMPILE file and adds a new sorted task library directory at the end of file lf_{n1}.

```
LIBTASK(P=lfn2,N=lfn1,Z);*name
P1,P2,...,Pn.
```

LIBTASK copies file lf_{n2} to file lf_{n1} and adds a new sorted task library directory at the end of file lf_{n1} according to input directives on the LIBTASK command.

```
LIBTASK(P=lfn2,N=lfn1,PR,Z);*name
P1,P2,...,Pn.
```

LIBTASK copies all active (not logically deleted or replaced) task binaries from file lf_{n2} to file lf_{n1} and adds a new sorted task library directory at the end of file lf_{n1} according to input directives on the LIBTASK command.

```
LIBTASK(P=lfn2,N=lfn1,B=lfn3,I)
```

LIBTASK copies file lf_{n2} to file lf_{n1}, copies file lf_{n3} to file lf_{n1} according to input directives in the COMPILE file, and adds a new sorted task library directory at the end of file lf_{n1}.

```
LIBTASK(P=lfn2,N=lfn1,B=lfn3,PR,I)
```

LIBTASK copies all active (not logically deleted or replaced) task binaries from file lf_{n2} to file lf_{n1}, copies file lf_{n3} to file lf_{n1} according to input directives in the COMPILE file, and adds a new sorted task library directory at the end of file lf_{n1}.

UPDATING AN EXISTING TASK LIBRARY

The following examples show how you can update (add new tasks and delete and replace existing tasks) an existing task library without supplying any binaries. An explanation is provided after each LIBTASK command.

```
LIBTASK(P=lfn1,Z);*name P1,P2,...,Pn.
```

LIBTASK adds a new sorted task library directory at the end of file lf_{n1} according to input directives on the LIBTASK command.

```
LIBTASK(P=lfn1,PR,Z);*name P1,P2,...,Pn.
```

LIBTASK copies all active (not logically deleted or replaced) tasks from file lf_{n1} onto a scratch file, copies the scratch file back onto file lf_{n1}, and adds a new sorted task library directory to the end of file lf_{n1} according to the input directive on the LIBTASK command. This command is not valid when TAF is up and has attached file lf_{n1}.

```
LIBTASK(P=lfn1,B=lfn2,Z);*name
P1,P2,...,Pn.
```

LIBTASK copies file lf_{n2} to the end of file lf_{n1} according to input directives on the LIBTASK statement. LIBTASK adds a new sorted task library directory to the end of file lf_{n1}.

```
LIBTASK(P=lfn1,B=lfn2,PR,I=lfn3)
```

LIBTASK copies all active (not logically deleted or replaced) tasks from file lf_{n1} onto a scratch file, copies the scratch file back onto file lf_{n1}, copies file lf_{n2} to the end of file lf_{n1} according to input directives on file lf_{n3}, and adds a new sorted task library directory to the end of file lf_{n1}. This command is not valid when TAF is up and has attached file lf_{n1}.

DELETING A TASK FROM A TASK LIBRARY

With the DL input directive on the LIBTASK command, you can logically delete a task from the task library if TAF is up and has attached the task library as follows:

```
LIBTASK(P=lfn1,TT,Z);*name DL.
```

In this example, LIBTASK logically deletes a task (name) from file lf_{n1}. The TT option is required when TAF is up. The DL input directive is the only input directive allowed for the named task.

With the PR option on the LIBTASK command, you can delete inactive tasks from the task library when TAF is not up and when TAF is up but has not attached the task library.

```
LIBTASK(P=lfn1,PR,Z);*name DL.
```

In this example, LIBTASK physically deletes a task (name) from file lf_{n1}. TAF must not have the file attached when this command is executed.

IGNORING A TASK IN THE REPLACEMENT FILE

With the IG input directive on the LIBTASK command, the user can tell LIBTASK to ignore a task binary in the replacement file specified with the B option as follows:

```
LIBTASK(P=1fn1,B=1fn2,Z);*name IG.
```

In this example, LIBTASK skips a binary task (name) on file 1fn₂ when copying file 1fn₂ to file 1fn₁. The IG input directive is the only input directive allowed for the named task.

The applications programmer should be aware of the functions of these tasks but may omit the detailed information contained in this and section 12.

Several system tasks are required with every application: BTRAN recovery task (BTASK), data base recovery task (CTASK), initial task (ITASK), K display director (KDIS), logout transaction terminal (LOGT), message abort (MSABT), inactive task (OFFTASK), CDCS rerun task (RCTASK), run unit recovery task (RTASK), system message (SYSMSG), and execute named tasks (XTASK). Each must reside on the system task library of every transaction system. Limited versions of these system tasks are supplied by Control Data; however, modifications to these tasks are often necessary for specific applications. One additional system task, CRMTASK, is required if the TAF/CRM Data Manager is used. Refer to the TAF/CRM Data Manager 1 Reference Manual for information on CRMTASK.

Directions for installing these tasks are contained in the TAF Installation section of the NOS 2 Installation Handbook.

BTRAN RECOVERY TASK (BTASK)

BTASK recovers rerunnable BTRAN run units. BTASK must be installed on the TAF system library.

DATA BASE RECOVERY TASK (CTASK)

CTASK helps TAF/CRM recover the data base during TAF initialization, terminal failure, and NETWORK failure. CTASK must be installed on the TAF system library.

INITIAL TASK (ITASK)

The primary functions of ITASK are analyzing terminal input and TAF-originated transactions. If the TAF-originated transaction field is set in the communication block (figure 2-1), ITASK uses a table to initiate the processing routine associated with that type of transaction. Otherwise, ITASK examines three characters, starting with the first character of the input message. The first of these three characters is the transaction code; the other two comprise the subtransaction code entered from a terminal.

ITASK recognizes the transaction code EX., which is used to call the system transaction XTRAN. This special transaction code is reserved for XTRAN.

NOTE

Because ITASK schedules tasks, care must be taken to avoid entering an infinite loop when a task calls ITASK.

ITASK is described in more detail in section 12.

K DISPLAY TASK (KDIS)

KDIS is a system task scheduled by the transaction executive when the K.SWITCH command is entered by the console operator. KDIS requests the K display and displays a complete list of K display commands. The K display commands are discussed in the NOS 2 Analysis Handbook.

LOGOUT TASK (LOGT)

This task logs a terminal out of TAF. However, the task does not log the terminal off the computer system. The task sends a message, END OF TRANSACTION SESSION; performs a LOGT request; and performs a CEASE request. The terminal is then available for login to TAF or other network applications.

MESSAGE ABORT (MSABT)

Message abort (MSABT) is a system task that sends error messages to the originating terminal when a transaction is terminated due to a system-detected error. MSABT checks for error codes that were placed in the communication block by the transaction executive or the initial task. When an error code is detected, MSABT sends the appropriate error message and initiates an abnormal CEASE function.

MSABT is scheduled in the same manner as other tasks.

If the TOTAL Data Manager is used, MSABT must be modified to perform a FREEX command. This ensures that TOTAL records are released if a task aborts.

Error messages issued by MSABT are described in appendix B.

INACTIVE TASK (OFFTASK)

This task is provided to give you control over an internal condition that might be the result of operational error. OFFTASK is scheduled by the transaction executive when a request has been received for an inactive task. The task may be inactive or off due to the use of the K.OFFTASK command or use of a LIBTASK directive when the task was placed in the library. OFFTASK sends an error message to the originating terminal and ceases abnormally. It is expected that you will expand the task to handle any specific conditions that may exist when a transaction is interrupted by an inactive task.

CDCS RERUN TASK (RCTASK)

When CDCS becomes available after a failure, RCTASK is activated. RCTASK determines which terminals were effected by the CDCS failure and reruns their transactions (if rerunnable).

RUN UNIT RECOVERY TASK (RTASK)

The recovery task RTASK is a Control Data-supplied recovery task that will handle most transaction application needs; however, some sites may wish to modify RTASK to handle special requirements. The following discussion will aid understanding of RTASK in case a modification is required. RTASK must be available to all applications. This implies that it must be present on the TAF system task library (TASKLIB) or on all of the data base task libraries. It may be on the TASKLIB and any number of selected data base task libraries. In this case, only those applications without their own copy of RTASK will use the version on TASKLIB.

After terminal disconnects, or NAM, TAF, and system failures, the terminal must go through the login procedures with TAF. The following steps outline the decision process followed by RTASK. When a true condition is encountered, the remaining steps are skipped.

1. If the run unit is not recoverable because RC was not specified on the LIBTASK directive for the named-transaction, RTASK sends the message READY to the terminal.
2. If the run unit is not rerunnable because of an I/O error encountered by TAFREC, RTASK tells the terminal operator recovery has occurred and to enter the next command.
3. If the run unit is secure and the data base is committed, RTASK sends the terminal confirmation (SECURE) message.
4. RTASK tells the terminal operator that recovery has occurred, that the transaction is being rerun, and reruns the transaction.

SEND MESSAGE TASK (STASK)

This task sends a message to the terminal and then terminates. STASK is called from ITASK only when ITASK is to terminate after sending a message to the terminal.

SYSTEM MESSAGE TASK (SYSMSG)

This task sends K.MESSAGE-entered messages and K display diagnostic messages to terminals. It calls the subroutine in COMKCBT passing the user argument bits. The word returned is sent after the message is sent.

EXECUTE NAMED TASK (XTASK)

The execute named task (XTASK) is a special task scheduler used to execute most named tasks. It will not execute tasks by the names ITASK, KDIS, MSABT, OFFTASK, SYSMSG, and XTASK. These are the names of required system tasks. XTASK can be used to schedule LOGT as a means of logging a terminal out of transaction mode. XTASK is initiated via system transaction XTRAN, described following the XTASK description.

EX.task

EX. Special three-character transaction code which causes ITASK to schedule XTRAN via a CALLTRN request. XTRAN specifies XTASK as its only task.

task One- to seven-character alphanumeric task name, left-justified (no space between EX. and task) with a nonalphanumeric character terminator.

A site may want to delete the TRANS table entry of STRAN (X.), XTASK in the TRANT table of ITASK, or the system task XTASK altogether for security reasons. XTASK was provided as a means of scheduling tasks, and as an example of how a site might use such a task. However, it may provide too much flexibility, and therefore should be used with discretion.

XTRAN TRANSACTION

Transaction XTRAN must be defined when the system task library is created. XTRAN will be scheduled when the three-character code EX. is entered from a terminal. This transaction allows scheduling any task in the local or system task library as described under XTASK.

The characteristics of XTRAN are:

- Task name is XTASK.
- No data managers.
- Not recoverable.

This section describes some of the features, options, and requirements of ITASK.

MACROS

ITASK contains three tables. These tables are set up for the following codes.

- Time a task is to be run
- Transaction code
- Subtransaction code

A macro is used to set up each type of ITASK table. These three COMPASS macros, TIMCNT, TRAN, and STRAN, are discussed here.

TIMCNT MACRO

The TIMCNT macro makes an entry in the table of time-originating tasks (TTOT). TTOT contains the time a transaction or task is to be initiated and an address.

One TTOT contains all of the entries, regardless of the application that is to use them. This feature allows tasks to be run at predetermined times without specific task requests or terminal input to initiate the run.

The format is:

TIMCNT hr,min,sec,address

hr Hour.
 min Minute.
 sec Second.
 address Routine address.

The TTOT table format is:

Word 1 TTOT, number of entries in the table.

Word 2 and following Table entries.

The format of a TTOT entry is shown in figure 12-1.

TRAN MACRO

The TRAN macro sets up entries for the table of transaction codes (TRAN). TRAN contains the address of the table of subtransaction codes for that particular transaction code. ITASK assumes that the TRAN table is set up in 6-bit display code order. It uses the 6-bit display code value of the transaction code you enter to locate the proper entry in TRAN.

The format is:

TRAN code
 code One-character transaction code.

The format of an entry in TRAN is shown in figure 12-2.

STRAN MACRO

The STRAN macro makes entries in the table of subtransaction codes (STRAN). STRAN contains the subtransaction code and the corresponding transaction name.

The format is:

STRAN sc,tname
 sc Two-character subtransaction code.
 tname Name of the transaction to be called.

The format of an entry in STRAN is shown in figure 12-3.

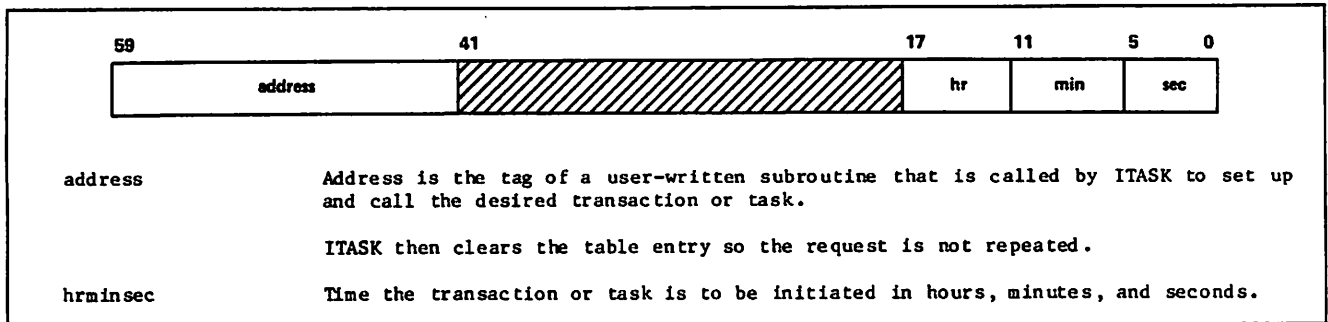


Figure 12-1. Format of a TTOT Table Entry

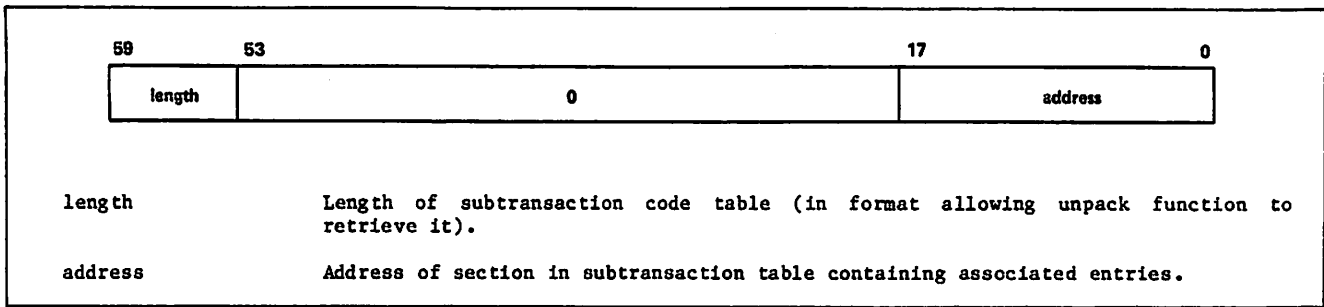


Figure 12-2. Format of a TRANT Table Entry

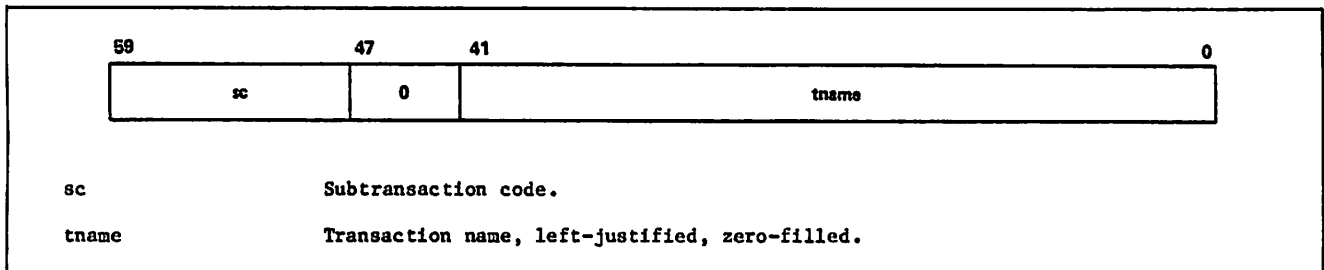


Figure 12-3. Format of a TRANS Table Entry

Each group of entries relating to the same transaction is placed immediately following the TRANT entry for that code, although the subtransaction table is assembled elsewhere. This is an example of the TRANT and TRANS tables.

```
TRANT BSS 0
      TRAN A
      STRAN 02,TRNAME
      STRAN 03,TRANN
```

A is the transaction code, 02 and 03 are the subtransaction codes, and TRNAME and TRANN are the respective transactions to be requested.

A site may want to delete the TRANS table entry of STRAN (X.), XTASK in ITASK for security reasons. XTASK is described in more detail in section 11.

ITASK DESCRIPTION

Entries in all ITASK tables, as well as the user subroutines for TIMCNT entries in TTOT, should be inserted in the designated places within the program before assembly. TRANT and TRANS entries must be in alphabetical or numerical order according to 6-bit display codes.

Initial terminal input, while TAF is running normally, results in execution of a transaction as follows:

- If the first three characters of input are recognized by ITASK as a transaction code that is associated with a transaction, then ITASK attempts to schedule the transaction. Transaction codes can be associated with transaction units through the STRAN macro, previously described.

- If no transaction was associated with the transaction code, then ITASK interprets the string of characters starting with the first character input and terminating with the first nonalphanumeric character as a transaction name. If this string is null or more than seven characters long, an error message is issued; otherwise, ITASK will attempt to schedule the named-transaction. If the named-transaction does not exist in the task library corresponding to the terminal application or in the system task library, an error message is issued via MSABT.

Initial terminal input while TAF is idling down (following an IDLE,TAF or K.IDLE command) is processed as follows:

- If the input is EX.LOGT, the logout is processed normally.
- If the input is anything else, ITASK issues the message TAF IDLING DOWN to the terminal and ceases without trying to schedule the requested transaction.

TTOT entries must be accompanied by user-written subroutines to set up and call the task, as follows:

- The system time is retrieved from the appropriate word in the communication block. TTOT is then searched for any entries that were to be run by that time.
- If such an entry is found, ITASK jumps to the user's subroutine to set up and call the task. This subroutine must set up the communication block required by the task that is to be run. It must perform a CALLTSK without the cease option set to ensure that control is returned to ITASK.

TAF-ORIGINATED TRANSACTIONS

TAF recognizes certain conditions not normally noted or processed by user applications. Some of these conditions are:

- Exceeding a length of time as specified by an assembly option (TACTL).
- K.IDLE command telling the transaction subsystem to idle down.
- Transaction executive recovery.
- Terminal login.
- Network and terminal failures.

When any of these conditions occurs, a TAF-originated transaction may be processed. Code in common deck COMKSTC identifies the condition. ITASK examines the communication block (figure 12-4) to determine the condition and jumps to a subroutine which sets up and calls the proper task or sequence of tasks to respond properly to the identified condition. Although the Control Data-supplied ITASK contains code to process each of the conditions, sites may want to modify ITASK and provide transactions to suit their individual needs.

To provide TAF-originated tasks that properly respond to the NAM conditions, the analyst should refer to the NAM/CCP 1 Reference Manual.

A number of common decks aid in processing the TAF-originated transactions as follows:

- COMKSTC gives the values for the condition field of the communication block.
- COMKNWF describes the fields for the NAM message word of the communication block.
- COMSNCD gives values to fields in the NAM message word.

CONDITIONS REQUIRING TAF-ORIGINATED TRANSACTIONS

The conditions that require TAF-originated transactions and possible methods of processing are described here. Limited versions of routines needed to process these transactions are available in ITASK. To change the existing routines, ITASK must be modified to include the necessary code or call the necessary tasks. It may be advisable to include short procedures as subroutines in ITASK, but long or frequently updated transactions should be maintained as separate programs apart from ITASK and merely called from the ITASK subroutine. The recovery situations described in this subsection apply to sites not using TAF automatic recovery.

Time Activation

This might be processed by ITASK as follows:

1. ITASK retrieves the system time from the appropriate word in the communication block.
2. ITASK then searches the TTOT for any transactions that are to be run at that time.
3. If such an entry is found, ITASK jumps to the user-supplied subroutine that must set up and call the next task for that transaction. This subroutine must set up the communication block required by the task that is to be run.

K.IDLE Console Operator Command

Same as IDLE,TAF command. Refer to ITASK Description earlier in this section.

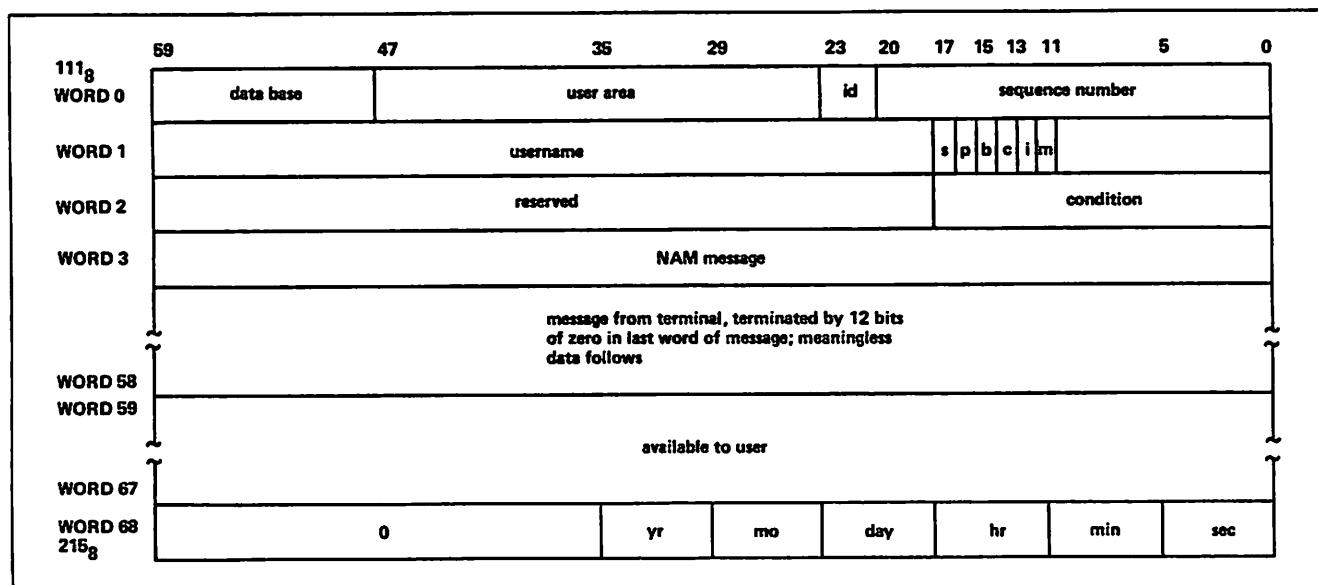


Figure 12-4. Communication Block for TAF-Originated Transactions

Terminal Login

The application may want to prompt the terminal user, set application character type, or go through recovery procedures if the user has the recovery field set in the user area of the terminal status table. Upon terminal login, the NAM message word of the communication block (figure 12-4) contains the following information:

<u>Bits</u>	<u>Significance</u>
35-25	Block size of terminal.
24	Signifies a hardwired terminal if set.
23-16	Terminal class.
15-8	Page width of terminal.
7-0	Page length of terminal.

Terminal Break Conditions

If some tasks perform SENDs without the recall flag set, breaks can occur because of the following.

- User break keys.
- Output device not ready.
- Incorrectly formatted data.

For a user to resume processing after a terminal break, the following must occur.

1. Terminal communication with NAM must be reestablished. When the break occurs, NAM places the supervisory message associated with the break in the NAM message word of the communication block. To restart terminal communications, TAF must send a reset supervisory message to NAM. The individual application need not provide this function. Upon the user task SEND request, TAF issues the reset supervisory message prior to honoring the SEND.
2. The user must resume interaction with the application. The user application must provide this break recovery. The following description suggests one method.

ITASK and any recovery tasks can communicate through the terminal status table user area. For instance, when ITASK initiates a transaction, it could set a series of fields in the TST to indicate important events in the transaction. The recovery task could examine the fields to determine what events had occurred. Nonrecovery tasks could watch a recovery field in the TST and CEASE to allow the recovery task to proceed in a quiet environment. (TAF sets the recovery field in the TST before calling ITASK.)

Terminal Connection Broken

A connection may be broken because a line has been disconnected. If a SEND with recall was active, the task issuing the SEND rather than ITASK is notified of the error. It may be desirable for the application to process these incomplete transactions. TAF sets the recovery field in the terminal status table user area. The application can save terminal output on a data manager file and send it when the user logs in again. The outstanding block limit specified in the local configuration file determines how many messages must be saved. (Refer to the Network Products NAM 1 Network Definition Language Reference Manual.)

Network Normal Shutdown

TAF monitors task activity on communication blocks generated by terminal input. When all tasks have been completed and no tasks are performing WAITINP requests, TAF sends end connection supervisory messages to NAM for each logged-in user and performs a NETOFF to end processing with the network. To initiate this condition, it may be desirable for ITASK to notify users of the pending shutdown and begin rejecting new transactions.

Network Failure

TAF sets the recovery field in the user area of the terminal status table for all logged-in users. All tasks performing WAITINP requests are scheduled. The application, through ITASK, may want to begin data base recovery.

Network Immediate Shutdown

TAF performs a NETOFF and sets the recovery field in the user area of the terminal status table. The application may want to begin recovery by processing only transactions from batch input.

Terminal Inactive

NAM issues a terminal inactive supervisory message when it detects that 10 minutes have elapsed without message traffic over a connection. TAF passes the transaction sequence number of any running task using that user name to ITASK. The application may want to log out the user after a period of time to reduce line connect costs.

Logical Error

One of the application's tasks may have set up an application block header or message block incorrectly. If the task performed a SEND request with recall set, the task, rather than ITASK, is notified of the error.

Block Not Delivered

This occurs because of a system error. If the task performed a SEND request with recall set, the task, rather than ITASK, is notified of the error. Before scheduling ITASK, TAF sets the recovery field in the user area of the TST. It may be desirable for the application to recover by sending the lost block. The application block number of the block that was lost is returned in the NAM message word of the communication block. Tasks may specify a block number on the SEND or obtain the block number used by TAF. Resending the lost block may cause the block to be delivered out of sequence, if further blocks had been sent following the lost block.

Terminal Characteristic Change

A NAM terminal characteristic supervisory message is sent to ITASK when the terminal operator has redefined the terminal class, page width, or page length.

Terminal Input Too Large

The terminal operator sent a message longer than the maximum input message size. TAF will not have journalized the input.

TERMINAL TYPE AHEAD

The network allows terminal operator type-ahead. In other words, the terminal operator can enter several transactions without waiting for the response from the first. The network ignores all type-ahead when the terminal operator enters a user break 1 or user break 2.

USER STATUS

The network allows a terminal user to make a status request by typing:

ct (alpha) $\text{\textcircled{R}}$

ct The network control character.

(alpha) Any alphabetic character.

TAF will right justify the character in word 3 of the communication block and initiate ITASK. At this point a user-written routine in ITASK must take over to perform whatever action is desired. The ITASK reason code will indicate a user status message.

1. The first part of the document discusses the importance of maintaining accurate records of all transactions. It emphasizes that this is essential for the proper management of the organization's finances and for ensuring compliance with applicable laws and regulations.

2. The second part of the document outlines the specific procedures that should be followed when recording transactions. This includes details on how to handle receipts, invoices, and other financial documents, as well as the frequency and timing of record-keeping activities.

3. The third part of the document addresses the issue of data security and confidentiality. It provides guidance on how to protect sensitive financial information from unauthorized access, theft, or loss, and discusses the importance of implementing robust security measures.

4. The fourth part of the document discusses the role of technology in record-keeping. It explores the benefits of using accounting software and other digital tools to streamline the record-keeping process and improve the accuracy and efficiency of financial reporting.

5. The fifth and final part of the document provides a summary of the key points discussed throughout the document. It reiterates the importance of maintaining accurate records and offers final thoughts on how to ensure the success of the organization's financial management practices.

6. The sixth part of the document discusses the importance of regular audits and reviews of the record-keeping process. It explains how these activities can help identify any errors or discrepancies in the records and ensure that the organization's financial reporting remains accurate and reliable.

7. The seventh part of the document addresses the issue of training and education for staff involved in record-keeping. It emphasizes the need for ongoing training and development to ensure that all employees are up-to-date on the latest record-keeping practices and technologies.

8. The eighth part of the document discusses the importance of maintaining clear communication and collaboration between all departments involved in the record-keeping process. It highlights the need for regular meetings and updates to ensure that everyone is on the same page and working towards the same goals.

9. The ninth part of the document provides a list of resources and references for further information on record-keeping. This includes books, articles, and online resources that can provide additional insights and guidance on this important financial management topic.

10. The tenth and final part of the document offers a concluding statement on the importance of record-keeping for the success of any organization. It encourages all readers to take the time to implement the best practices discussed throughout the document and to continuously improve their record-keeping processes over time.

CHARACTER SETS

A

Table A-1 lists the character codes for the ASCII 63/64-character sets. Users should check with a site analyst to see which character set is being used.

Table A-1. 7-Bit ASCII Code Standard Print 63/64-Character Set (Sheet 1 of 2)

ASCII Character	7-Bit ASCII Code		6-Bit Display Code (Octal)
	Octal	Hexadecimal	
:	072	3A	00†
A	101	41	01
B	102	42	02
C	103	43	03
D	104	44	04
E	105	45	05
F	106	46	06
G	107	47	07
H	110	48	10
I	111	49	11
J	112	4A	12
K	113	4B	13
L	114	4C	14
M	115	4D	15
N	116	4E	16
O	117	4F	17
P	120	50	20
Q	121	51	21
R	122	52	22
S	123	53	23
T	124	54	24
U	125	55	25
V	126	56	26
W	127	57	27
X	130	58	30
Y	131	59	31
Z	132	5A	32
0	060	30	33
1	061	31	34
2	062	32	35
3	063	33	36
4	064	34	37
5	065	35	40
6	066	36	41
7	067	37	42
8	070	38	43
9	071	39	44
+	053	2B	45
-	055	2D	46
*	052	2A	47
/	057	2F	50
(050	28	51
)	051	29	52
\$	044	24	53
=	075	3D	54

† In the 63/64-character set, this 6-bit display code represents a null character. Also, use of the colon in program and data files may cause problems. This is particularly true when it is used in print and format statements.

Table A-1. 7-Bit ASCII Code Standard Print 63/64-Character Set (Sheet 2 of 2)

ASCII Character	7-Bit ASCII Code		6-Bit Display Code (Octal)
	Octal	Hexadecimal	
(space)	040	20	55
,	054	2C	56
.	056	2E	57
#	043	23	60
[133	5B	61
]	135	5D	62
z	045	25	63†
"	042	22	64
-	137††	5F	65
!	041	21	66
&	046	26	67
'	047	27	70
?	077	3F	71
<	074	3C	72
>	076	3E	73
@	100	40	74
\	134	5C	75
^	136	5E	76
;	073	3B	77

†In the 63/64-character set, this 6-bit display code represents a colon (:), 7-bit ASCII code 072, 7-bit hexadecimal code 3A.
 ††On TTY models having no underline, the back arrow (←) takes its place.

DIAGNOSTIC MESSAGES

B

This appendix contains an alphabetical listing of messages which may be important to the TAF programmer. Each message is followed by an explanation of the message and/or the circumstances causing it to be issued, the recommended action, and the routine which issued the message.

Lowercase letters are used within a message to identify variable fields. All messages beginning with lowercase (variable) fields are listed alphabetically according to the first nonvariable field after the messages that begin with nonvariable fields.

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
cmd ACCEPTED..	The network has accepted your terminal definition command cmd.	None.	CCP
AFTER IMAGE ACCUMULATOR TABLE OVERFLOW.	When updating a file, the after image accumulation table was filled.	The size of the AAIT table in DMPEC must be increased. Inform the data administrator.	DMREC (AAI)
AFTER IMAGE RECOVERY FILE IS FULL.	The active after image recovery file used by TAF is full.	This file must be dumped by DMREC.	TAF
db AFTER IMAGE RECOVERY FILES UNAVAILABLE.	TAF/CRM data manager tried to bring up an after image recovery file for data base db but failed. Possible reasons for the failure: - ATTACH error - ARF header name is incorrect - ARF header status error - CIO error on ARF - Change in ARF block length - ARF full If the data base was down previously, it will remain down. If not down, it will now be idling down.	Correct problem with ARFs.	AAMI
AIP DEBUG OPTION TURNED OFF.	Self-explanatory.	None.	TAF
AIP DEBUG OPTION TURNED ON.	Informative message. All network messages will be placed on trace file ZZZZDN.	None.	TAF
AIP TOO LARGE FOR LOADING.	A fatal error occurred causing TAF to abort.	Inform site analyst. TWFA must be increased in deck COMKTAF.	TAF
filename ALLOCATED.	Informative message.	None.	DMREC
APPLICATION ERROR - NO TERMINAL OUTPUT.	A task error. A transaction has run without either an error message or a SEND being sent to the originating terminal.	Correct task or inform data base administrator.	MSABT
APPLICATION FAILED - applnam	An application has stopped without making a request to NAM to terminate.	Restart application and/or contact central software support. analyst.	NVF
APPLICATION NETTED OFF - applnam	Application applnam has requested termination of network access.	None.	NVF
APPLICATION NETTED ON - applnam	An application has successfully requested and attained network access.	None.	NVF
ARF BLOCK SIZE IS LARGER THAN BUFFER.	The maximum block length field in the header record for an ARF is larger than the maximum block length (installation parameter) allocated by TAF/CRM data manager.	The file must be dumped using DMREC and preallocated.	TAF
ARF DUMP TAPE HEADER ERROR.	No header found on ARF tape.	Use alternate tape if available.	DMREC
ARF ENTRY TABLE OVERFLOW	Too many recoverable file names exist on this ARF.	Increase the size of the TLOG table with installation parameter TLOGL.	AAMI
ARF FILE HEADER ERROR.	No header was found on the ARF file.	Down the data base, dump the data base, purge the old ARF, and create a new one.	DMREC
ARF HEADER ERROR.	No ARF header on what is supposed to be an ARF.	Check ARF for validity, inform data administrator.	DMREC
ARF HEADER STATUS (3) POSSIBLE ERROR.	ARF header status shows a possible error condition.	Inform data administrator.	DMREC
ARF ORDINAL MUST BE 01 OR 02.	ARFs can have only 01 or 02 as an ordinal.	Correct directive name (ARF/BRF).	DMREC
ARITHMETIC OR OTHER SOFTWARE ERROR IN TASK.	A task error. Hardware or the system detected a software error in the task.	Correct task or inform data base administrator.	MSABT
ATTACH ERROR ON - filename.	The transaction executive cannot attach the file filename under present conditions. This usually implies that the file does not exist or modify permission has not been given to the TAF user name.	Correct error and reinitialize executive, or inform the data administrator.	TAF

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
ATTACH ERROR ON PF xxxxxxx.	When attempting to attach a permanent file xxxxxxx, an error was encountered.	Check for valid directive file name and presence of a direct access permanent file by that name.	DMREC
ATTACH MODE MUST BE W, M, R, OR RM.	The mode parameter on the CYBER Record Manager (CRM) statement must be one of the specified values.	Correct the mode parameter on the CRM statement or inform the data administrator.	TAF
n.nnn AVERAGE ACTIVE SUBCONTROL POINTS.	n.nnn is the average number of simultaneously active subcontrol points when TAF is not rolled out. An active subcontrol point is one which is in recall, is waiting to use the CPU, or is currently assigned the CPU. The sampling rate is once per second.	None.	TAF
n.nnn AVERAGE OUTSTANDING COCS REQUESTS.	n.nnn is the average number of outstanding (uncompleted) system control requests per second. The sampling rate is once per second.	None.	TAF
BACKUP DIRECTORY - xxxxxxx HAS BEEN RECONSTRUCTED.	No directory file for the specified data base. A new directory was generated from the information on the xxj file.	Inform data administrator.	DMREC
BATCH CONCURRENCY DISABLED.	TAF was brought up without BATCH concurrency enabled. (TBCON=Q).	Ask data administrator to bring up TAF with BATCH concurrency ENABLED.	BAAML
BATCH RECOVERY ACTIVE ON DATA BASE	Self-explanatory	Wait for the completion of batch recovery, then reinitialize the transaction subsystem.	TAF1
BEGIN CRM TASK RECOVERY.	TAF/CRM has received a recovery indication from TAF.	None.	AAMI
BFL = xxxxxxxB, MINIMUM REQUIRED = zzzzzzB.	Information on the actual value of the beginning field length and the minimum value TAF/CRM requires for the Common Memory Manager Buffer.	None.	AAMI
BINARY RECORD WAS LESS THAN 64D WORDS LONG.	The task binary file contains records in the wrong format.	Ensure that a LOAD was performed after compilation of the tasks.	LIBTASK
BINARY RECORD WAS NEITHER ABSOLUTE NOR OVERLAY FORMAT.	The task binary file contains records in the wrong format.	Ensure that a LOAD was performed after compilation of the tasks.	LIBTASK
BLOCK BUFFER OVERFLOW.	When attempting a record load operation, the data block buffer was found to be too small.	Increase the length of the working buffer WBUF. Inform the data administrator.	DMREC
BLOCK BUFFER TOO SMALL.	Tape block on ARF was too large for buffer.	Increase size of buffer and try again.	DMREC
BLOCK LOAD ERROR.	Block load of a file failed because no header was found on dump tape. Or, the targeted file or the tape does not match request file name.	Load from previous dump tape.	DMREC
BLOCKAGE AMONG CM RESIDENT TASKS.	The sum of initial field lengths for the CM resident tasks exceeds the minimum size of total task area.	Correct error.	TAF
BOTH AFTER IMAGE FILES ACTIVE.	Both after image recovery files for a data base are active. The after image files must be dumped by DMREC, if not already dumped.	Dump the after image recovery files.	TAF
BRF PARAMETER TOO LARGE OR INVALID.	The parameter on the BRF statement is nonnumeric, the value is greater than the installation parameter BMAX, or too many parameters were specified.	Correct BRF statement or reassemble TAF and related decks with a larger value for BMAX (if possible).	TAF
BTASK - TAF AUTOMATIC RECOVERY NOT INSTALLED.	A keyword specified in a TSTAT request requires TAF automatic recovery to be installed.	Inform the data administrator.	BTASK
BTASK - TASK LOGICAL ERROR.	An unexpected error status was returned.	Inform the data administrator.	BTASK

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
BTASK - xxxxxxx TRANSACTION NOT RUNNABLE NOW.	This occurs when an I/O error is detected and the data base cannot be recovered for the user xxxxxxx.	Inform the data administrator.	BTASK
BTASK - USER NOT DEFINED IN NETWORK FILE.	The user is not defined in the NETWORK file on a TSTAT request.	Inform the data administrator.	BTASK
C. B. NOT AVAILABLE TO SCHEDULE CTASK.	Communication block not available to schedule CTASK.	Increase the number of communication blocks at initialization time using the K.MCB= command.	TAF
CALLRTN NESTED CALL LIMIT EXCEEDED.	A task error. An attempt was made to nest CALLRTN requests more than 16 levels.	Correct task or inform data base administrator.	MSABT
CANNOT DEFINE REFORMATTED FILE - filename.	A reformat of the TAF COMMUNICATIONS recovery file was necessary, but when attempting to DEFINE the reformatted file (ZZCRFA1), an error occurred.	Purge the existing file indicated by filename.	TAFREC
CDCS ABORTED PLEASE WAIT.	A TAF transaction was using CDCS while CDCS aborted. The transaction will be rerun after CDCS comes back up.	Operator can log off by typing in EX.LOGT, or wait for CDCS.	ITASK
CDCS CONNECTION ESTABLISHED.	Informative message. The connection between TAF and CDCS has been established.	None.	TAF
CDCS INVOKE FAILURE.	CDCS files are not available.	Inform site analyst or data administrator. The transaction should be rerun after the problem is fixed.	MSABT
CDCS NOT AVAILABLE.	Your TAF transaction attempted to use CDCS when it was not present in the system, CDCS aborted while your transaction was waiting for completion of a CDCS request (SSC), or CDCS aborted while your transaction was rolled out.	Inform data administrator.	MSABT
CDCS REQUESTED ABORT.	TAF aborted the TAF transaction as a result of a request by CDCS. The detailed error message returned by CDCS is available in the JOURD file and will be sent to the terminal.	Inform data administrator.	MSABT
CHANGED TLD DETECTED - filename, username.	An unrecognizable library directory format was encountered during a library directory update attempt.	Inform site analyst.	TAF
CIO ERROR ON RECOVERY FILE	A CIO error was encountered while trying to position the ARF or BRP correctly.	Reinitialize the transaction subsystem or inform the site analyst.	TAF1
CMM ERROR.	A *CMM* error occurred. Memory cannot be granted to load *OPEN* routine.	Inform data base administrator.	AAMI
CMMBFL IS NOT LARGE ENOUGH.	The size of the beginning field length (BFL) for the Common Memory Manager Buffer is not large enough to contain CMNTFL and CMMCAP and the user's FIT-S and FSTT-S.	Data base administrator can either 1) increase the size of CMMBFL, or 2) decrease the size of CMNTFL, or 3) decrease the size of CMMCAP.	TAF
CONFLICTING LIBTASK OPTIONS - p1,p2.	The user specified an incorrect combination of LIBTASK command options. Possible incorrect combinations of p1 and p2 are: - CR,PR - CR,TT - PR,TT - TT,N Another possible incorrect combination gives P=0,-N for p1,p2. This means that the user specified P=0 but did not specify N. User specified TT option when task library was not attached by TAF.	The user must choose either p1 or p2. For the P=0,-N combination, the user must specify N if no old task library (P=0) exists.	LIBTASK
CONTROL CARD ARGUMENT ERRORS.	Errors exist in the parameters on the LIBTASK command.	Correct and rerun.	LIBTASK
CRF HEADER ERROR - filename.	I/O errors or logical errors were encountered in the header record of the named CRF. The file is unusable. filename File name	Initialize the file using the K.INT initial K display command.	TAFREC

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
CRF RECOVERY/INITIALIZATION ABORTED.	Informative message.	The action depends on the message preceding this one in the dayfile.	TAFREC
CRF RECOVERY/INITIALIZATION COMPLETE.	Informative message.	None.	TAFREC
CRF RECOVERY UNIT ERROR - filename.	I/O errors or logical errors were encountered in a run unit of the named CRF.	Attempt to recover any good information using the K.ERO=YES initial K display command, or initialize the file using the K.INT initial K display command.	TAFREC TAF
CRM(...parameter-List...)	This is a copy of a CRM statement that is in error. A subsequent message follows.	Inform the data administrator.	TAF
CRM DATA MANAGER NOT LOADED.	An AAM request for CRM was issued by a task and the CRM Data Manager was not loaded into TAF at initialization time.	Check the DMS statements in the TCF which specify which data managers are to be loaded by TAF. For CRM users, you need a DMS statement such as: DMS,CRM,ON,xx,xx,....,xx . . . where xx = any data base identifier.	MSABT
CRM DATA MANAGER SUCCESSFULLY LOADED.	Self-explanatory.	None.	TAF
CRM ERROR nnnB IGNORED ON OPEN OF filenam.	A non-fatal CRM error, nnnB, was returned to the TAF/CRM data manager while issuing CRM requests associated with opening a recoverable CRM file, filenam, for the first time. This error is ignored.	Informative.	AAMI
CRM ERROR ENCOUNTERED.	A CRM error occurred while processing the directory. The CRM error is either an incorrect key or end of file when this is the only output message.	Inform data administrator.	DMREC
CRM ERROR xxx IN UPDATE PROCESSOR.	A CRM error has occurred during the application of the after image to a data file.	Note which error has occurred and locate references to the error in the CRM/AAM Reference Manual for appropriate action.	DMREC TAF
CRM ERROR IN ZZdbDIR (GET).	A CRM error was encountered when trying to retrieve a VSN record from the directory on a file load.	Inform data administrator.	DMREC
CRM TASK RECOVERY IMPOSSIBLE.	Recovery is not possible when the recovery file structure is found to be inconsistent with TAF/CRM tables and parameters specified on xxJ files.	The last reported BRF must be corrected or reallocated.	AAMI
CRMTASK - DATA BASE DOWNED BY OPERATOR.	DMREC notified TAF that the data base was recovered. But AAMI was not able to access it because it had been downed by the operator.	Only an operator may up the data base that he/she downed. It cannot be upped automatically.	CRMTASK
CRMTASK - DATA BASE/FILE BEING IDLED.	AAMI will not process any requests on an idle data base/file.	After the cause of the data base/file being idle has been fixed, the operator must bring it up.	CRMTASK
CRMTASK - DATA BASE/FILE CANNOT BE UPPED.	AAMI was not able to bring up the data base or file.	Data administrator must investigate the cause of AAMI inability to bring file/data base up.	CRMTASK
CRMTASK - DATA BASE/FILE NAME UNKNOWN.	DMREC request had an incorrect data base or file name.	Call the data administrator to investigate the cause.	CRMTASK
CRMTASK - INCORRECT REQUEST.	CRMTASK was not called correctly by DMREC.	There may be a transaction in the system attempting to use functions reserved for DMREC.	CRMTASK
CRMTASK - INCORRECT REQUEST FORMAT.	AAMI rejected this request.	An analyst should investigate the reason for this rejection.	CRMTASK

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
CRMTASK - INCORRECT TT SEQUENCE NUMBER.	AAMI does not recognize the TT sequence number returned by DMREC.	Call the analyst to investigate the discrepancy between DMREC and TAF.	CRMTASK
CRMTASK - NOT ALL DATA BASE FILES UPPED.	DMREC recovered the data base but AAMI was unable to attach/open all of data base files.	Data administrator should investigate why only some of the data base files came up.	CRMTASK
CRMTASK TIMEOUT.	A terminal job timed out while waiting for the next command.	A valid command must be entered within the default time of 480 seconds.	CRMTASK
CRT - CIO ERROR.	A CIO error was generated when creating a log file.	Purge the log file and try again (check file name).	DMREC
CSM - INCORRECT COMMUNICATION FUNCTION.	An incorrect or unrecognizable request was received by the transaction executive from the CPU monitor.	Inform site analyst.	TAF
CTASK - DATA BASE OR FILE DOWN.	The data base, TAF/CRM, or file is down on an RSTDBI request.	Inform the data administrator.	CTASK
CTASK - DATA BASE OR FILE IDLE.	The data base, TAF/CRM, or file is idle on an RSTDBI request.	Inform the data administrator.	CTASK
CTASK - FILE IS NOT INSTALLED.	The data base or file is not available in the xxJ file on a CRMSTAT request.	Inform the data administrator.	CTASK
CTASK - INCORRECT RECOVERY CASE.	The recovery case selected for processing was incorrect.	Inform the data administrator.	CTASK
CTASK - NO CRM RECOVERY FILES FOR DATA BASE.	There is no recovery file assigned to the data base on an RSTDBI request.	Inform the data administrator.	CTASK
CTASK SYSTEM IDENTIFIER UNKNOWN.	The old system identifier is unknown when issuing a TINVOKE request.	Inform the data administrator.	CTASK
CTASK - TABLE AREA NOT LARGE ENOUGH.	Take table area supplied by CTASK for a CRMSTAT request is not large enough.	Inform the data administrator.	CTASK
CTASK - TASK LOGICAL ERROR.	An unexpected error status was returned.	Inform the data administrator.	CTASK
CTASK - TRANSACTION NOT RERUNABLE.	The TAF transaction was not rerunnable and a RERUN was attempted.	Inform the data administrator.	CTASK
CTASK - TRMREC ERROR.	One of the following occurred: - There was no outstanding DBEGIN request. - An error was encountered on a data base or recovery file. - The data base or TAF/CRM is down.	Inform the data administrator.	CTASK
CTASK - USER NOT DEFINED IN NETWORK FILE.	The user specified on a TSTAT, WSTAT, or RERUN request is not defined in the NETWORK file.	Inform the data administrator.	CTASK
CYCLE NUMBER IS OUTSIDE LIMITS.	The specified cycle number is outside the limits set by the installation parameter (CYUCM).	Check the maximum cycle number and rerun.	DMREC (CYC)
DATA BASE/FILE ALREADY DOWN OR IDLE.	CRMTASK issued a DBDOWN request to AAMI, but the data base or file name was already down or idle.	Inform data base administrator.	CRMTASK
DATA BASE/FILE CANNOT BE UPPED.	CRMTASK issued a DBUP request but AAMI was not able to bring up the data base or file name.	Inform data base administrator.	CRMTASK
DATA BASE/FILE NAME UNKNOWN.	CRMTASK - issued a CRMSTAT, DBUP, or DBDOWN request but the data base or file name was not found.	Inform data base administrator.	CRMTASK
DATA BASE NAME IN CRM FILE NAME DOES NOT MATCH xxJ.	The two-character data base name from the file name on the CRM statement does not match the xxJ header.	Correct the CRM statement and try again.	DMREC
DATA BASE NAME OR FILE NAME MISSING.	No file name or data base name on directive.	Include file name or data base name on directive.	DMREC
DATA BASE NAME OR FILE NAME(S) BOTH SPECIFIED.	Self-explanatory.	Eliminate file name or data base name from the directive.	DMREC

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
DATA BASE NOT IN EDT	Self-explanatory.	Reinitialize the transaction executive, or inform the site analyst.	TAF1
DATA BLOCK BUFFER OVERFLOW.	ARF tape block size is too large for buffer allocated for it.	Inform the data administrator and lengthen buffer WBUF.	DMREC
DATA MANAGER USAGE NOT SELECTED.	A recoverable TAF transaction tried to use either the CDCS or CRM Data Managers and the usage of the data manager was not selected on the LIBTASK input which declared the named-transaction to TAF.	Respecify the characteristics of the named-transaction via the LIBTASK directive, or design the transaction not to use CDCS or CRM Data Managers.	MSABT
DB - NAME MUST NOT START WITH THE LETTER Z.	A data base name was found which starts with the letter Z. These are reserved names.	Change the data base name to one not beginning with Z. If the name appears on an NCTF entry, contact the user and request a new identifier. If the name appears on the TCF, make the change on the applicable DMS statement or inform the data administrator.	TAFREC
DEFINE ERROR ON FILE.	An error in defining PRU size or in trying to define the log file was encountered on a create directive.	Correct PRU length on create directive.	DMREC
DEL	The system has deleted the last line of input. This occurred when the terminal operator entered the cancel character (CN) followed by the end of line/end of block terminator.	None.	CCP
DELIMITER WAS NOT RECOGNIZED.	An incorrect delimiter or an unrecognizable delimiter was encountered on a directive.	Check directive format and rerun.	DMREC
DIRECTIVE CONTAINS AN INCORRECT DATE/TIME.	A directive contains an unrecognizable date/time.	Correct the directive and rerun.	DMREC
DIRECTIVE FORMAT ERROR.	Error in one or more directive parameter formats.	Correct directive and rerun.	DMREC
DIRECTIVE KEYWORD NOT VALID.	The wrong delimiter on the directive parameter was used, or the directive keyword is not valid.	Correct directive and rerun.	DMREC (SPR)
DIRECTIVE NOT PRECEDED BY EDIT DIRECTIVE.	This directive must be preceded by an edit directive.	Include an edit directive.	DMREC
DIRECTORY HEADER FROM THE COPY.	Informative message.	None.	DMREC
DIRECTORY IS FULL-NO MORE ADDITIONS ALLOWED.	The maximum length of the task library directory has been reached.	LIBTASK must be reassembled to allow for a larger directory.	LIBTASK
DIRECTORY UNUSABLE.	Attempt to reconstruct the directory failed.	Check the output for the detailed error message.	DMREC
DISPLAY DUMP NOT ALLOWED TO TERMINAL.	A 6-bit display code dump to a terminal is not allowed.	Specify the 0 option on the KTSDMP command if a dump to a terminal is desired.	KTSDMP
DMREC COMPLETE.	Informative message. The output file may contain other informative messages and should be reviewed.	None.	DMREC
DMREC FAILED - xxxxxxx zz.	The DMREC job submitted by TAF failed. xxxxxxx is the directive being processed, and zz is the data base name.	Notify the data base administrator and correct as directed.	DMREC
DMREC TAPE LABEL ERROR.	No tape header was found on ARF to be used for an update function.	Check for correct ARF tape. Use alternate ARF tape if available.	DMREC (UPD)
DOWNED BRF TABLE OVERFLOW.	Table of downed BRFs has overflowed.	Inform data administrator.	DMREC (BRT)

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
DSP ERROR ec RETURNED ON JOB ROUTE.	DSP has informed the transaction subsystem that an error was detected in routing a job for task dump processing. ec is the DSP error code.	Inform site analyst.	TAF
DSP ERROR ec RETURNED ON TASK SUBMIT.	DSP has informed the transaction subsystem that an error other than a job or user statement error was detected. ec is the DSP error code.	Inform site analyst.	TAF
DUMP TAPE SPECIFIES NON-STANDARD COMPRESSION.	When trying to reload a record formatted dump tape, the compression mode was nonstandard.	Try loading from a previous dump tape and inform the data administrator.	DMREC
DUMP WILL START OVER	DMREC attempted to execute the DUMP directive but encountered errors.	None.	DMREC
cmd DUPLICATE CHARACTER	The value you specified in the terminal definition command cmd conflicts with another terminal definition.	Try another value.	CCP
DUPLICATE DATA BASE IN TCF - xx.	Active data base identifier, xx, in the TCF is not unique.	Fix TCF so that xx appears only once among active (ON) DMS statements.	TAF
DUPLICATE DUMP ENTRY ON ADD.	When trying to add a file dump entry to the directory with an add directive, a duplicate entry was found.	List the directory for visual check and try again.	DMREC
DUPLICATE ENTRY ON ADD.	When trying to modify the directory a duplicate of the entry was found.	List the directory and check for the needed entry.	DMREC
DUPLICATE PARAMETER.	A duplicate VSN or duplicate file name was detected on a single directive.	Correct directive and rerun.	DMREC
DUPLICATE VSN ENTRY ON ADD.	When trying to add a VSN entry to the directory with an add directive, a duplicate entry was found.	List the directory for visual check and try again.	DMREC
DXB CONVERSION ERROR ON TRANSACTION SEQUENCE NUMBER.	An error occurred while converting the number in the table entry to binary.	Inform data administrator.	DMREC
ECS READ ERROR.	Self-explanatory.	Inform customer engineer.	TAF
ECS TASK taskname NOW MS RESIDENT.	Task taskname could not be loaded into extended memory because of insufficient storage. It is loaded into mass storage.	If task must be resident in extended memory, more extended memory space must be allocated for the TAF user name. Refer to the NOS 2 Installation Handbook.	TAF
ECS WRITE PARITY ERROR ENCOUNTERED.	Self-explanatory.	Inform customer engineer.	TAF
EDITING COMPLETE.	Informative message.	None.	DMREC
END OF FILE REACHED.	Informative message.	None.	DMREC
END OF VSN TABLE.	DMREC has reached the end of the VSN table on a LOAD or RECOVER process.	Check the dayfile for possible tape errors which may cause this condition.	DMREC
ENTER NEXT TRANSACTION.	Option C was selected to allow the entry of the next transaction. The previous recoverable information is lost.	Enter next transaction.	RTASK
ERROR IN ACCOUNT/USER CARD ARGUMENT.	The charge or user statement in the xxJ file is in error.	Correct the charge/user statement and try again.	DMREC
ERROR IN ADD/DELETE VSN.	The number of added or deleted entries does not match the number of active files.	Check edit directives and list the directory for a visual check.	DMREC
ERROR IN BRJ PARAMETER IN xxJ FILE.	The parameter on the BRJ statement is non-numeric, or the value exceeds the installation parameter BMAX.	Correct value parameter.	DMREC
ERROR IN BUILDING DIRECTORY ENTRIES.	An error was encountered when DMREC attempted to add or update a directory record.	Inform data administrator, correct as directed and rerun.	DMREC (BBE)
ERROR IN BUILDING RECOVERY TABLES.	An error was encountered while attempting to retrieve a record from the directory file.	Try an update from an earlier file and inform the data administrator.	DMREC (BRT)

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
ERROR IN BUILDING *RMKDEF* FILE.	DMREC is trying to build a file of RMKDEF statements to be used in a subsequent MIPGEN run. An error has occurred in building the RMKDEF file.	Make sure that RMKDEF statements are correctly formatted in the XXJ file; then rerun the program.	DMREC GRM
ERROR IN CRM CARD ARGUMENTS.	The CRM statement in the xxj file is in error.	Correct the CRM statement and try again.	DMREC
ERROR IN CRM -PUT- (RECORD LOAD).	An error was encountered while attempting to execute a CRM put onto the data file.	Try to load from previous dump tape.	DMREC
ERROR IN EDIT PROCESSING.	Editing processor has encountered an error in trying to execute the directive.	Check the edit directive parameters and inform data administrator.	DMREC
ERROR IN IXN CARD ARGUMENTS.	The IXN statement in the xxj file is in error.	Correct the IXN statement and try again.	DMREC
ERROR IN LIST PROCESSING.	Errors in generating list as described on directive.	Check list directive for accuracy and retry.	DMREC
ERROR IN LOADING AAMI.	The loader encountered errors while loading the TAF CRM AAM interface (AAMI).	The site analyst should consult the CYBER Loader Reference Manual (listed in the preface).	TAF
ERROR IN LOADING HASH CODE filename.	The loader encountered errors while loading the hashing routine code that is on file filename.	The site analyst should consult the CYBER Loader Reference Manual (listed in the preface).	TAF
ERROR IN LOADING NAM.	The library file NETIO does not contain the entry points required for TAF.	The site analyst should consult the CYBER Loader Reference Manual (listed in the preface).	TAF
ERROR IN LOADING TOTAL.	The loader encountered errors while loading Total and the data base descriptor modules (DEMOS).	The site analyst should consult the CYBER Loader Reference Manual (listed in the preface).	TAF
ERROR IN READING TASKLIB-filename.	Error occurred during transaction executive initialization or extended memory-resident task loading. File specified as task library was incorrectly formatted; therefore, it could not be read or loaded into extended memory correctly.	Inform site analyst.	TAF
ERROR IN RECORD DUMP.	During a record dump, DMREC is unable to recognize the first record on the dump file as an FSTT.	Check structure of file to be dumped for IS, DA or AK type and try again.	DMREC
ERROR IN RETRIEVING VSN.	No VSN has been found in the directory that satisfies the directive.	Check directive parameters. If correct, inform data administrator.	DMREC
ERROR IN SUBMITTED FILE.	An incorrect parameter has been detected on a SUBMT call from a task.	Correct task or inform data base administrator.	MSABT
ERROR ON xxj FILE ARGUMENTS.	The xxj file contains statements in error, which causes the transaction subsystem to abort.	Examine xxj file. Inform TAF data administrator.	TAF
ERROR STATUS nnnB ON BACKUP DIRECTORY.	CRM error has occurred on directory file.	Inform site analyst.	DMREC (FER)
ERROR(S) ENCOUNTERED IN DMREC PROCESSING.	Fatal errors were encountered during processing.	Check the output file for the detailed error message.	DMREC
FATAL CIO ERROR STATUS.	A TAF CIO operation returned a fatal error status which aborted TAF.	Inform site analyst.	TAF
FIELD LENGTH EXCEEDED FOR CMM.	TAF does not have enough field length to allocate the space potentially required by CMM.	Increase TAF initialization field length.	TAF
FIELD LENGTH EXCEEDED FOR LOCKS.	TAF does not have enough initialization field length for allocating lock tables.	Decrease the locks parameter on the CRM statement, increase the TAF initialization field length, or inform site analyst.	TAF

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
FIELD LENGTH EXCEEDED FOR RECORD.	TAF does not have enough field length to allocate the space for the record buffer.	Decrease the record size specified in the xxJ file or increase the TAF initialization field length.	TAF
FIELD LENGTH EXCEEDED FOR USERS.	TAF does not have enough initialization field length for allocating file control tables.	Decrease the users parameter on the CRM statement, increase the TAF initialization field length, or inform site analyst.	TAF
FILE ATTACH/DEFINE ERROR	An error was detected in attaching or defining an ARF or BRf.	Check the ARFs and the BRfs for attach mode and processing mode and correct if necessary.	TAF1
xxfilen - FILE, ERROR ON LOCK FUNCTION.	An error occurred while attempting to lock the specified file. Possible causes are: - No free file lock table entry (TKOK). - Another task has file locked. - Another task has record locked.	Rerun the job at another time. If problem persists, increase the number of lock table entries.	AAMI
xxfilen - FILE, ERROR ON OPEN FUNCTION.	An error occurred while attempting to open the specified file. Possible causes are: - No free file control table entry (TFcb). - *CRM* error on OPEN function. - Incorrect key length. - Incorrect record length.	Correct error. If not possible, contact site analyst.	AAMI
FILE xxJ NOT FOUND.	Transaction subsystem aborts. Data base in TCF file has no xxJ file, or a PFM error occurred.	Inform TAF data administrator or site analyst.	TAF
FILE NAME CONFLICT.	The input file name specified on the KTSdMP command is the same as the output file name specified.	Correct error and rerun.	KTSdMP
FILE NAME MISMATCH ON TAPE HEADER RECORD.	File name on ARF tape and attached ARF don't match.	Check file name on ARF tape being used.	DMREC
FILE NAME MUST BE 2-7 CHARACTERS.	The xxpfni parameter on the CRM statement must be two to seven characters, the first two (xx) being the data base name.	Correct the xxpfni parameter on the CRM statement or inform data administrator.	TAF
FILE NOT ATTACHED - filename.	Either the P or the N parameter file was not attached when the DA option was used; filename is the permanent file name.	Ensure that the file is defined or attached prior to entering the LIBTASK statement.	LIBTASK
FILE hash NOT FOUND.	The indirect file named hash containing the binary code of the hashing routine was not found under the username parameter on the USER statement in the xxJ file or a PFM error occurred.	Ensure that file hash is saved under the username parameter or inform data administrator. Consult the CYBER Loader Reference Manual (listed in the preface).	TAF
FILE ORGANIZATION IS NOT IS, DA OR AK.	The file organization parameter on the CRM statement was not specified as either IS, DA, or AK.	Correct the CRM statement and try again.	DMREC
FILE TCF EMPTY.	An empty TCF exists under the TAF user name.	Place the necessary information on TCF.	TAF
FILE TCF NOT FOUND.	The TCF was not found under the user name of the Transaction Facility.	Create a TCF file under the TAF user name.	TAF
FILE TYPE MUST BE AK, DA, OR IS.	The type parameter on the CRM statement must be AK (actual key), IS (indexed sequential) or DA (direct access).	Correct the type parameter on the CRM statement or inform the data administrator.	TAF
FILE TYPE NOT ARF OR BRf.	On a create function a file name was used that does not conform to the ARf/BRf naming conventions.	Check file name on create directive.	DMREC
FL TOO LARGE- nnnnnnB,taskname,tasklibrary.	The initial load field length, nnnnnnB, for task taskname on task library tasklibrary exceeds the minimum size of the transient task area (potential space available to contain transient tasks). Thus a situation could arise in which it would not be possible to load the task.	Correct error.	TAF

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
FORMAT ERROR IN THE NETWORK DESCRIPTION FILE.	During transaction executive initialization, one or more errors were found to exist in the network description file.	Inform site analyst.	TAF
FSTT READ ERROR.	No FSTT found on a record load dump tape.	Load from previous tape.	DMREC
FWA .GE. LWA+1.	There is a logical error in the structure of the input file which implies that the first word address is greater than or equal to the last word address plus one.	Inform data base administra	KTSOMP
GET ERROR ON PF xxxxxxx.	No indirect access permanent file, for use as an own code file, has been found.	Check for the presence of the file and try again.	DMREC
HOST AVAILABLE CONTROL CHARACTER=char display ENTER INPUT TO CONNECT TO HOST	Your selected host is available. display The host availability display (HAD), which appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	Enter any input to initiate your connection to the host.	CCP
HOST BUSY CONTROL CHARACTER=char display ENTER char HD TO SEE HOST STATUS	The host rejected your attempt to connect. display The host availability display (HAD), which only appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	Check the status of the hosts in your configuration.	CCP
HOST CONNECTED CONTROL CHARACTER= char display READY FOR INPUT	You are connected to the host. display The host availability display (HAD), which appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	Proceed with your login to the host.	CCP
HOST CONNECTED CONTROL CHARACTER=char display TERMINAL DISABLED BY NOP	The network operator has disabled your terminal. No further input is possible. display The host availability display (HAD), which appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	None.	CCP
HOST CONNECTED CONTROL CHARACTER= char display READY FOR INPUT	You are connected to the host. display The host availability display (HAD), which appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	Proceed with your login to the host.	CCP
HOST DISCONNECTED CONTROL CHARACTER=char display ENTER INPUT TO CONNECT TO HOST	You have been disconnected from the host. display The host availability display (HAD), which appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	None.	CCP
HOST DISCONNECTED CONTROL CHARACTER=char display TERMINAL DISABLED BY NOP	The network operator has disabled your terminal. No further input is possible. display The host availability display (HAD), which appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	None.	CCP
HOST NOT AVAILABLE.	NAM is not communicating with the 255x communications processors. Either NAM was not initialized or has since failed.	Initialize NAM if it was not initialized previously; inform site analyst if NAM was active but a malfunction occurred.	TAF

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
HOST UNAVAILABLE CONTROL CHARACTER=char display ENTER char HD TO SEE HOST STATUS	You have attempted a connection to a host that is not up. display The host availability display (HAD), which appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	Check the status of the hosts in your configuration.	CCP
IGNORE TABLE OVERFLOW.	Ignore table, TTIG, is too small.	Inform site analyst to enlarge TTIG table.	DMREC
xx ILLEGAL COMMAND	The characters xx must be a terminal definition mnemonic or interactive status command.	Ensure the accuracy of your entry and retry.	CCP
ILLEGAL DATA BASE IN xxJ FILE.	One of the statements in the xxJ file specifies an incorrect xx parameter and causes the transaction subsystem to abort.	Examine xxJ files. Inform the TAF data administrator.	TAF
ILLEGAL FL REDUCTION.	Task request to reduce FL would cause the last word address of the communication block to be outside the task FL.	Inform data administrator.	MSABT
ILLEGAL JOURNAL FILE REQUEST.	Either a task requested that an entry be made on an undefined TAF journal file, or the task attempted to make an entry longer than 2499 CM words on a TAF journal file.	Inform data base administra	MSABT
ILLEGAL KEY BEGINNING CHARACTER.	The key offset parameter on an AKY statement in the xxJ file, is missing or incorrect.	Correct AKY command.	TAF (AKY)
ILLEGAL KEY ORDINAL.	The key ordinal parameter on the AKY statement is missing or incorrect.	Correct AKY statement.	TAF (AKY)
ILLEGAL LIBTASK ATTEMPT - filename, username.	The transaction executive validates all dynamic attempts to change the task library by comparing the user name of the requester against the list of data base user names. If it does not match, or if the library file is not attached by TAF, the transaction executive issues this dayfile message, where username is the user name of the incorrect attempt.	Correct and reinitialize transaction executive.	TAF
ILLEGAL NUMBER FOR LOCKS.	The Locks parameter on the CYBER Record Manager (CRM) statement is in error. One of the following format conditions exists. - A nonnumeric character. - A character after a postradix of B or D. - An 8 or 9 with a postradix of B.	Correct the Locks parameter on the CRM statement or inform the analyst.	TAF
ILLEGAL NUMBER FOR USERS.	The user's parameter on the CRM statement is in error. One of the following format conditions exists. - A nonnumeric character. - A character after a postradix of B or D. - An 8 or 9 with a postradix of B.	Correct the user's parameter on the CRM statement or inform the data administrator.	TAF
ILLEGAL NUMBER OF ALTERNATE KEYS.	The number of alternate keys on the IXN statement is missing or incorrect.	Correct IXN statement.	TAF (IXN)
ILLEGAL PARAMETER IN RA+1 CALL.	A task error. The parameters specified in a request from a task are either insufficient or incorrect.	Correct task or inform data base administrator.	MSABT
ILLEGAL TASK NAME REQUESTED FOR SCHEDULING.	The task requested was not found in the task library directory.	Correct task or inform data base administrator.	MSABT
cmd ILLEGAL TERMINAL CLASS	The value you specified in the terminal definition command cmd is not valid for your terminal.	Choose an appropriate value (refer to appendix J).	CCP
ILLEGAL TERMINAL INPUT REQUEST.	A task error. A WAITINP request was issued by a memory resident task or by a task that did not originate from a terminal.	Correct task or inform data base administrator.	MSABT
cmd ILLEGAL VALUE	You specified an invalid value in the terminal definition command cmd.	Choose a valid value (refer to appendix J).	CCP
INACTIVE TASK REQUESTED.	An attempt was made to schedule a task which has been turned off with a K.OFFTASK command or a LIBTASK directive.	Correct condition or inform site analyst.	OFFTASK
INCOMPLETE PARAMETER.	A parameter on a DMREC directive was not completed correctly.	Correct the parameter on the faulty directive.	DMREC

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
INCORRECT COMMAND.	One of the following errors occurred: 1. A command was entered other than what was on the menu. 2. A terminal CRMSTAT request did not request its own data base. 3. A terminal origin job tried to request the menu. 4. Unpack errors on the terminal message.	Ensure that the command is correct. If problem persists, inform the site analyst.	CRMTASK
INCORRECT COMMON MEMORY MANAGER REQUEST.	Memory request with reserved bits in the parameter block set incorrectly.	Do not set reserved bits.	MSABT
INCORRECT FAMILY NAME IN EDT.	Self-explanatory.	Reinitialize the transaction executive or inform the site analyst.	TAF1
INCORRECT FWA FOR LOAD.	The FWA of the load was not equal to 111B. Task binaries must be in absolute format.	Ensure that the correct load procedure is performed.	LIBTASK
INCORRECT POSITION IN THE DIRECTORY.	The directory structure is inconsistent or positioning is incorrect.	Inform data administrator.	DMREC (GPL)
INCORRECT SUB-TRANSACTION CODE - INITIAL TASK.	There is an unrecognizable subtransaction code in the terminal input.	Correct terminal input or inform data base administrator.	MSABT
INCORRECT TOTAL INTERLOCK REQUEST.	One of the following. - A task that is not on the system task library issued a CHKON request. - A task issued a CHKON request with a CHKON request already in effect. - A task issued a CHKOFF request without first issuing a CHKON request. - A task that issued a CHKOFF request had a sequence number different from a task that issued a CHKON request.	Inform data base administrator.	MSABT
INCORRECT WORD COUNT ON TERMINAL OUTPUT.	One or more of the following task errors occurred on a SEND request. - The word count was less than or equal to zero. - There was an attempt to use SEND MAXWS or more CM words of information in one SEND; MAXWS is an installation option. - There was an attempt to use SEND MAXTO or more CM words of information by one task through a series of SEND requests. MAXTO is an installation option.	Correct task or inform data base administrator.	MSABT
INITIAL TASK NOT IN TASK LIBRARY DIRECTORY.	The task library file does not contain the initial task (ITASK).	Inform site analyst.	TAF
INITIALIZATION OPTIONS.	This message precedes messages indicating the values of the initial K display options either during initialization or recovery.	None.	TAF
INPUT DISCARDED	The network has discarded the last input from your terminal. May be caused by entering a user break or interactive status command while awaiting a response to a previous user break or interactive status command.	Reenter your last input.	CCP
INPUT DISCARDED	The system has discarded user input.	Reenter input.	TAF1
INSUFFICIENT FIELD LENGTH FOR THIS COMMAND.	CRMTASK issued a CRMSTAT request but AAMI was not able to complete it because the table area supplied by the user was not large enough.	Inform data base administrator.	CRMTASK
INSUFFICIENT FL FOR DATA MANAGER.	The transaction executive requires more field length at initialization time than is available.	Correct error and reinitialize executive.	TAF
INTERMEDIATE IGNORE TABLE OVERFLOW.	Intermediate ignore table FTAB is too small. Too many potentially recoverable files have been encountered when reading ARF's.	Inform site analyst to enlarge table FTAB.	DMREC (BRT)
INVALID DATA BASE NAME ON DMS STATEMENT.	A data base name associated with TAF, CRM, or OTHER exceeds two characters.	Correct the DMS statement on TCF file.	TAF
INVALID DATA MANAGER PARAMETER	An error has been detected in the parameter list on a data manager request.	Correct the data manager request statement.	MSABT

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
INVALID TASK NAME.	The error message was issued for one of the following reasons: - Initialization of a task by the name of BTASK, CTASK, ITASK, KDIS, MSABT, OFFTASK, RCTASK, RTASK, SYMSG, or XTASK was attempted using XTASK. These are names of required system tasks and will not be initiated by XTASK. - A zero length task name was specified. - A task name of more than seven characters was specified.	Correct the task name.	XTASK
INVALID TCF ENTRY.	The previous dayfile message is the statement in TCF which is incorrect.	Correct that statement in TCF.	TAFREC
INVALID TRANSACTION DIRECTORY HEADER - filename. filenam IS INCONSISTENT.	The transaction directory (TRD) header word is not TRD. The inconsistent file bit is set for recoverable file, filenam. This indicates a failure (either of TAF or the sytem) occurred while CRM was trying to update more than one block. All of the blocks may not have been updated and this would cause the file to be flagged as inconsistent.	Inform TAF site analyst. It is necessary to recover the affected file. AAMI will submit an OMREC job to do this.	TAF AAMI
xxJ FILE ERROR - CRM DM NOT LOADED.	A data manager related statement is specified in the xxJ file but that data manager is not loaded on to TAF.	Inform site analyst to change the xxJ or TCF file.	TAF
xxJ FILE ERROR - TAF DM NOT LOADED.	A data manager related statement is specified in the xxJ file but that data manager is not loaded on to TAF.	Inform site analyst to change the xxJ or TCF file.	TAF
xxJ FILE ERROR - TOTAL DM NOT LOADED.	A data manager related statement is specified in the xxJ file but that data manager is not loaded on to TAF.	Inform site analyst to change the xxJ or TCF file.	TAF
xxJ FILE NOT FOUND.	When using the TAF's user name and password, an xxJ file for this data base was not found.	Ensure xxJ file exists for this data base and try again.	DMREC
JOURNAL TYPE DOES NOT MATCH xxJ FILE.	TAF journal file entries in the xxJ file do not match the files themselves. This causes the transaction subsystem to abort.	Inform TAF data administrator. Examine xxJ file for the TAF journal file entries.	TAF
K.BFL=n. K.CMB=nn. K.ECS=nnK. K.ERO=a. K.ERO=CRF,xxx. K.INT=1. K.INT=CRF,xxxxxx. K.INT=CRM,xxxx. K.MDM=n. K.MFL=nnnnnnB. K.REC=a. K.SCP=nn. K.TLF=a.	Values of the initial K display options at either initialization or recovery.	None.	TAF
K. COMMAND NOT VALID.	A K. command in the TCF file is improperly formatted.	Correct the statement in the TCF or inform the TAF analyst.	TAF
K-DISPLAY COMMAND UNSUCCESSFUL, DROP IGNORED--command.	A K.DROP or K.DDROP could not be processed because the task was in recall, the interactive executive denied a request, or a command was issued during the initial load of the task. The command field is the attempted command or request; blanks indicate no command was issued.	Reenter the K.DROP or K.DDROP command. When the recall operation, time-sharing request, or initial load operation is completed, TAF accepts the command and aborts the task.	SYMSG
K-DISPLAY COMMAND UNSUCCESSFUL, FORMAT ERROR--command.	There is an error in the syntax of the command or in one of the parameters. The command field is the attempted command or blanks if the command is longer than seven characters or contains an incorrect character.	Correct the command and retry the operation.	SYMSG
K-DISPLAY COMMAND UNSUCCESSFUL, ILLEGAL COMMAND--command.	The command is not a valid K display command or was misspelled. The command field is the attempted command.	Correct the command and retry the operation.	SYMSG
K DISPLAY COMMAND UNSUCCESSFUL, SYSTEM BUSY--command.	The system is busy and cannot respond to the request. The command field is the attempted command.	Retry the operation.	SYMSG

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
K.MAXFL,nnnnnnB.	The run-time K display command K.MAXFL was entered with the indicated value.	None.	TAF
K.MAXFL REJECTED.	A value was entered which caused potential blocked tasks to be detected.	Reenter K.MAXFL with a larger value.	TAF
KEYWORD IS ILLEGAL FOR THIS FUNCTION.	A keyword was used that is not valid for the selected directive.	Check format of directive and valid key words for that directive.	DMREC
n.nnn KILO CDCS REQUEST REJECTS FOR BUSY.	Total number of SSC rejects for busy when less than seven outstanding CDCS SSC requests existed at the time of the current request.	None.	TAF
n.nnn KILO CDCS REQUEST REJECTS FOR MAXR.	Total number of SSC attempts when there were seven (MAXR) outstanding CDCS SSC requests.	None.	TAF
n.nnn KILO CDCS REQUESTS FROM TASKS.	Total number of CDCS SSC requests issued by tasks. The number does not include terminate requests which are blocked by TAF.	None.	TAF
n.nnn KILO ITASK RELOADS.	Upon transaction facility termination, this message indicates how many reloads of ITASK occurred. Because ITASK is a central memory resident task, if it makes a fatal error, it must be reloaded.	Inform site analyst.	TAF
n.nnn KILO LOCK REJECTS-filename.	TAF CRM could not lock records in filename for a task because the locks parameter on the CRM statement for that file was not large enough.	The site analyst should consider increasing the locks parameter on the CRM statement if lock failures occur frequently.	TAF
n.nnn KILO LOCKS-filename.	TAF CRM processed nnnn locks for file filename.	None.	TAF
n.nnn KILO NO COMMUNICATION BLOCKS.	Upon transaction facility termination, this message indicates the number of times a communication block was needed and was not available.	Inform site analyst. The site analyst might want to consider using the K.CMB command during TAF initialization.	TAF
n.nnn KILO NO FL FOR TASK LOAD.	Upon transaction facility termination, this message indicates the number of times TAF was unable to obtain enough memory to load tasks.	Inform site analyst. The site analyst might want to consider using the K.MFL command during TAF initialization.	TAF
n.nnn KILO NO SUBCONTROL POINTS.	Upon transaction facility termination, this message indicates the number of times a subcontrol point area was needed and was not available.	Inform site analyst. The site analyst might want to consider using the K.SCP command during TAF initialization.	TAF
n.nnn KILO OPEN REJECTS-filename.	TAF CRM could not open file filename for a task because the user's parameter on the CRM statement is not large enough.	The site analyst should consider increasing the users parameter on the CRM statement if open failures occur frequently.	TAF
n.nnn KILO OPENS-filename.	TAF CRM processed nnnn opens for file filename.	None.	TAF
n.nnn KILO STORAGE MOVES OF TASKS.	Upon transaction facility termination, this message indicates the number of subcontrol points that were moved. This is done to make more effective use of the field length of TAF.	None.	TAF
n.nnn KILO TAF FL INCREASES.	Upon transaction facility termination, this message indicates the number of times TAF increased its field length while trying to satisfy an internal request for a block of memory. This internal request is often issued to obtain memory in which to load a task.	None.	TAF
n.nnn KILO TASK RECALLS.	Upon transaction facility termination, this message indicates the number of times tasks were put in recall. This is an internal statistic over which the applications programmer has little control.	None.	TAF

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
n.nnn KILO TASK RECALLS FOR OUTPUT.	Upon transaction facility termination, this message indicates the number of times tasks were put in recall because the network block limit (NBL parameter on a TERMINAL statement) for the terminal in use was reached.	Refer to the Network Access Method Network Definition Language Reference Manual (listed in the preface).	TAF
n.nnn KILO TASK RELOADS.	Upon transaction facility termination, this message indicates how many reloads of tasks occurred. TAF reloads tasks that have fatal errors if one of the following is true. - The task is central memory resident. - There are queued transactions waiting to use the task.	Inform site analyst.	TAF
n.nnn KILO TASK ROLLOUT COMPLETES.	Upon transaction facility termination, this message indicates the number of times TAF rolled out tasks. This statistic excludes the number of times the rollout operation did not complete.	None.	TAF
n.nnn KILO TASK ROLLOUT STARTS.	Upon transaction facility termination, this message indicates the number of times TAF initiated a rollout operation. Rollout includes writing a task to disk and releasing the task subcontrol point space. In some cases, the task is allowed to stay central memory resident for a while before it is rolled out.	None.	TAF
n.nnn KILO TRANSACTION ABORTS.	Upon transaction termination, this message indicates how many transaction tasks have aborted.	Data administrator may have to correct data base to account for transactions.	TAF
n.nnn KILO TRANSACTIONS PROCESSED.	Upon TAF termination, this message indicates the number of TAF transactions processed.	None.	TAF
KL PARAMETER OR CRM CARD NOT SPECIFIED PROPERLY.	The KL parameter or the CRM statement was specified improperly or specified as zero length.	Correct the CRM statement and try again.	DMREC
KTSOMP COMPLETE.	Dump completed normally.	None.	KTSOMP
LBL - CIO ERROR.	A CIO error was encountered because no trailer record was found on a block load.	Load from previous dump tape.	DMREC
LIBRARY DIRECTORY EMPTY - filename.	The task library file indicated does not contain a directory.	Inform site analyst.	TAF
LIBRARY DIRECTORY ERROR - filename.	The task library file indicated contains a nonrecognizable directory.	Inform site analyst.	TAF
LIBRARY DIRECTORY TOO LONG - filename.	The directory record on the task library file indicated exceeded the maximum length allowed by the transaction executive (398 entries).	Inform site analyst.	TAF
LIBRARY FILE TIME OUT.	The library file was busy for the length of time specified by an assembly constant, so LIBTASK was unable to attach it. Another user apparently has the file attached and will not allow modification.	Retry at a later time.	LIBTASK
LIBTASK ERROR CONDITION UNKNOWN.	The error condition for the error processing routine ERR within LIBTASK was unknown.	Inform data base administra	LIBTASK
LIBTASK PFM ERROR DETECTED.	A permanent file error other than FILE BUSY was received when the library file attach was attempted.	Correct and rerun.	LIBTASK
LIST COMPLETE.	Informative message.	None.	DMREC
LOAD ERROR IN HASHING ROUTINE.	When attempting to load the hashing routine on a record load operation, a load error or no entry point in the hashing routine was found.	Check for valid hashing routine, inform analyst.	DMREC
LOADING ECS taskname.	Informative message. The transaction subsystem is loading task taskname.	None.	TAF
LOG ENTRY TABLE OVERFLOW.	Too many recoverable data base file names exist on ARF.	Increase size of table (TLOG).	DMREC (BLT)

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
MAXIMUM TERMINALS EXCEEDED.	More than IPTST transaction terminals have been defined in the Network File(s).	Reduce number of terminal definition statements or increase IPTST and reassemble TAFREC.	TAFREC
MEMORY OVERFLOW DURING INITIALIZATION.	TAF aborted because its field length for initialization was insufficient.	Inform site analyst. IFL= in deck TAF should be increased. Increasing the central memory field length parameter on the RFL command in the TAF initialization procedure file (ffff) does not correct this problem.	TAF
MEMORY REQUEST FOR ECS NOT ALLOWED.	Incorrect request of extended memory by task.	Contact data administrator.	MSABT
MEMORY REQUEST WITH D.M. REQUEST OUTSTANDING.	A task error. MEM or RFL request attempted with data manager requests outstanding.	Correct task or inform data base administrator.	MSABT
MFL TOO LARGE - nnnnnB,taskname,tasklibrary.	The MFL (initial field length plus expandable field length) of the non-CM resident task (taskname) on task library (tasklibrary) exceeds the minimum size of the transient task area (potential space available to contain transient tasks). Thus a situation could arise in which it would not be possible to complete processing of this task.	Reduce the task FL or EF, or increase the TAF FL.	TAF
MINIMUM TAF MFL NEEDED = nnnnnB.	Potentially blocked tasks were detected at one of the following times: - TAF initialization - Attempted task library update - Attempt to change TAF maximum FL via K.MAXFL command The above operation did not complete normally. The maximum FL of TAF must be at least nnnnnB. If nnnnnB exceeds the largest field length possible for TAF (377700B), then other corrective action is needed.	Correct error.	TAF
MISSING AKY CARD.	Number of AKY statements does not match number of alternate keys on IXN statement.	Modify the IXN statement or adjust the number of AKY statements so the two match.	TAF (XXJ)
MISSING HEADER WORD ON xxJ FILE.	The first statement on the xxJ file is in error, causing the transaction subsystem to abort.	Examine xxJ files for header xxJ. Inform the TAF data administrator.	TAF
MISSING HEADER WORD ON XXJ FILE.	No header word on xxJ was found.	Correct xxJ file and rerun.	DMREC
MORE THAN FIVE TASKS IN TASK CHAIN.	A task error. Only five tasks can be requested for scheduling in one CALLRTN, CALLTSK, or NEWTRAN request; more than five were requested.	Correct task or inform data base administrator.	MSABT
MORE THAN ONE ARF SPECIFIED.	More than one ARF was specified on the DUMP directive.	Correct the directive and rerun.	DMREC
MORE THAN ONE ENTRY POINT ON A *54* TABLE.	Multiple entry points are not allowed in tasks.	Rewrite the task with only one entry point.	LIBTASK
MORE THAN ONE USER STATEMENT IN TCF.	Self-explanatory.	Delete the extra USER statement.	TAFREC
MRL PARAMETER ON CRM CARD NOT SPECIFIED PROPERLY.	The MRL parameter on the CRM statement was specified improperly or specified as zero length.	Correct the CRM statement and try again.	DMREC
MS REDUCTION INVALID - filename.	The maximum message size, specified via the MS parameter on the RECOVER directive, was less than the actual message record size on the named CRF.	Correct the RECOVER directive or select initialization of the named CRF via the K.INT initial K display command if reducing the message size is actually required. (Note that K.INT will destroy the current recovery information).	TAFREC

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
MSABT CALL ERROR.	MSABT was called with an incorrect error code.	Inform data base administrator.	MSABT
NAKY PARAMETER ON THE IXN CARD NOT SPECIFIED PROPERLY.	The NAKY parameter is not specified properly or of zero length.	Correct the IXN statement and try again.	DMREC
NAM ERROR - ILLOGICAL ABT.	The application block type (ABT) sent to TAF by NAM is unrecognizable.	Contact Central Software Support.	TAF
NAM ERROR - INCORRECT ABH.	The application block header (ABH) sent to TAF by NAM is unrecognizable.	Contact Central Software Support.	TAF
NAM FUNCTION NOT FOUND.	TAF received a supervisory message from NAM which had an unrecognizable primary or secondary function code.	Contact Central Software Support.	TAF
NAM LOGICAL ERROR.	NAM sent TAF a message out of order or an unrecognizable message.	Contact Central Software Support.	TAF
NAM NOT AVAILABLE.	Informative message indicating that TAF is currently at a control point but NAM is not. TAF transactions can be initiated from batch only, or TAF-CRM data bases may be accessed from batch or interactive jobs.	Bring NAM to a control point, if desired.	TAF
NAM PHYSICAL ERROR EC=ec.	NAM has detected a physical error indicated by error code ec.	Refer to the NAM Reference Manual for the meaning of this error code.	TAF
NAM REJECT.	During login processing, NAM rejected the terminal.	Inform site analyst.	TAF
NEED AT LEAST xx SUBCONTROL POINTS.	There are more CM resident tasks defined than subcontrol points. If non-CM resident tasks exist, there must be at least one more subcontrol point than there are CM resident tasks.	Reinitialize the transaction executive and assign more subcontrol points, or reduce the number of CM resident tasks.	TAF
NETOFF COMPLETE.	Informative message indicating that TAF is no longer communicating with NAM. NAM initiated shutdown procedures prior to loss of communications.	When NAM is available, the central site console operator command K.NAMON can be used to resume communications between TAF and NAM.	TAF
NETON COMPLETE.	Informative message indicating that TAF is communicating with NAM.	None.	TAF
NETWORK FILE NOT FOUND - filename.	The network description file, NCTFi, could not be found.	Check that correct parameters were specified on the NETWORK directive corresponding to filename.	TAF TAFREC
NETWORK SHUT DOWN DETECTED.	Self-explanatory.	None.	TAF
NM REDUCTION INVALID - filename.	The maximum number of user messages, specified by the NM parameter on the RECOVER directive, was less than the actual number of user messages on the named CRF.	Correct the RECOVER directive or select initialization of the named CRF via the K.INT initial K display command if reducing the number of user messages is actually required. (Note that K.INT will destroy the current recovery information).	TAFREC
NO ACCOUNT/USER CARD IN xxJ FILE.	The USER statement in the xxJ file is not present, causing the transaction subsystem to abort.	Add USER statement in xxJ file. Inform the TAF data administrator.	TAF
NO ACCOUNT/USER CARD ON XXJ FILE.	No user statement exists on the xxJ file.	Correct the xxJ file and rerun.	DMREC (CXJ)
NO ALTERNATE KEY SPECIFIED ON IXN CARD.	Self-explanatory.	Correct the IXN statement and try again.	DMREC
NO ARF DUMP ENTRIES - DUMP IGNORED.	When trying to build directory entries for an ARF dump function, no ARF file was found.	Include ARF file name on dump.	DMREC
NO ARF DUMP ENTRIES IN DIRECTORY.	A search of the directory for the targeted VSNs of ARF tapes was unsuccessful.	Check time and data or VSN used to delineate update.	DMREC

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
NO BRF CARD FOR CRM DATA BASE.	There is no BRF statement in the xxJ file before the first CRM statement for the data base.	A BRF statement is required. If no recovery is desired, the number of specified files on the BRF should be zero.	TAF
NO BRF CARD FOUND IN xxJ FILE.	There is no BRF statement in the xxJ file for the data base.	A BRF card is required and must be added to the xxJ file.	DMREC
NO CRM CARD FOUND IN xxJ FILE.	Self-explanatory.	Correct the CRM statement and try again.	DMREC
NO DATA BASE ID FOR DATA MANAGER.	At least one data base identifier must be specified on each active (ON) DMS statement.	Add data base identifier to DMS statement(s) or specify status as OFF.	TAF
NO DATA BASE NAME IN xxJ FOR TOTAL.	Self-explanatory.	Add data base name to xxJ file.	TAF
NO DIRECTIVES.	No directives were supplied to DMREC.	Supply the correct directives and rerun.	DMREC
NO DUMP RECORD WITH SPECIFIED VSN.	DMREC could not locate a dump record with the specified VSN.	Check directory with list directive for proper VSN.	DMREC
NO FILE NAME SPECIFIED ON CRM CARD.	A CRM statement in the xxJ file did not contain a file name.	Correct the xxJ file and try again.	DMREC
NO FILE NAME SPECIFIED ON IXN CARD.	The xxJ file has an IXN statement that does not contain a file name.	Correct IXN statement on xxJ file and rerun.	DMREC (RXJ)
NO FILES TO DUMP.	No files have been specified on a dump directive.	Include file name on directive.	DMREC
NO HOSE CONNECTED.	YOU HAVE ENTERED A (ct)TM COMMAND WHEN THERE IS NO HOST CONNECTION.	Enter (ct)HD to display host status and then enter (ct)HN=nn or =mm to connect to the selected host.	RTN?
NO HOST AVAILABLE CONTROL CHARACTER=char display ENTER char HD TO SEE HOST STATUS	No hosts are available. display The host availability display (HAD), which appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	Check the status of the hosts in your configuration.	CCP
NO HOST CONNECTED CONTROL CHARACTER=char display ENTER INPUT TO CONNECT TO HOST	You have attempted to disconnect your terminal from a host when you have no connection to a host. display The host availability display (HAD), which only appears if you have enabled it using the HD terminal definition command.	Enter a HD terminal definition command to determine the status of hosts in the system, or enter any input to initiate your connection to the host.	CCP
NO HOST SELECTED CONTROL CHARACTER=char display ENTER char HN=NODE NUMBER or HS=NAME TO SELECT HOST	You have not selected a host and a host is available. display The host availability display (HAD), which appears if you have enabled it using the HD terminal definition command. char The network control character for your terminal.	Select a host.	CCP
NO KEY LENGTH SPECIFIED ON CRM CARD.	The KL parameter on the CRM statement was not specified.	Correct the CRM statement and try again.	DMREC
NO MAXIMUM RECORD LENGTH SPECIFIED ON THE CRM CARD.	The MRL parameter on the CRM statement was not specified.	Correct the CRM statement and try again.	DMREC
NO RECORD FOUND FOR GIVEN VSN - DATE/TIME.	The file specified or implied on the load cannot be found on the directory.	Check load directive for correct file loading parameters.	DMREC
NO SPACE FOR ARF/BRF BUFFER.	No space was available for an ARF or BRF buffer.	Increase field length of DMREC.	DMREC

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
NO TERMINALS IN NETWORK FILE - filename.	A valid network file was found but no transaction terminals were defined in it.	Ensure the network file is correctly named. The network file is NCTFi (i=id specified on network directive in TCF).	TAFREC
NO TRANSACTION NAME IS DEFINED IN TASK LIBRARY.	No named transaction appears in any task library.	If appropriate, define named transactions.	TAF
NOT ALL DATA BASE FILES UPPED.	CRMTASK issued a DBUP request but AAMI was unable to attach/open all of the data base files.	Inform data base administrator.	CRMTASK
NOT AVAILABLE.	K display message indicating that the entry is not available for this version.	None.	TAF
filename NOT FOUND.	The user specified the P option on the KTSMDP command; one of the two files was not in the user's permanent file catalog or a PFM error occurred.	Specify a different file name or inform a site analyst.	KTSMDP
xxxxxxx NOT FOUND IN DIRECTORY.	Specified file is not found in the directory.	Check the file name and the list directory.	DMREC
xxxxxxx NOT IN DIRECTORY.	Error - The task or named TAF transaction unit, xxxxxxx, was specified in the TN parameter but was not found on the task library or transaction unit directory.	Ensure the correct task or transaction name was specified with the TN parameter. Check task or transaction listing for presence of specified unit.	TAF
xxxxxx NOT INITIALIZED BY TOTAL. STATUS IS yyyy.	An error was encountered on the TOTAL data base.	Regenerate TOTAL data base. Refer to TOTAL Reference Manual for status.	TAF
NOTE FAILURE, THEN TYPE IN CFO, JSN. GO.	An error occurred that requires operator notification.	The operator should log the failure, or notify the data base administrator and/or correct as directed, if the error was not recovered by DMREC.	DMREC
NULL DESCRIPTION FILE.	Self-explanatory.	Create a description file (NCTFi where i is the machine identifier specified on the network statement in the TCF).	TAF
OFF TASK taskname-LIBRARY libraryname.	Task taskname in task library libraryname could not be loaded from extended memory or recovered and loaded from mass storage. Task was turned off. TAF transactions using tasks will abort.	Inform site analyst. Library must be recreated.	TAF
ONLY *BLOCK* or *RECORD* CAN FOLLOW FILE NAME.	Self-explanatory.	Correct error and rerun.	DMREC
OPEN ERROR ON COPY OF THE DIRECTORY.	An open error was generated when trying to access the directory file.	Inform data administrator.	DMREC
OPERATOR OVERRIDE.	Operator has overridden TAF.	None.	TAF
OPERATOR TERMINATION.	Operator typed in K.STOP in K display command.	None.	TAFREC
PACKED DATE/TIME CONVERSION ERROR.	An error was generated when converting the directive date/time to a packed date/time.	Check the date and time used on directive and rerun.	DMREC
PARAMETER FORMAT ERROR.	The input directive contains an incorrect file name - too many characters or nonalphanumeric characters.	Correct directive and rerun.	DMREC
PARITY ERROR IN TAPE WITH VSN=vsnn.	Tape with vsnn has been mounted and has a parity error.	Clean the tape or replace it.	DMREC
nnnn PER CENT CPU USAGE.	Summary message indicating CPU usage by the transaction subsystem.	None.	TAF
PERCENT PARAMETER NOT SPECIFIED PROPERLY.	The percent parameter was not of the correct format or was specified greater than 100.	Correct directive and rerun.	DMREC
PF xxxxxxx - NOT ON xxJ FILE.	The xxJ file does not have the specified CRM data file defined.	Make necessary xxJ file entry (CRM statement).	DMREC (RXJ)

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
PF xxxxxxx - READ ONLY.	An attempt has been made to write on a file defined in the xxJ file as read only.	Change the xxJ file access mode.	DMREC
POSSIBLE BLOCKAGE AMONG CM RESIDENT TASKS.	The sum of the maximum field lengths (MFLs) for the CM resident tasks exceeds the minimum size of the total task area (potential space available to contain tasks). Thus one or more CM resident tasks could be blocked from completing.	Correct error.	TAF
POTENTIALLY BLOCKED TASKS DETECTED.	During TAF initialization, potentially blocked tasks were detected. Preceding error messages contain additional details.	Correct error.	TAF
RA+1 CALL ERROR.	There was an unrecognizable task request.	Correct task or inform data base administrator.	MSABT
RA+1 CALL PARAMETER ADDRESS OUTSIDE FL.	A parameter in a task request was outside the field length of the task.	Correct task or inform data base administrator.	MSABT
RA+1 CALL WITH ILLEGAL FUNCTION CODE.	There was an unrecognizable task request for system action.	Correct task or inform data base administrator.	MSABT
READ ERROR ON TAPE.	Error in attempting to read an after image dump tape.	Inform the data administrator.	DMREC
READ/WRITE ERROR ON TAPE.	A tape error has been encountered. If possible, use another tape.	On all dumps, use another tape; on other DMREC functions, inform data administrator.	DMREC
RECORD NUMBER ERROR.	No header or an incorrect header or a missing header has been found on an ARF when updating a file.	Inform data administrator.	DMREC
xx - RECOVERABLE DATA BASE.	TAF/CRM is beginning task recovery process for recoverable data base xx.	None.	AAMI
RECOVERABLE FILE PARAMETER MUST BE *R* OR *N*	The recovery parameter of the *CRM* statement in the *xxJ* file is incorrect.	Fix the *CRM* statement.	DRNEC
RECOVERABLE RUN UNIT DELETED - username.	The recoverable run unit for username which appeared on the CRF has been deleted because username did not appear in the corresponding NCTFi file.	None.	TAFREC
RECOVERY COMPLETE.	The transaction executive or interactive subsystem has successfully completed recovery.	None.	TAF IAFEX
RECOVERY FILE INCONSISTENT	The ARF or BRF table header information does not match the buffer header information.	Dump the database associated with the inconsistent ARF or BRF using DMREC. Re-create the ARF or BRF using DMREC and reinitialize the transaction subsystem.	TAF1
RECOVERY FILE INITIALIZATION ERROR	An error has occurred during initialization because of the ARF or BRF file.	Refer to the error message following this one for the error condition description.	AAMI
RECOVERY IMPOSSIBLE.	A fatal error occurred during transaction executive initialization or the recovery file prepared by the TAF initialization routine does not exist or could not be read.	Check dayfile for a specific error message or inform the site analyst.	TAF
RECOVERY NOT POSSIBLE NOW BECAUSE DATA BASE FILE IS DOWN. SELECT DESIRED OPTION FROM FOLLOWING LIST BY ENTERING LETTER.	Self-explanatory.	Inform data base administrator. The data base file must be brought up for recovery.	RTASK
A WAIT 5 MINUTES AND TRY AGAIN. TAF WILL WAIT 5 MINUTES BEFORE ATTEMPTING RERUN OF TRANSACTION.			
B LOG OFF. THE PREVIOUS RECOVERABLE INFORMATION WILL REMAIN INTACT. LOG IN AT A LATER TIME TO ATTEMPT RERUN.			
C ALLOW ENTRY OF NEXT TRANSACTION. THE PREVIOUS RECOVERABLE INFORMATION WILL BE LOST.			

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
RECOVERY, RERUNNING TRANSACTION.	The last entered TAF transaction input caused TAF a rerunnable transaction to be executed. A failure occurred before the completion of the transaction and the transaction executive is rerunning the transaction.	Wait for TAF transaction output.	RTASK
RECOVERY, TRANSACTION NOT RERUNNABLE, ENTER NEXT TRANSACTION.	An input/output error in the TRF caused loss of recovery data and the system cannot restart the TAF transaction automatically.	Inform the data administrator or the transaction administrator. The data base may or may not have been committed. Do not rerun the TAF transaction until the state of the data base can be determined.	RTASK
REFERENCE MADE TO INCORRECT TERMINAL NAME.	The terminal name or user name specified has not been defined.	Correct task or inform data base administrator.	MSABT
REPEAT..	Data will now be accepted by the network.	None.	CCP
REQUEST COMPLETE.	The terminal origin CRMTASK request is complete.	None.	CRMTASK
REQUESTED ECS NOT AVAILABLE.	The amount of extended memory requested was not available in a contiguous block.	Reinitialize with less extended memory requested.	TAF
SEND TO TERMINAL NOT LOGGED IN.	A terminal issued a SEND without recall and one of the following occurred. - TAF was not connected to NAM. - The terminal to which the SEND was attempted was not logged into TAF. - TAF received a STOP supervisory message for the destination terminal.	Correct error condition or inform data base administrator.	MSABT
SSF FUNCTION nn RECEIVED ERROR mm FOR zzzzzzzz.	TAF received error code mm (octal) while issuing SSF function code nn (octal) for job zzzzzzzz.	Inform TAF site analyst.	TAF
SYSTEM ABORT OF TASK.	The operator aborted the task (via a K.DROP or K.DDROP command), or a data manager error caused the task to abort.	Inform data base administrator.	MSABT
SYSTEM TASK LIBRARY MISSING TASK, taskname.	Taskname is not on system library or not enough communication blocks are allocated.	Add taskname to system library or increase number of communication blocks.	TAF
TABLE OVERFLOW-TOO MANY TASKS.	The managed tables used by LIBTASK are not large enough for the number of tasks or directives.	LIBTASK must be reassembled with larger table allocation.	LIBTASK
TAF COMMUNICATIONS RECOVERY FILE NOT FOUND - filename.	The recovery file could not be found.	- Check that the correct familyname and user name were specified on NETWORK statement. - Check if TRFi should be initialized via K.INT initial K display command.	TAFREC
TAF/CRM CRM STATUS nnnnB IN FUNCTION yyyyyyy.	A CRM status error occurred on a TAF/CRM request.	Check AAM reference manual for meaning of the status code and correct error.	BAAML
TAF/CRM DATA MANAGER NOT LOADED IN TAF.	This message is returned to the dayfile of a batch concurrency job that attempted to access a CRM file while CRM was not loaded in TAF.	Ask the data base analyst to bring up TAF with CRM. Rerun job when TAF/CRM is available.	BAAML
TAF/CRM FUNCTION CODE NOT VALID.	TAF/CRM cannot process the request issued because it is an unrecognizable function code. This is an internal error.	Inform data base administrator.	BAAML
TAF/CRM TAF status nnn IN FUNCTION yyyyyyy.	A TAF status error occurred on a TAF/CRM request.	Check TAF/CRM reference manual for meaning of status code and correct error.	BAAML
TAF DATA NOT WITHIN UCP FL.	TAF cannot access data from user program because the address specified is outside the user control point field length.	Correct program.	BAAML

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
TAF ERROR CODE NOT DEFINED.	TAF or TAF/CRM has returned an error code that the routine RQS was not programmed to handle.	Inform responsible individual for maintaining system.	BAAML
TAF FUNCTION CODE NOT VALID.	TAF cannot process the request issued because it is an unrecognizable function code. This is an internal error.	Inform data base administrator.	BAAML
TAF IDLE.	Informative message. This message appears only on the B display when TAF is idle.	None.	TAF
TAF IDLE (GO OR DROP).	Access to TAF denied because it is idling down.	Operator: Type GO,jsn. or drop job with DROP,jsn. Others: Informative message only.	BAAML
TAF IDLING DOWN.	TAF is idling down only EX.LOGT will be accepted by TAF.	Log off by typing EX.LOGT.	ITASK
TAF INTERNAL ERROR.	TAF has found internal data to be inconsistent.	Perform a dump of TAF or inform site analyst.	TAF
TAF NOT PRESENT (GO OR DROP).	The TAF subsystem was not present when a batch job tried to connect to it.	Operator: Bring up TAF and type GO,jsn. or drop job with DROP,jsn. Others: Informative message only.	BAAML
TAF RECOVERY REQUEST ERROR.	TAF cannot recover a batch user because of an internal TSTAT error.	Inform data base administrator.	BAAML
TAF SUBSYSTEM NOT DEFINED AS A SCP	TAF was not defined as a system control point when it was brought up.	Inform data base administrator.	BAAML
TAF TERMINATE.	The transaction subsystem was dropped via DSD command IDLE,TAF and was not restarted.	None.	TAF TAF
TAF TRANSACTION NAME UNKNOWN.	The transaction name requested for scheduling was not found in either the data base or system task library in TAF.	Enter correct transaction name or define the unknown transaction name in the task library or check to see whether nonexistent tasks are defined in the transaction.	MSABT
TAF TRANSACTION NOT RERUNNABLE.	There was an attempt to rerun a transaction and an error occurred.	Inform the data administrator. The administrator may check the TAF dayfile for the possible cause of the error.	RTASK
TAF TWO OUTSTANDING REQUESTS.	TAF received another request from a user program before the previous request was satisfied. This is an internal error.	Inform data base administrator.	BAAML
TAF USER NOT VALID FOR TAF ACCESS.	The batch job tried to access TAF (via a BTRAN) and the user name under which the job was run was not validated in the network file of TAF.	Inform TAF data administrator to enter the user name in the network file of TAF.	BAAML TAF
TAF USER NUMBER ACTIVE.	Another batch or terminal job is currently accessing TAF/CRM under this user name.	Resubmit job at a later time.	BAAML
TAPE NOT READABLE.	On a record load of a file the dump tape was found incomplete - no trailer record.	Load from previous dump.	DMREC
TASK CEASE WITH REQUESTED ABORTED.	The task requested on abnormal CEASE with or without memory dump.	Inform data base administrator.	MSABT
TASK EDITING COMPLETE.	Dayfile message indicating that the run has terminated. Normal LIBTASK output consists of four lists - the input directives, the tasks on the current library, the tasks added to the library, and the tasks replaced on the library. After these lists, the time and date of the last library modification are given. If errors occurred during processing of the task binary file, a list of error messages also appears. If the LIBTASK run was initiated from a terminal, the directive and current library lists do not appear.	Refer to next dayfile message.	LIBTASK
TASK LIBRARY DIRECTORY EMPTY.	The last record of the library file is empty.	Ensure that the correct file has been specified; if so, the library must be recreated.	LIBTASK

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
TASK LIBRARY DIRECTORY EMPTY - Libraryname.	The file specified as the task library contains no recognizable directory. TAF aborts.	Inform site analyst. Task library libraryname must be corrected and TAF reinitialized.	TAF
TASK LIBRARY DIRECTORY ERROR - Libraryname.	Task library libraryname contains no recognizable directory. TAF aborts.	Inform site analyst. Task library must be corrected.	TAF
TASK LIBRARY DIRECTORY ERRORS.	The first word of the last record on the library file does not contain the directory header.	Correct and rerun.	LIBTASK
TASK LIBRARY DIRECTORY TOO LONG.	The directory is too long for the buffer allocated.	This may be altered by changing an assembly constant.	LIBTASK
TASK LIBRARY DIRECTORY TOO LONG - Libraryname.	The length of the indicated task library directory exceeds the limit specified by the installation parameter TLDMT.	Size of task library must be reduced or TAF and LIBTASK must be reassembled with TLDMT increased.	TAF
TASK NOT VALIDATED FOR REQUEST.	One of the following actions has occurred. - The terminal operator initiated a TAF transaction which tried to perform an action associated with a data base for which the terminal was not validated. - A NEWTRAN request was issued by a task not in the system task library (TASKLIB).	Perform the appropriate action. - Inform the data administrator. Set up the terminal name in the network file to use the data base. The system data base (SY) may be used. - Put the task on TASKLIB.	MSABT
taskname - TASK RECOVERED.	The specified task has been recovered.	None.	AAMI
taskname - TASK RECOVERY FAILED.	The specified task recovery attempt has failed.	Inform data administrator.	AAMI
TASK REQUEST ARGUMENT ERROR.	A parameter error occurred in a request.	Correct task or inform data base administrator.	MSABT
TASK REQUESTED CEASE BEFORE DM FINISHED.	The task requested to end before the data manager completed all outstanding requests.	Correct task or inform data base administrator.	MSABT
TASK TIME LIMIT.	The task exceeded n time slices. The length of the time slice is set by an installation parameter. The number of time slices, n, allowed for the execution of a task can be changed with a task ITL request.	Increase time slices or inform data base administrator.	MSABT
nnn TASKS NOT LOADED INTO ECS.	An insufficient amount of extended memory was available to load all tasks. The nnn field is the number of tasks not loaded.	Check extended memory requested and reinitialize with more extended memory if appropriate.	TAF
TERMINAL DISABLED BY NOP.	The network operator has disabled your terminal. No further input is possible.	None.	CCP
TERMINAL NOT AVAILABLE.	An interactive user has attempted to login with recovery outstanding on a batch concurrency or BTRAN user	Recover the batch concurrency or BTRAN user.	RTASK
TERMINAL TIMEOUT, OPTION B ASSUMED.	Self-explanatory.	None.	RTASK
TERMINATION IN PROGRESS.	The interactive subsystem has begun dump/disconnect/recovery procedures due to an abort or termination condition.	None.	TAF IAFEX
TLD OVERFLOW.	Space is reserved at initialization to dynamically add tasks to the TAF resident copy of each task directory. The number of tasks is limited (default is 10) by the installation parameter TLDL. This message indicates that more tasks than allowed have been added since the most recent TAF initialization. The extra tasks will be ignored by TAF until it is reinitialized.	None.	TAF
TN REQUIRES EITHER LO=T OR LO=X.	The TN parameter was specified on the LIBTASK command without a corresponding LO=T or LO=X parameter.	Specify correct combination of list option parameters.	LIBTASK
TOO MANY BRANCHES IN TASK CHAIN REQUESTED.	Only three branches are allowed in the task chain; more than three were requested.	Correct task or inform data base administrator.	MSABT

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
TOO MANY DATA BASE NAMES.	The number of data base names associated with one data manager via DMS statements exceeds the value of MAXDB.	Decrease the number of data base names associated with the data manager.	TAF
TOO MANY FILE NAMES OR VSN-S SPECIFIED.	The VSN or file name limit was exceeded on a directive.	Check the installation parameters TVSNL and TDFNL for maximums.	DMREC
TOO MANY FILES IN TOTAL DATA BASE.	Self-explanatory.	Reduce the number of entries in the TCF file or increase TMAXFIL.	TAF
TOO MANY NEW TRANSACTIONS ON TRANSACTION DIRECTORY - filename.	Informative message. During an on line update of a task library, TAF detected that the transaction directory (TRD) was longer than allowed (the number of new additions exceeds installation parameter TRDMN). Excess named TAF transaction definitions will be discarded.	None. When TAF is next initialized, the new transaction definitions will be included.	TAF
TOO MANY RA+1 CALLS.	Only n monitor requests are allowed; more than n occurred, where n is an assembly constant.	Inform site analyst. Possibly a task is in a loop.	MSABT
TOO MANY TAF JOURNAL FILES IN xxJ FILE.	More than three TAF journal files per data base were specified, causing the transaction subsystem to abort.	Examine xxJ file for xxJOR entries. Inform the TAF data administrator.	TAF
TOTAL DATA MANAGER NOT LOADED.	TOTAL data manager was called but has not been loaded.	Inform data base administrator.	MSABT
TOTAL DATA MANAGER SUCCESSFULLY LOADED.	Self-explanatory.	None.	TAF
TOTAL DID NOT RECOVER PROPERLY STATUS IS yyyy.	An error status yyyy was returned on a TOTAL FINAL call. Refer to Diagnostics in the TOTAL Reference Manual for yyyy.	Correct error and reinitialize transaction executive.	TAF TAF
TRAN. SEQ. zzzzzzzzB with ID xxxxx MIGHT BE BAD.	A BRf disk error has occurred and the transaction sequence numbers listed in the dayfile might contain errors.	Inform data base administrator of the error.	DMREC REC
TRANSACTION DIRECTORY EMPTY.	Probable corruption of task library file.	Regenerate task library.	LIBTASK
TRANSACTION DIRECTORY EMPTY - filename.	Probable corruption of task library file.	Regenerate task library.	TAF
TRANSACTION DIRECTORY ERROR.	Probable corruption of task library file.	Regenerate task library.	LIBTASK
TRANSACTION DIRECTORY ERROR - filename.	Probable corruption of task library file.	Regenerate task library.	TAF
TRANSACTION DIRECTORY TOO LONG - filename.	Error -- Too many transaction definitions exist in the transaction directory.	Regenerate task library. Make sure the number of transactions does not exceed installation parameter TRDMT or reassemble TAF and LIBTASK increasing the size of the installation parameter TRDMT.	LIBTASK
TRANSACTION DIRECTORY TOO LONG.	Too many transactions exist in named-transaction directory.	Either reduce number of named-transactions declared or contact TAF site analyst.	LIBTASK
TRANSACTION SESSION ENDED.	Option B was selected and TAF logged the terminal off leaving the previous recoverable informaton intact.	Log in at a later time to attempt rerun.	RTASK
TT OPTION REQUIRES USER NUMBER.	When updating a task library on-line (TT option is specified on LIBTASK command), the user name must be specified prior to the LIBTASK command so the library associated with that user name can be found.	Specify user name via USER or CHARGE command before LIBTASK command and rerun job.	LIBTASK
UN=username NOT VALID ON FM=family.	Username on the specified familyname is not valid. The user name and familyname may not be defined or are incorrect in xxJ file.	Inform site analyst.	TAF
UNABLE TO ATTACH TOTAL BINARIES.	File of TOTAL binaries is not under the user index of the transaction subsystem or a PFM error occurred.	Correct error and reinitialize transaction executive, or inform site analyst.	TAF

MESSAGE	SIGNIFICANCE	ACTION	ROUTINE
UNABLE TO ATTACH TOTAL DEMOD BINARIES.	One or more of the DEMOD files listed on the TOTAL DMS statement in the TCF file could not be attached under the user name of the transaction subsystem or a PFM error occurred.	Correct error and reinitialize transaction executive, or inform site analyst.	TAF
UNKNOWN FILE FORMAT.	There is a logical error in the structure of the input file. It does not conform to the established format rules.	None.	KTSOMP
UNKNOWN NAMED TRANSACTION UNIT REQUESTED FOR SCHEDULING.	The transaction name requested for scheduling was incorrect.	Enter the correct transaction name.	ITASK
UNRECOGNIZABLE CIO ERROR CODE.	Self-explanatory.	Inform site analyst.	TAF
UNRECOGNIZABLE HEADER TYPE.	When interpreting ARF after image header types, one has been found incorrect.	Inform data administrator.	DMREC
USER STATEMENT DOES NOT APPEAR IN TCF.	Self-explanatory.	Add a USER statement to the TCF.	TAFREC
USER STATEMENT NOT VALID ON FAMILY.	The user name specified by the USER statement is not valid on the family.	Validate user on specified family or correct USER statement.	TAFREC
VALID LO OPTIONS ARE LO, LO=A, LO=T or LO=X.	The LO parameter was specified on the LIBTASK command with an incorrect list option.	Specify one of the following valid list options LO, LO=A, LO=T or LO=X.	LIBTASK
VALID OPTIONS ARE A, B, OR C.	The option entered was not A, B, or C.	Reenter a valid option.	RTASK
cmd VALUE INAPPROPRIATE	The value you specified in the terminal definition command cmd is not appropriate for your terminal.	Choose an appropriate value (refer to appendix J).	CCP
VSN ALREADY EXISTS.	When attempting to use the edit/add directives, the VSN specified was found to already exist on the directory.	Continue processing.	DMREC
VSN - xxxxxx ALREADY IN DIRECTORY.	The VSN supplied for this operation was found to already exist on the directory.	Try another VSN.	DMREC
VSN AND DATE/TIME CANNOT CO-EXIST ON LOAD DIRECTIVE.	On a load directive the date/time keywords cannot be used with VSN keyword.	Correct the directive and rerun.	DMREC
VSN ASSIGNED DOES NOT MATCH VSN REQUESTED.	The VSN assigned as a result of a DMREC ADD subdirective does not agree with the VSN specified on the subdirective.	Correct the VSN conflict.	DMREC
VSN DOES NOT EXIST.	When attempting to access a directory, the specified VSN was not found on the directory.	List the directory for further information.	DMREC
VSN IS NOT FIRST REEL.	When attempting to modify a directory, the specified VSN was not the first reel of the set.	Specify the first reel of the multireel set.	DMREC
VSN OR DATE/TIME NOT SPECIFIED.	When attempting to delete directory entries, no VSN or date/time was specified.	Specify VSN or date/time and rerun.	DMREC
VSN TABLE OVERFLOW.	Too many VSN entries were encountered for this directive.	Check directive and increase TVSN size if necessary.	DMREC (BVT)
WAIT..	The network is unable to accept any further data from your terminal. Data that was entered is discarded. This is normally a temporary situation caused by buffer shortages in the host or the network.	Wait for the REPEAT.. message.	CCP
WAITING	Option A was selected and TAF will wait 5 minutes before attempting a rerun of the transaction.	None.	RTASK
WAITINP FROM MULTI QUEUED TASK.	A WAITINP request was issued by a task that was scheduled to process more than one TAF transaction or by a task that was called with return.	Correct task or inform data base administrator.	MSABT
WRONG ENTRY WHILE READING VSN ENTRIES.	The VSN record read from the directory has an incorrect format.	Inform data administrator.	DMREC
WRONG VSN USED.	Either the wrong VSN was used or no header was found on ARF.	Make sure the correct ARF is being used and retry.	DMREC
ZZdbAnn - ACTIVE ARF.	File ZZdbAnn is the active image recovery file for data base db.	Informative message.	AAMI

MESSAGE

SIGNIFICANCE

ACTION

ROUTINE

ZZdbBnn - BRf RECOVERY
FILE.

TAF/CRM is beginning to roll back data base
updates from file ZZdbBnn.

File name entry can be
restored by adding an
appropriate CRM
statement to the xxJ
file.

AAMI

ZZdbDIR UNREADABLE.

When trying to expand a data file, an
attempt to read the directory file failed.

Inform data
administrator.

DMREC

Faint, illegible text at the top of the page, possibly a header or title area.



AAM

Refer to Advanced Access Methods.

ABH

Refer to Application Block Header.

ABN

Refer to Application Block Number.

Absolute code

Code in which all required external references have been satisfied. All addresses refer to specific locations within the job field length.

ABT

Refer to Application Block Type.

ACN

Refer to Application Connection Number.

ACT

Refer to Application Character Type.

Actual Key

The primary key for a record in a file with actual key organization; indicates the storage location of the record.

Actual Key (AK) File

A mass storage file in which each record is stored at the location indicated by the primary key. For initial actual key files, the primary key must specify the block and record slot number in which the record is stored. For extended actual key files, the primary key is a record number that AAM converts to the storage location of the record. Access is random or sequential.

Address

The location of a word in memory. The location is designated by number or symbolic name.

Advanced Access Methods (AAM)

A file manager that processes indexed sequential, direct access, and actual key file organizations, and supports the Multiple Index Processor. Refer to CYBER Record Manager.

After Image

A copy of a DMS-170 data base record after it has been modified.

After Image Recovery File (ARF)

A file upon which TAF writes after images of records in recoverable files when a request to update such a record is processed within a begin-commit sequence.

AK File

Refer to Actual Key File.

Alias

A data name used in a subschema in place of a schema data name.

Alphabetic Character

A character belonging to the set of letters A through Z and the space.

Alphanumeric

Consisting of alphabetic and/or numeric characters only.

Alternate Key

A key, other than the primary key, by which an indexed sequential, direct access, or actual key file can be accessed.

ANSI

American National Standards Institute. An organization that establishes standards for the benefit of its member organizations.

Application

In TAF, a set of data files and the set of tasks associated with those files.

Application

A program resident in a host computer that provides an information storage, retrieval, and/or processing service to a remote user via the data communication network. IAF is an example of an application.

Application Block Header (ABH)

A single 60-bit word description accompanying every block passing between an application and NAM.

Application Block Number (ABN)

A field in the application block header. An application-assigned number used to identify a particular data message block.

Application Block Type (ABT)

A field in the application block header defining the accompanying block as either data or supervisory, null or not null, and indicating if this is the last block of a message.

Application Character Type (ACT)

A field in the application block header defining the byte size of text characters.

Application Connection Number (ACN)

A number assigned by the communications supervisor program to identify a particular logical connection within an application.

Application Switching

The process of leaving the control of one application and entering the control of another, without going through another complete login sequence.

Area

A uniquely named DMS-170 data base subdivision that contains data records; a file.

ARF

Refer to After Image Recovery File.

ASCII

American National Standard Code for Information Interchange. The standard character set and code used for information interchange between systems.

ASCII Mode

Use of the American National Standard Code for Information Interchange; a 128-character set. It includes both uppercase and lowercase letters.

Assemble

To transform a COMPASS language program into a form that the computer can execute directly. During assembly, machine language operation codes are substituted for COMPASS codes and machine addresses for symbolic addresses. COMPASS performs the assembly.

Attach

The process of making a permanent file accessible to a job by specifying the proper permanent file identification and passwords.

Auto Mode

Mode of entering information into a primary file where the system automatically generates line numbers. This allows you to correct, insert, and delete lines easily because you can reference each line by its line number.

Automatic Login

The process whereby one or more of the Network Validation Facility login dialog parameters is

supplied to NVF from the local configuration file. Parameters supplied through automatic login configuration of a terminal suppress prompting for the corresponding dialog entries and override any entries made from the terminal. Automatic login is required for terminals (passive devices) incapable of conducting dialog with NVF.

Backup Directory

A direct access file automatically allocated and maintained by DMREC. It contains information about the data base, index and after image recovery files for an application. The file name is ZZxxDIR where xx is the application identifier.

Backup Dump

A copy of all or selected portions of a data base, which is produced on a regularly scheduled basis for the explicit purpose of data base recovery.

Batch Job

Instructions and data submitted as a complete unit without further user intervention. The job can be punched on cards or created and submitted from a terminal.

Batch Mode

The state of a mode 4A terminal during which batch data is transmitted from the terminal's card reader to central files and/or from central files to the terminal's line printer. Also, the state of an asynchronous terminal during which input transmission occurs on a block-by-block basis.

Batch Test Facility

An absolute program residing on the CDCS system library that allows CDCS to run at the same control point as a user program.

Before Image

A copy of a DMS-170 data base record before it has been modified.

Before Image Recovery File (BRF)

A file to which TAF/CRM writes the before images of records in recoverable files when a request to read such a record is processed from within a begin-commit sequence.

Begin-commit sequence

A series of data base updates beginning with a data base begin request and ending with a data base commit or data base free request.

Beginning-of-Information (BOI)

The start of the first user record in a file. Tape labels and system-supplied information, such as an index block or control word, do not affect beginning-of-information.

Binary File

Usually a noneditable file containing a pre-compiled program.

Bit

An abbreviation of binary digit. It is a single digit, 0 or 1, in a binary number. Also used to represent the smallest unit of information. A central memory word (one storage location) contains 60 bits.

Block

In the context of network communications, a portion or all of a message. A message is divided into blocks to facilitate buffering, transmission, error detection, and correction for variable length data streams.

During input from a terminal, a block is a single transmission comprising one or more lines of one or more messages.

During input to an application, a block is a single line comprising part or all of a message. Terminal transmission blocks are divided into as many application input blocks as needed to provide one block of input per terminal input line, until the message is completed.

During output to a terminal, a block is one or more lines comprising one message.

BOI

Refer to Beginning-of-Information.

BP

Beginning Priority.

BRF

Refer to Before Image Recovery File.

Called Task

A task which is the object of a CALLTSK or a CALLRTN request.

Calling Task

A task which executes a CALLSTK or a CALLRTN request.

Carriage Control

The control exercised over the format of printed output. The leftmost character of the data line to be printed is the carriage control character. Carriage control is also called format control.

CB

Communication Block.

CCP

Communications Control Program.

CDCS

Refer to CYBER Database Control System.

Central Memory (CM)

The main storage device whose storage cells (words) can be addressed by a computer program and from which instructions and data can be loaded directly into registers from which the instructions can be executed or from which data can be manipulated.

Character

Any alphabetic, numeric, or special symbol that can be encoded. This term applies to the graphic characters for an input or output device, and to uniquely encoded control characters used by a terminal. Unless specified otherwise, characters are ASCII 8-bit byte characters.

Charge Number

An alphanumeric identifier the installation uses to allocate charges to individual users for system usage.

CIO

Circular Input/Output.

Close

A set of terminating operations performed on a file when input and output operations are complete. All files processed by the CYBER Record Manager must be closed.

CM

Refer to Central Memory.

CMM

Common Memory Manager.

COBOL

Common Business-Oriented Language. This higher level language simplifies the programming of business data applications.

Command

A sequence of words and characters that call a system routine to perform a job step. The command must conform to format specifications and end with either a period or a right parenthesis. A command is sometimes called a control statement.

Communication Recovery File (CRF)

Files used for the automatic recovery feature.

Compile

To translate a program from a higher level programming language (for example, FORTRAN or BASIC) into machine instructions called object code.

Concurrency

Simultaneous access to the same data in a data base by two or more applications during a given span of time.

Console

A terminal devoted to network control processing. There may be three such terminals: the network operator's (NOP) terminal, the local operator's (LOP) terminal, and the network processing unit console.

CRF

Refer to Communication Recovery File.

CRM

Refer to CYBER Record Manager.

CRT

Cathode Ray Tube.

CYBER Database Control System (CDCS)

The DMS-170 controlling module that provides the interface between an application and a data base.

CYBER Loader

The utility that prepares programs for execution by placing program instruction and data blocks in central memory.

CYBER Record Manager (CRM)

A software product that allows a variety of record types, blocking types, and file organizations to be created and accessed. Products like COBOL 5, FORTRAN Extended 4, FORTRAN 5, Sort/Merge 4, Sort/Merge 5, ALGOL 5, and DMS 170 use the CYBER Record Manager to manage execution time input/output. Neither the input/output of the operating system nor that of most of the system utilities such as COPY or SKIPF is implemented through the CYBER Record Manager. All CYBER Record Manager file processing requests ultimately pass through the operating system input/output routines.

DA

Refer to Data Administrator.

Data Administrator (DA)

The individual who leads the design, programming, implementation, and maintenance efforts associated with a DMS-170 controlled data base.

Data Base

A systematically organized, central pool of information; organization is described by a specific schema.

Data Base Status Block

An area of memory defined within an application to which CDCS returns information concerning the status of operations on data base files and relations. The status block is updated after each CDCS operation.

Data Block

A block in which user records are stored in an indexed sequential or actual key file. Data block structure is defined by the user, or AAM defaults are accepted. Contrast with Index Block for indexed sequential files.

Data Description Language (DDL)

The language used to structure a DMS-170 schema and subschema.

Data Management System

Refer to Database Management System.

Database Management System

A generalized computer program which handles the mechanics of storing, updating, and accessing data for multiple applications.

Dayfile

A chronological file created during job execution which forms a permanent accounting and job history record. Dayfile messages are generated by operator action or when commands are processed. A copy of the dayfile is printed with the output for each job. You must explicitly request it in an interactive job.

DBREC

Utility used to prepare files for transaction recovery using CDCS.

DDL

Refer to Data Description Language.

Deadlock

A situation that arises in concurrent data base access when two or more applications are contending for a resource that is locked by one of the other ones, and none of the programs can proceed without that resource.

Deadstart

The process of initializing the system by loading the operating system library programs and any of the product set from magnetic tape or disk. Deadstart recovery is reinitialization after system failure.

Default

A system-supplied option used when you do not supply the option.

Delimiter

A character used to separate statement elements, such as words and literal constants, or other strings of text.

Device

The physical recording medium of random mass storage, such as a disk pack.

Device Set

A group of rotating mass storage devices. No device can belong to more than one device set. Every file must be contained within one device set, but can be on different devices in that device set.

Direct Access File

A CYBER Record Manager file containing records stored randomly in home blocks according to the hashed value of the primary key in each record. Files must be mass storage-resident. All allocation for home blocks occurs when the file is opened on its creation run. Access is random or sequential. It may be attached to your job and all changes are made on the file itself rather than a temporary copy of the file.

Directives

The instructions that supplement processing defined by a command or by a program call for execution of a utility function or member of a product set. Directives do not appear in the command record; they are usually in a separate record of the file INPUT or a file referenced in a command. Directives are required for execution of FORM, the CREATE utility, and EDITLIB, among others.

Disk

A unit composed of one or more flat, circular plates with magnetic material on both sides that is used to store large amounts of data or programs.

Display Code

A 6-bit-display code set used to represent alphanumeric and special characters.

Displays

Two console screens or a split screen used to display system and job information, operator messages, and contents of central memory. Through the console keyboard, the operator can control the operation of the system. The displays are identified by alphabetic characters; some used frequently are: job status (B), system files (H), and dayfile messages (A).

Dump

A printed listing of the contents of the central processor registers and a predetermined number of words within central memory that pertain to a job.

Dynamic Access

Access mode that allows a nonsequential file on mass storage to be accessed randomly or sequentially, depending on the format of the access statement.

ECS

Refer to Extended Core Storage.

End-of-File (EOF)

A boundary within a sequential file, but not necessarily the end of a file, that can be referenced by name. The actual end of a named file is defined by EOI. For labeled tape, EOF and EOI (denoted by the EOF1 label) are the same. For multifile tape files, EOF and EOI do not correspond. In the product set manuals, an end-of-file is also referred to as an end-of-partition.

End-of-Information (EOI)

An indicator that marks the physical end of a named file.

End-of-Record (EOR)

The terminator of a logical record. On a PRU device, a short PRU or a zero-length PRU with a level designator of 0 indicates EOR. On tapes that are not PRU devices, an interrecord gap indicates EOR.

EOF

Refer to End-of-File.

EOI

Refer to End-of-Information.

EOR

Refer to End-of-Record.

ESM

Extended Semiconductor Memory.

Exchange Package

A table that contains information used during job execution. It is printed as part of the output when a job aborts.

Extended

A term used in conjunction with indexed sequential, direct access, and actual key files to denote a specific type of internal processing by AAM and MIP. Processing is indicated by setting the ORG FIT field to NEW.

Extended Core Storage (ECS)

Optional additional memory. ECS contains 60-bit words; it has a large amount of storage and fast transfer rates. ECS can be used only for program and data storage, not for program execution. Special hardware instructions exist for transferring data between central memory and ECS.

Extended Memory

An extension to central memory which is physically located outside of the machine. Formerly referred to as extended core storage (ECS) or large central memory (LCM).

Family Name

Name of the permanent file storage device or set of devices on which all of your permanent files are stored. When you request a permanent file, the system looks for it on this family (group) of devices. Usually a system has only one family of permanent file devices but it is possible to have alternate families in the system. You specify which family you are using when you log in. Your family name is given to you by your employer, instructor, or computer personnel.

Fast Dynamic Loader (FDL)

A facility that provides fast loading and unloading of specially formatted code called capsules. The amount of memory required for job execution can be greatly reduced because capsules can be easily loaded and unloaded as needed, freeing memory for other uses.

FDL

Refer to Fast Dynamic Loader.

FE

Refer to Format Effector.

FET

Refer to File Environment Table.

Field Length (FL)

The area in central memory allocated to a particular job; the only part of central memory that a job can directly access. Also the number of central memory words required to process a job.

File

A collection of information referred to by a file name (from one through seven characters, either letters or numbers). You can create a file at the terminal or retrieve a file from permanent file storage for use during a terminal session.

FILE Command

A command that supplies file information table values after a source language program is compiled or assembled but before the program is executed. Basic file characteristics such as organization, record type, and description can be specified in the FILE command.

File Environment Table (FET)

A table within a program's field length through which the program communicates with operating system input/output routines. One FET exists for each file in use by the program.

File Information Table (FIT)

A table through which a user program communicates with the CYBER Record Manager. For direct processing through CRM, a user must initiate establishment of this table. All file processing executes on the basis of information in this table. The user can set FIT fields directly or use parameters in a file access call that sets the fields indirectly. Some product set members set the fields automatically for the user.

File Organization

Defined by ORGANIZATION clause. Can be sequential, indexed, relative, direct, actual-key, or word-address. Established at the time the file is created and cannot change as long as the file exists. Affects access mode, open mode, and formats of statements that can be used to manipulate file records.

File Statement

Command for the CYBER Record Manager that specifies parameters for the FIT at the time the file is opened.

File Statistics Table (FSTT)

A table generated and maintained by the CYBER Record Manager to collect statistics about each direct access, indexed sequential, and actual key file. The FSTT is a permanent part of a file and contains information such as organization type, size of blocks and the number of current accesses.

FIT

Refer to File Information Table.

FL

Refer to Field Length.

Flag

A character or bit that signals the occurrence or presence of a particular condition.

FORM

A general purpose file management utility for manipulating records and creating and converting files.

Format Effector (FE)

A control symbol used by certain protocols (for instance, the HASP protocol).

FORTRAN

Formula TRANslation. A language that solves algebraic and scientific problems using symbols and statements that closely resemble mathematical notation.

FSE

Refer to Full Screen Editor.

FSTT

Refer to File Statistics Table.

Full Screen Editor

The editing facility that enables you to edit files in a screen format or a line-by-line format.

GTDL

Global Task Dump Limit.

HAD

Host availability display.

Hashing

The method of using primary keys to search for relative home block addresses of records in a file with direct access storage structure.

Header

A word or set of words at the beginning of a block, record, file, or buffer which contains control information for that unit of data.

Host

A computer that executes applications.

IAF

Refer to Interactive Access Facility.

Index

In the context of AAM, a series of keys and pointers to records associated with the keys. The system creates an index for AAM files which relates alternate record keys to primary keys, using the file defined by the second implementor-name of an ASSIGN clause.

In general context, a computer storage area or register, the content of which represents the identification of a particular element.

Index Block

For an indexed sequential file, a block with ordered keys and pointers to data blocks and other index blocks, forming a directory of the records within a file.

Indexed Sequential (IS) File

A file organization in which the CYBER Record Manager maintains records in sorted order by use of a programmer-defined key, which need not be within the record. Keys may be integer, floating point, or symbolic; access is random or sequential. Files contain index blocks and data blocks.

Indirect Access File

A NOS permanent file that you access by making a temporary copy of the file (GET or OLD command). You create or alter it by saving or substituting the contents of an existing temporary file (REPLACE or SAVE command).

Input

Information flowing upline from the terminal to the host computer.

Interactive

A mode of job processing in which the user enters a command or input to an executing program and receives an immediate response.

Interactive Access Facility (IAF)

An application that provides a terminal operator with the interactive processing capability. The interactive facility makes terminal input/output and file input/output appear the same to an executing program.

I/O

Input/Output.

IS File

Refer to Indexed Sequential File.

Itemize

A utility routine that produces a listing of the contents of a file or library.

IVT

Interactive Virtual Terminal.

Job

All computer-related activity associated with a terminal session from login to logout. A batch job consists of instructions and data submitted as a complete unit.

Key

A group of contiguous characters or numbers the user defines to identify a record in a GRM file.

LCME

Large Central Memory Extended.

LDSET

The loader directive used to control the load process under a variety of conditions.

LGO

The default name of the file to which language processors write executable code during program assembly or compilation.

Load Sequence

A sequence of load operations which encompasses all of the loader's processing from the time that nothing is loaded until the time execution begins. It includes initialization, specification of specified loader requests, and the completion of the load.

Loader

A software product that prepares programs for execution by placing program instructions and data blocks in central memory and linking references in the program to the appropriate external routines.

Local

Refers to data that exists only during the processing of a single job and that can only be accessed by that job.

Local Operator (LOP)

The network operator who monitors line, terminal, and application activities in the network.

Login

The procedure used at an interactive terminal to gain access to the system.

Logout

The procedure by which you end terminal session. You type BYE.

LOP

Refer to Local Operator.

Macro

A sequence of source commands that are saved and then assembled whenever needed through a macro call.

Mass Storage

Magnetic disk or extended memory that can be accessed randomly as well as sequentially.

Master Directory

A file containing information used by CDCS in processing. This information consists of schema and subschema tables, media parameters, and data base procedure library and logging specifications.

MFL

Maximum Field Length.

MIP

Refer to Multiple Index Processor.

MP

Maximum Priority.

Multifile Set

A tape file set having more than one tape file.

Multimainframe Operation

Operation that provides mechanisms by which more than one computer can share mass storage devices.

Multiple Index File

An indexed sequential, direct access, or actual key file that has alternate keys defined.

Multiple Index Processor (MIP)

A processor that allows AAM files to be accessed by alternate keys.

NAM

Refer to Network Access Method.

NCT

Network Communication Table.

NCTF

Refer to Network Description File.

NDL

Refer to Network Definition Language.

Network Access Method (NAM)

A software package that provides a generalized method of using a communications network for switching, buffering, queuing, and transmitting data. NAM is a set of interface routines used by a terminal servicing facility for shared access to a network of terminals and other applications, so that the facility program does not need to support the physical structures and protocols of a private communication network.

Network Definition Language (NDL)

The compiler-level language used to define the network configuration file and local configuration file contents.

Network Description File (NCTF)

File that must be present if the transaction facility is used. The file is prepared by the site analyst.

NOS

Network Operating System.

Numeric Character

Digits 0 through 9.

Object Code

The machine language version of a program that has been translated (compiled) from source code written in a higher level language.

Open

A set of preparatory operations performed on a file before input and output can take place; required for all CRM files.

Operating System

The set of system programs that controls the execution of computer programs and provides scheduling, error detection, input/output control, accounting, compilation, storage assignment, and other related services.

Operation

A particular function performed on units of data; for instance, opening or closing an area, or storing or deleting a record.

Order Dependent

Used to describe items which must appear in a specific order.

Order Independent

Used to describe items which need not appear in any specific order. Parameters, particularly those with keywords, may be order independent.

Output

Information flowing downline from host to terminal.

Output File

The file on which the system writes information to the user. Unless another file name is specified, the output file is printed at the terminal.

Pack Name

A one- to seven-character name that identifies the auxiliary device to be accessed in a permanent file request.

Parameter

A variable that is given a specific value for a particular purpose or process.

Parity

In writing data, an extra bit is either set or cleared in each byte so that every byte has either an odd number of set bits (odd parity) or an even number of set bits (even parity). Parity is checked on a read for error detection and possible recovery.

Password

A name or word entered during login to provide extra security for your user name. A unique password ensures that no one else can log into the system using your user name and access your files.

Permanent File

A file stored on mass storage. This file is cataloged by the system so that its location and identification are always known to the system. Permanent files cannot be destroyed accidentally during normal system operation. They are protected by the system from unauthorized access according to privacy controls specified by the file owner.

Physical Record Unit (PRU)

The amount of information transmitted by a single physical operation of a specified device. For mass storage files, a PRU is 64 central memory words (640 characters); for magnetic tape files, the size of the PRU depends upon the tape format. A PRU that is not full of user data is called a short PRU; a PRU that has a level terminator but no user data is called a zero-length PRU.

Primary Key

A key whose value uniquely identifies a record and determines the location of the record in a file. The primary key for a file must be defined when the file is created and must be used to update the file. Contrast with Alternate Key.

Procedure

A user-defined set of instructions that are referenced by name. The instructions consist of procedure directives and system commands.

Procedure File

A file containing one or more procedures.

PRU

Refer to Physical Record Unit.

PSN

Packet-Switching Network.

Purge

To delete a permanent file from the system. This enables releasing its mass storage space, erasing its catalog entries, and so forth.

Queue

A sequence of blocks, tables, messages, and so forth. Most NPU queues are maintained by leaving the queued elements in place and using tables of pointers to the next queued element. Most queues operate on a first-in, first-out basis. A series of worklist entries for a specific terminal is an example of an NPU queue.

RA

Refer to Reference Address.

RAE

Refer to Reference Address.

Record

A unit of information. In the CYBER Record Manager and its language processors, a record is a unit of information produced by a single read or write request. Eight different record types exist within CRM. The user defines the structure and characteristics of records within a file by declaring a record format.

Other parts of the operating system and its products might have additional or different definitions of records.

Record Slot Number

The position of a record within a block in an actual key file; specified by the low-order bits of the primary key.

Record Type

The term record type can have one of several meanings, depending on the context. The CYBER Record Manager defines eight record types established by an RT field in the file information table. Tables output by the loader are classified as record types such as text, relocatable, or absolute, depending on the first few words of the tables.

Recovery

A process that makes a data base useful after some type of software or hardware failure has occurred.

Reference Address (RA and RAE)

RA is the absolute central memory address that is the starting, or zero, relative address assigned to a program. Addresses within the program are relative to RA. RA+1 is used as the communication word between the user program and monitor. RAE is the absolute extended memory starting address assigned to the file.

Release System

A software system delivered to a customer. In installing a system, the customer, but not an individual applications programmer, can use default values or parameters that differ from the released system.

Relocatable Code

Code that can be loaded into any region of the job field length. External references have not yet been satisfied.

Request

A statement that identifies a function to be performed. Often there are items used by or resulting from the function that are required as parameters in the request.

Reserved Word

An identifier that has a predefined meaning to a compiler and must not appear in a source program as a user-defined word or name.

Restart Identifier File

Random permanent file used for a start identifier in recovery by CDCS.

Returned Value

The value returned from a procedure invoked as a function, or the value returned by a built-in function.

Rewind

An operation that positions a file at beginning-of-information.

RMS

Refer to Rotating Mass Storage.

Rollback

The process of restoring a data base to a previous state.

Rotating Mass Storage (RMS)

A disk storage device.

RTL

Requested Task List.

Satisfying External References

The process of searching one or more libraries and loading programs that contain entry points matching external references that are currently unsatisfied.

Schema

A detailed description of the internal structure of a data base.

Separator

A character used to separate parameters in a command.

Sequential (SQ) File

A file in which records are accessed in the order in which they occur. Any file can be accessed sequentially. Sequential files must be accessed sequentially because no key or address is associated with each record in the files.

Source Code

Code input to the computer for later translation into executable machine language instructions (object code).

SQ File

Refer to Sequential File.

Stimulator

A collection of central memory and peripheral processor programs which enters a hypothetical work load into the system to analyze the effects of such a load on response time and system reliability.

String

A sequence of contiguous characters or bits treated as a unit.

Subschema

A detailed description of the portion of a data base to be made available to one or more applications.

Supervisory Message

A message block in the host not directly involved with the transmission of data, but which provides information for establishing and maintaining an environment for the communication of data between the application and NAM, then through the network to a destination or from a source. Supervisory messages may be transmitted to an NPU in the format of a service message.

TAF Configuration File (TCF)

A file created and maintained by the data administrator specifying which data managers are active and the applications to be associated with each.

Task

A program to be executed under the transaction subsystem. Tasks are stored in absolute binary format on libraries which must be available to the transaction subsystem.

Task Library Directory (TLD)

The directory on each task library which contains information on the tasks in the library. TAF makes its own copy of each TLD at initialization and uses this copy to locate requested tasks.

TCF

Refer to TAF Configuration File.

Terminal

The equipment a person uses to communicate with the computer.

Terminal Class

An NDL parameter describing the physical attributes of a group of similar terminals, in terms of an archetype terminal for the group.

Terminal Session

Period between the time you physically connect the terminal to the system in preparation for login to the time you log out.

Text Mode

Mode of entering information into a primary file without specifying line numbers. It is usually used to create data files. If a file does not contain line numbers, you can change lines by using an editing facility such as the Full Screen Editor.

TLD

Refer to Task Library Directory.

Transaction

A task or a series of tasks executed as a unit under the transaction subsystem. All tasks in a transaction have the same transaction sequence number. In CDCS, the term data base transaction refers to the process which this manual calls a begin-commit sequence.

Transaction Library Directory (TRD)

The directory on each task library which contains information on the transactions in the library. TAF makes its own copy of each TRD at initialization and uses this copy to locate requested transactions.

Transaction Name

The one to seven alphanumeric character name given to a transaction created with the /tname input directive from the LIBTASK utility.

Transaction Recovery File

Random permanent file used for before image records necessary for recovery by CDCS.

Transaction Sequence Number

A unique number assigned by TAF to each transaction. It is used by TAF for task identification.

Transparent Mode

A software feature provided by the Network Access Method and the network processing unit (NPU). When transparent mode transmission occurs between an application and a terminal the Network Access Method does not convert data to or from 6-bit display code, and the NPU does not edit the character stream or convert the characters to or from 7-bit ASCII code. When no parity is in effect for the terminal and transparent mode transmission occurs, all seven bits of the character byte can be used to represent characters in character sets (such as EBCDIC).

TRD

Refer to Transaction Library Directory.

TST

Terminal Status Table.

TTOT

Table of Time-Originating Tasks.

UEM

Unified Extended Memory.

UJN

User Job Name.

User Index

A unique 17-bit identifier that is associated with each user name. The user index is used by the permanent file manager to identify the device and catalog track for the user's files.

User Name

A system access word that must be supplied by the user for validation purposes at login.

Utility

A program designed to help the user perform specific functions such as permanent file maintenance, library maintenance, and file editing.

Virtual Terminal

A software concept for CCP that converts all types of upline messages to one of two formats: batch virtual terminal or interactive virtual terminal. By this method, applications in the host need only to be able to process data in IVT or BVT format rather than in the multiplicity of formats which real terminals use. Downline messages from the host to real terminals are converted from IVT or BVT to real terminal format. The IVT/BVT processors are a part of the NPU's network communications software.

Volume Serial Number (VSN)

A one- to six-character identifier that identifies the volume of magnetic tape for the operator.

VSN

Refer to Volume Serial Number.

Word

A group of bits (or 6-bit characters) between boundaries imposed by the computer system. A word is 60 bits in length. The bits are numbered 59 through 0, starting from the left. It is also composed of five 12-bit bytes, numbered 0 through 4, from the left.

Working Storage Area

An area within the user's field length intended for receipt of data from a file or transmission of data to a file.

Write Mode

Allows a user to write, modify, append, read, execute, or purge the file (modify permission applies only to direct access files).

xxJ Files

Files created and maintained by the data administration which associates a particular user with a particular application.

Zero-Length PRU

A PRU that contains system information, but no user data. Under the CYBER Record Manager, a zero-length PRU level of 17 is a partition boundary. Under NOS, a zero-length PRU defines EOF.

The sample command list structures in this appendix demonstrate the use of various parameters that are available. The default input and output files can be used to reduce the number of control language statements needed to bring about a compile, load, and LIBTASK sequence, because the default binary output files are the binary input files for subsequent processors.

For example:

```
COBOLS,TAF,UC1.
LOAD,LGO.
NOGO,LGOB.
LIBTASK,P=RLTASKL,TT.
```

It is assumed that each of the following lists is preceded by job and USER commands. Some sites may require a CHARGE command.

COMPASS TASK COMMAND LIST STRUCTURE

A sample command list that assembles a COMPASS application program and loads it onto a task library, RLTASKL, is illustrated in figure D-1.

```
ATTACH,OPL/UN=username.
COMPASS.
LDSET,LIB=TRANLIB,LIB=TRANF4.
LOAD,LGO.
NOGO,LGOB.
LIBTASK,P=RLTASKL,B=LGOB TT I=0.
end-of-record
      IDENT  TASK
      ENTRY  TASK
OPL  XTEXT  COMKMAC
COMBL BSS 69
      . ← COMPASS task appears here
      .
      END
end-of-file
```

Figure D-1. COMPASS Command List Structure

NOTE

The system OPL must be attached for COMKMAC to be called in the COMPASS program. Consult site personnel for the appropriate command.

FORTRAN TASK COMMAND LIST STRUCTURE

A sample command list that compiles a FORTRAN application program and loads it onto a task library, RLTASKL, is illustrated in figure D-2.

```
FTN,DB=0.†
LDSET,LIB=TRANLIB,LIB=TRANF5.
LOAD,LGO.
NOGO,LGOB.
ATTACH,RLTASKL/M=W.
LIBTASK,P=RLTASKL,TT,I=0.
end-of-record
OVERLAY,ABS,0,0.
PROGRAM TASK2 ††
      .
      . ← FORTRAN task appears here
      .
end-of-file
```

†The SEQ, TS, and OPT=0 options are not allowed.
 ††TASK2 is the name of the task.

Figure D-2. FORTRAN Command List Structure

COBOL TASK COMMAND LIST STRUCTURE

A sample command list that compiles a COBOL application program and loads it onto a task library, RLTASKL, is illustrated in figure D-3.

```
COBOLS,TAF,UC1.
LDSET,LIB=TRANLIB,LIB=TRANF5.
LOAD,LGO.
NOGO,LGOB.
LIBTASK,P=RLTASKL,B=LGOB,TT,I=0.
end-of-record
      .
      . ← COBOL task appears here
      .
end-of-file
```

Figure D-3. COBOL Command List Structure

NOTE

Under COBOL, the PROGRAM-ID name is used for the task name. Only alpha-numeric names can be used as task names.

1. The purpose of this document is to provide a comprehensive overview of the current state of the project and to identify the key challenges and opportunities that we face.

2.1. Project Overview

The project is currently in the planning phase, and we have identified several key areas that require attention. These include the development of a robust infrastructure, the implementation of a secure communication system, and the establishment of a clear governance structure.

2.1.1. Infrastructure Development

One of the primary challenges we face is the need for a reliable and scalable infrastructure. This involves the selection of appropriate hardware and software, as well as the implementation of rigorous security protocols to protect our data and systems.

2.1.2. Communication System

Establishing a secure and efficient communication system is essential for the success of the project. This requires the use of encrypted channels and the implementation of strict access controls to ensure that only authorized personnel can access sensitive information.

Additionally, we must ensure that our communication system is resilient to various threats, including network outages and data breaches.

2.1.3. Governance Structure

A clear and well-defined governance structure is crucial for the effective management of the project. This involves the establishment of a steering committee and the delegation of responsibilities to various teams and individuals.

2.2. Key Challenges and Opportunities

While there are many challenges ahead, we also have several opportunities that we can leverage to our advantage. These include the growing demand for secure communication solutions, the availability of advanced technologies, and the expertise of our team.

By focusing on these key areas and opportunities, we can ensure that the project is completed on time, within budget, and to the highest standards of quality and security.

2.2.1. Infrastructure Challenges

One of the most significant challenges we face is the need for a highly secure and resilient infrastructure. This requires the use of cutting-edge technologies and the implementation of rigorous security protocols.

2.2.2. Communication Challenges

Another major challenge is the need for a secure and efficient communication system. This requires the use of encrypted channels and the implementation of strict access controls.

2.2.3. Governance Challenges

Finally, we must ensure that we have a clear and well-defined governance structure in place. This involves the establishment of a steering committee and the delegation of responsibilities to various teams and individuals.

2.2.4. Infrastructure Opportunities

One of the key opportunities we have is the growing demand for secure communication solutions. This provides us with a clear market for our services and a strong incentive to invest in research and development.

2.2.5. Communication Opportunities

Another major opportunity is the availability of advanced technologies. This allows us to leverage the latest in hardware and software to create a more secure and efficient communication system.

2.2.6. Governance Opportunities

Finally, we have the expertise of our team as a key opportunity. This allows us to draw on the skills and knowledge of our employees to overcome the challenges we face and to seize the opportunities that are available to us.

By focusing on these key areas and opportunities, we can ensure that the project is completed on time, within budget, and to the highest standards of quality and security.

Additionally, we must ensure that our communication system is resilient to various threats, including network outages and data breaches.

A clear and well-defined governance structure is crucial for the effective management of the project. This involves the establishment of a steering committee and the delegation of responsibilities to various teams and individuals.

One of the most significant challenges we face is the need for a highly secure and resilient infrastructure. This requires the use of cutting-edge technologies and the implementation of rigorous security protocols.

Another major challenge is the need for a secure and efficient communication system. This requires the use of encrypted channels and the implementation of strict access controls.

Finally, we must ensure that we have a clear and well-defined governance structure in place. This involves the establishment of a steering committee and the delegation of responsibilities to various teams and individuals.

By focusing on these key areas and opportunities, we can ensure that the project is completed on time, within budget, and to the highest standards of quality and security.

Additionally, we must ensure that our communication system is resilient to various threats, including network outages and data breaches.

A clear and well-defined governance structure is crucial for the effective management of the project. This involves the establishment of a steering committee and the delegation of responsibilities to various teams and individuals.

One of the most significant challenges we face is the need for a highly secure and resilient infrastructure. This requires the use of cutting-edge technologies and the implementation of rigorous security protocols.

Another major challenge is the need for a secure and efficient communication system. This requires the use of encrypted channels and the implementation of strict access controls.

Finally, we must ensure that we have a clear and well-defined governance structure in place. This involves the establishment of a steering committee and the delegation of responsibilities to various teams and individuals.

By focusing on these key areas and opportunities, we can ensure that the project is completed on time, within budget, and to the highest standards of quality and security.

Additionally, we must ensure that our communication system is resilient to various threats, including network outages and data breaches.

A clear and well-defined governance structure is crucial for the effective management of the project. This involves the establishment of a steering committee and the delegation of responsibilities to various teams and individuals.

One of the most significant challenges we face is the need for a highly secure and resilient infrastructure. This requires the use of cutting-edge technologies and the implementation of rigorous security protocols.

Another major challenge is the need for a secure and efficient communication system. This requires the use of encrypted channels and the implementation of strict access controls.

Finally, we must ensure that we have a clear and well-defined governance structure in place. This involves the establishment of a steering committee and the delegation of responsibilities to various teams and individuals.

By focusing on these key areas and opportunities, we can ensure that the project is completed on time, within budget, and to the highest standards of quality and security.

Additionally, we must ensure that our communication system is resilient to various threats, including network outages and data breaches.

A clear and well-defined governance structure is crucial for the effective management of the project. This involves the establishment of a steering committee and the delegation of responsibilities to various teams and individuals.

One of the most significant challenges we face is the need for a highly secure and resilient infrastructure. This requires the use of cutting-edge technologies and the implementation of rigorous security protocols.

FILES USED BY TAF

E

Table E-1 lists files used by TAF.

Table E-1. Files Used by TAF

Name	File Name	Type†	Creation	Residence	Quantity	Description
Communication recovery file	ZZCRF1	D	Automatic if K.INT declared	Same as NCTF1	1 - 8	Contains recovery information for each user. Includes status, initial input, RPUT messages, and SECURE message.
Before image recovery file	ZZxxBnn	D	User or automatic	Application	1 - 63 per data base	Contains before images of data records.
After image recovery file	ZZxxAnn	D	User or automatic	Application	2	Contains after images of data records.
Network file	NCTF1	D	User	Specified in TCF; default is TAF UI	1 - 8	Defines valid users for each network id.
Temporary recovery file	ZZCRFA1	L	Automatic	Local	1	Created during CRF reformatting. Becomes new CRF.
Backup directory file	ZZxxDIR	D	Automatic	Application	1 per data base	Contains record of data file dumps and ARF dumps.
TAF initialization procedure file	Selected by user	L	User or automatic	System	1	Contains commands for initializing and recovering TAF.
TAF initialization file	ZZTIF	L	Automatic	Local	1	Contains TAF low core pointer TST.
TAF initialization recovery report	ZZIRRF	L	Automatic	Local	1	Contains recovery report for data administrator.
TAF configuration file	TCF	I	User	TAF	1	Specifies data managers and applications in use. Contains K display commands, CRF ids, NCTF ids.
xxJ file	xxJ	I	User	TAF	1 per data base	Associates applications with data managers and describes data files and their attributes.
System task library	TASKLIB	D	User	TAF	1	Contains required and optional system tasks and transactions.
Task library	xxTASKL	D	User	Application	1 per data base	Contains all tasks and transactions for a particular application.

† Type D is a direct access permanent file; type L is a local file; type I is an indirect permanent file.

In addition to the files described in table E-1, the following reserved files are used by TAF. No task or procedure file called by TAF can attach or use any file having one of these names.

KTSROLL

OUTPUT

RECOVER

SCRA-SCRZ

SCR1-SCR5

TROB

MAP

ZZZZZ.. (file names starting with five Zs)

INSTALLATION OVERVIEW

F

Table F-1 is provided to aid the systems analyst and data administrator in performing TAF installation. It is a chronologically ordered outline of the

procedures detailed in the respective manuals. The complete name and publication number of each entry in the Manual column follows the table.

Table F-1. Installation Overview (Sheet 1 of 3)

Area	Step	Manual	Manual Keyword
1. Install NOS.		NOS IHB NOS ANHB	
2. Install communications subsystem.		NOS IHB NOS ANHB CCP Reference NAM Reference NDL Reference	
3. Install product set (COBOL, and so forth).		NOS IHB	COBOL 5 FORTRAN 4 FORTRAN 5 COMPASS
4. Modify/create TAF subsystem.	<p>1. Evaluate installation parameters. Some items of interest at initial installation are:</p> <ul style="list-style-type: none"> • TAF username. • TAF user index. • TAF password. • TAF charge, project numbers. • TAF communication interface - 2550s or multiplexers. • Number of terminals. <p>2. Modify TAF on deadstart media if required, or</p> <p>3. Modify TAF on a running NOS system.</p> <p>4. Install TAF username and charge.</p> <p>5. Install procedure file to initiate TAF.</p> <p>6. Install automatic initiation of TAF upon deadstart if desired.</p>	<p>NOS IHB</p> <p>NOS IHB NOS Reference V3</p> <p>NOS ANHB</p> <p>NOS IHB</p> <p>NOS ADHB</p> <p>NOS IHB</p> <p>NOS ANHB</p>	<p>Transaction facility</p> <p>Transaction facility, GTR, LIBEDIT, LIBGEN</p> <p>SYSEDT</p> <p>TAF installation parameters</p> <p>MODVAL, PROFILE</p> <p>TAF</p> <p>IPRDECK</p>
5. Install transaction application.	<p>1. Assign a team to do data base design. TAF interfaces to CDCS, CRM, and TOTAL. Areas in data base design that require examination include:</p> <ul style="list-style-type: none"> • Elements to be accessed. • Relationships between the elements. • Data manager to be used. • File, record types, and element types to be used to store the data. • Data formats for the tasks. • Mass storage of the data base. 	<p>TAF Reference TOTAL Reference AAM Reference TAF/CRM Reference TAF User's Guide CDCS Reference</p>	

Table F-1. Installation Overview (Sheet 2 of 3)

Area	Step	Manual	Manual Keyword
	<ul style="list-style-type: none"> • Specific processing to be assigned to tasks and the relationships between tasks. For TAF, tasks involved with I/O should be small since the tasks will occupy memory for some time. Tasks that are often used and run quickly may be made core-resident. Tasks that are often used should be reusable to avoid task loading. • Data base recovery. 		
	<p>2. Assign a team to do the communications interface. Areas to examine are:</p> <ul style="list-style-type: none"> • Network Access Methods features and workings. • Communication Control Program and 2550 hardware, communication lines, terminals, and so forth. • Network definition language (NDL). • TAF terminal I/O requests. • Message formats, terminal operator interactions. • Recovery of terminal communications. • Interface specifications with the tasks's data management activities. • NOS and network stimulators. 	<p>NAM Reference</p> <p>CCP Reference</p> <p>NDL Reference</p> <p>TAF Reference</p> <p>NOS ANHB NPS Reference</p>	
	<p>3. Install a prototype application using the following steps. Revise design as more is learned about the capability and performance trade-offs. Use stimulators to test the design under a large terminal load.</p>		
	<p>4. Describe and install the data base using the appropriate data manager.</p> <ul style="list-style-type: none"> • CDCS. • CRM. • TOTAL. 	<p>TAF Reference CDCS Reference</p> <p>TAF Reference TAF/CRM Reference</p> <p>TAF Reference TOTAL Reference</p>	<p>xxJ files</p> <p>xxJ files</p>
	<p>5. Describe and install terminal network files.</p> <ul style="list-style-type: none"> A. Install TAF's NCTF1 file. B. Install NAM's LCF and NCF network files. C. Install terminal users in NOS validation files. 	<p>TAF Reference NOS ANHB</p> <p>NDL Reference</p> <p>NOS ADHB</p>	<p>Network files Network description files</p> <p>MODVAL, PROFILE</p>
	<p>6. Modify procedure file that initializes TAF for special preprocessing procedures, such as preparing for recovery.</p>	<p>NOS IHB</p>	<p>TAF</p>

Table F-1. Installation Overview (Sheet 3 of 3)

Area	Step	Manual	Manual Keyword
	<p>7. Write application tasks.</p> <ul style="list-style-type: none"> • COBOL 5. • FORTRAN 4. • FORTRAN 5. • COMPASS. <p>8. Modify ITASK to map transaction codes to their associated transactions. Code for handling system transactions and error processing may be desired depending upon application.</p> <p>9. Modify other TAF-furnished system tasks if desired.</p> <ul style="list-style-type: none"> • LOGT - terminal logout request. • MSABT - task. • SYSMG - TAF system messages. • RTASK. • RCTASK. • CTASK. • BTASK. • CRMTASK. • Interface specifications with the tasks data management activities. <p>10. Build data base task libraries.</p> <p>11. Build TAF system library.</p> <p>12. Build data base journal files xxJ to give TAF user name for data base.</p> <p>13. Build the TAF configuration file.</p>	<p>TAF Reference TAF User's Guide</p> <p>COBOL 5 Reference COBOL 5 User's Guide</p> <p>FORTRAN 4 Reference</p> <p>FORTRAN 5 Reference</p> <p>COMPASS Reference NOS Reference V4</p> <p>TAF Reference NOS IHB</p> <p>TAF Reference</p> <p>TAF/CRM Reference</p> <p>TAF/CRM Reference CDCS Reference</p> <p>TAF Reference</p> <p>NOS IHB TAF Reference</p> <p>TAF Reference TAF/CRM Reference</p> <p>TAF Reference</p>	<p>ITASK TAF</p> <p>System tasks</p> <p>LIBTASK</p> <p>TAF LIBTASK</p> <p>xxJ files xxJ files</p>
6. Test the transaction application.			
7. TAF operational commands.	<p>1. Initialize TAF from computer console.</p> <p>2. Status TAF from computer console.</p> <p>3. Terminate TAF from computer console.</p>	NOS OHB	Transaction subsystem
8. Modify performance of TAF and application.	<p>1. TAF installation parameters.</p> <p>2. Terminal stimulations.</p>	NOS IHB	TAF

<u>Control Data Manual</u>	<u>Manual Title</u>	<u>Publication Number</u>
AAM Reference	CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300
CCP Reference	Communications Control Program Version 3 Reference Manual	60471400
CDCS Reference	CYBER Database Control System Version 2 Data Administration Reference Manual	60485200
	CYBER Database Control System Version 2 Application Programming Reference Manual	60485300
COBOL 5 Reference	COBOL Version 5 Reference Manual	60497100
COBOL 5 User's Guide	COBOL Version 5 User's Guide	60497200
COMPASS Reference	COMPASS Version 3 Reference Manual	60492600
FORTRAN 4 Reference	FORTRAN Extended Version 4 Reference Manual	60497800
FORTRAN 5 Reference	FORTRAN Version 5 Reference Manual	60481300
NAM Reference	Network Products Network Access Method Version 1 Reference Manual	60499500
NDL Reference	Network Products Network Access Method Version 1 Network Definition Language Reference Manual	60480000
NOS ADHB	NOS Version 2 Administration Handbook	60459840
NOS ANHB	NOS Version 2 Analysis Handbook	60459300
NOS IHB	NOS Version 2 Installation Handbook	60459320
NOS OHB	NOS Version 2 Operations Handbook	60459310
NOS Reference V3	NOS Version 2 Reference Set, Volume 3, System Commands	60459680
NOS Reference V4	NOS Version 2 Reference Set, Volume 4, Program Interface	60459690
NPS Reference	Network Products Stimulator Version 1 Reference Manual	60480500
TAF/CRM Reference	TAF/CRM Data Manager Version 1 Reference Manual	60459510
TAF Reference	TAF Version 1 Reference Manual	60459500
TAF User's Guide	TAF Version 1 User's Guide	60459520
TOTAL Reference	TOTAL - CDC Version 2 Reference Manual	76070300

Section 1

Section 2

Section 3

Section 4

Section 5

Section 6

Section 7

Section 8

Section 9

Section 10

Section 11

Section 12

Section 13

Section 14

Section 15

Section 16

Section 17

Section 18

Section 19

Section 20

Section 1

Section 2

Section 3

Section 4

Section 5

Section 6

Section 7

Section 8

Section 9

Section 10

Section 11

Section 12

Section 13

Section 14

Section 15

Section 16

Section 17

Section 18

Section 19

Section 20

Section 21

Section 22

Section 23

Section 24



INTRODUCTION

The network, through which terminals access a host and its applications (like TAF), can support many types of terminals. All supported terminals are grouped by the network into 18 standard terminal classes. Each terminal class has specific operating characteristics, also referred to as terminal characteristics. These terminal characteristics, taken together, make up a terminal definition and are predefined by the network to closely match the operating characteristics of actual terminals. The network accepts changes to any of these terminal definitions through terminal definition commands. These commands specify one or more parameters, each of which may give a new value to a terminal characteristic. Tables G-1 and G-2 list these terminal definitions, the default values, and possible values for each of the 18 classes.

The tables use the following abbreviations when identifying the terminal types for each terminal class:

<u>Abbreviation</u>	<u>Terminal</u>
HASP	Terminals using the Houston Automatic Spooling Program communication protocol.
HAZ	Hazeltine.
POST	HASP terminals that only support postprint format control.
PRE	HASP terminals that only support preprint format control.
Tek	Tektronix (teletypewriter emulation).
TTY	Teletype.
UT	User Terminal.
X.25	Terminals connected to a packet switching network (PSN) using the X.25 communication protocol.
X3.64	Terminals conforming to the ANSI X3.64 standard.

When a terminal logs in to the network, either the network assumes it to be of a certain terminal class or assigns it a terminal class that resembles the terminal's actual characteristics. If you do not know the terminal class in use, you can enter the CH command (described in this appendix) to ascertain the terminal class and some of the characteristics of your terminal. In either case, if the character-

istics of your terminal do not match those of its assigned terminal class, you can change the values of the terminal characteristics or even change the terminal class by using the terminal definition commands described in this appendix. You can enter terminal definition commands any time the terminal is connected to the network (refer to appendix J for information on how to access a host).

When you use a terminal definition command to change a value, that change remains in effect until the terminal is disconnected from the network or another terminal definition command is used to change the value. Even application switching and logout do not change the values of the terminal characteristics if the terminal has not been disconnected from the network. Disconnecting your terminal from the network (that is, disconnecting the phone and redialing on dial-up terminals or turning a dedicated terminal off and then on again) causes all terminal characteristics to be reset to site-defined default values. The site defines these values using the Network Definition Language (NDL).

You can also reset your terminal's characteristics by entering the TC command. The TC command resets your terminal characteristics to the default values specified by the Communications Control Program (CCP). Note that CCP-defined values may differ from the site-defined NDL values.

It is also possible to change terminal definitions using the TRMDEF command or by using control byte 16 from an interactive job. The TRMDEF command is described under Terminal Control Commands in the NOS Reference Set, Volume 3, System Commands. Refer to NOS Reference Set, Volume 4, Program Interface, for information on control bytes.

TERMINAL DEFINITION COMMAND FORMAT

This section describes the general format of a terminal definition command. The specific format of each command appears later in this appendix.

General formats:

```
ct keyword=value†
ct keyword1=value1 ct keyword2=value2
...ct keywordn=valuen†
```

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command. Table G-1 gives

†The spaces shown are for clarity only. Do not include spaces when you enter a terminal definition command.

<u>Parameter</u>	<u>Description</u>
	the default network control character for your terminal.†
keyword	Represents a mnemonic associated with a terminal characteristic.
value	Specifies a particular value associated with the terminal definition mnemonic. Table G-2 gives the range of values for each terminal definition keyword. On some commands the =value portion of the command is optional and defaults to =Y.
keyword ₁	Same as keyword.
value ₁	Same as value.

The second format allows you to make multiple terminal definitions in one line. If the line extends beyond character position 54, the results are unpredictable. Also, if you change the network control character *ct* in a concatenated group of commands, the CT command does not take effect until the remaining commands in the group have been processed.

As with all commands described in this manual, you must press the message transmission key in order for the system to process your command. Appendix K of the NOS Reference Set, Volume 3, System Commands, shows the default message transmission keys for the various terminal classes. This appendix describes how you can change the message transmission key (refer to the EB and EL commands).

Example:

To change the page width of a 200 User Terminal, enter these characters

```
%PW=50
```

and press the SEND key. The % character is the network control character, PW is the keyword associated with page width, 50 is the new page width for the terminal, and the SEND key is the message transmission key for a 200 User Terminal.

RESTRICTIONS

You should not enter terminal definition commands while output is in progress. The network may not be able to perform such commands immediately.

The network presets all user-adjustable terminal characteristics any time you establish a connection to the network. The site administrators choose these preset values. Additionally, when a terminal is disconnected from the network and then reconnected, the network resets any terminal characteristics you have changed to their preset values.

To enter a terminal definition command at asynchronous terminals from which you can enter more than one line before transmitting (IN=BK is specified),

the terminal definition command must be the last input line in the transmission. If the input is from paper tape (IN=PT is specified), the tape reader must stop. Unpredictable results can occur if more input follows the terminal definition command in the transmission block.

Characters or character codes you specify for the value portion of terminal definition commands are subject to the following restrictions:

- You cannot use the following ASCII characters (or their character codes) in the value portion of the AB, CN, BS, B1, B2, CT, EB, or EL commands:
 - any lowercase or uppercase alphabetic character
 - any digit 0 through 9
 - NUL
 - SOH
 - STX
 - =
 - DEL
 - space
- The ASCII characters or character codes you specify for the AB, BS, B1, B2, CN, CT, EB, and EL terminal definitions must all be different, except that AB and CN can have the same value and EB and EL may have the same value.
- If output flow control is used, do not use DC1s and DC3s in the AB, CN, BS, B1, B2, CT, EB, or EL commands.

TERMINAL DEFINITION COMMANDS

This subsection describes the function and format of each terminal definition command. The headings consist of the command name (a two-character mnemonic) followed by the terminal attribute it defines. Generally, the format descriptions do not show the default values or ranges of values since they vary with terminal class. Tables G-1 and G-2 show this information. If you are unable to find the key on your keyboard that corresponds to a particular character given in table G-1, consult the documentation for your terminal.

This section refers to categories of terminal classes to which commands apply. The categories are as follows:

<u>Category</u>	<u>Terminal Classes</u>
Asynchronous	1 through 8
Mode 4	10, 11, 12, 13, 15
HASP	9, 14
Bisynchronous	16, 17, 18
X.25 PSN	1, 2, 3, 5, 6, 7, 8
3270	18

† You must press the ATTN key on a 2741-compatible terminal before entering this character.

(AB) - ABORT OUTPUT BLOCK CHARACTER

The AB command specifies the character used to abort an output block. When you enter this character from the terminal as the only character in a line, the current transmission block of output to the terminal is discarded.

Format:

ct AB=ab

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
ab	Specifies the new abort output block character. This character must differ from the characters defined for BS, B1, B2, CT, EB, and EL.

This command only applies to asynchronous terminals and is not valid for X.25 PSN terminals.

(AR) - AUTOMATIC CHARACTER RECOGNITION

The AR command allows you to change the terminal's character set or its line speed, in many cases, by means of a switch on the terminal or a removable type ball. You can enter this command any time after the system prompts you for input, even before you enter your user name during login.

Format:

ct AR

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.

After entering the AR command, you must select the terminal speed and enter a carriage return. The system acknowledges the carriage return operation and indicates that the line speed recognition is complete by returning two line feeds. If your terminal cannot change its line speed, no line feed responses occur. At this point, enter a closing parenthesis followed by a carriage return so that the system can recognize the terminal's new character set. When the system recognizes the new character set, it issues a line feed and the message AR ACCEPTED.. to indicate you can continue. This is also the auto-recognition procedure used to connect the terminal to the system (refer to the login/logout procedures in Volume 1, Introduction to Interactive Usage).

This command only applies to asynchronous terminals and is not valid for X.25 PSN terminals. X.25 PSN terminals use the ASCII code and character set.

(BF) - BLOCKING FACTOR

The BF command sets the size of intermediate data blocks that will be delivered to the host.

Format:

ct BF=bf

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
bf	Specifies the blocking factor. You can specify the following values for bf:

<u>bf</u>	<u>Description</u>
0	Selects a line feed character and a character count of 100 as data-forwarding signals.
1	Selects a character count of 100 as a data-forwarding signal.
2	Selects a character count of 200 as a data-forwarding signal.

Regardless of the blocking factor, an end-of-line character (carriage return) forwards data to the host.

(BR) - BREAK KEY AS USER BREAK 1

The BR command determines if the BREAK key (interruption key for interactive output) also functions as the interruption sequence (user break 1 sequence) when entered during output and as the cancel character when entered during input.

Format:

ct BR=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Determines if this feature is enabled or disabled. A value of Y enables and a value of N disables this feature.

This command only applies to asynchronous terminals. For X.25 PSN terminals, the BREAK key can only

function as the interruption sequence whether entered during input or output.

(BS) - BACKSPACE CHARACTER

The BS command specifies the character used to delete the previous input character.

Format:

ct BS=bs

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
bs	Specifies the new backspace character. This character must differ from the characters currently defined for AB, B1, B2, CN, CT, EB, and EL.

It is possible to backspace only to the beginning of the current physical line; additional backspaces are disregarded. When a page width of 0 is selected, the network assumes a page width of 100 characters for backspacing. If the page width is in the range 20 through 56, backspacing into previous physical lines is sometimes permitted.

This command only applies to asynchronous terminals only.

(B1) - INTERRUPTION OR USER BREAK 1 CHARACTER

The B1 command specifies the character that, when entered as the only character in a logical line (interruption character followed by the message transmission key), causes program interruption.

Format:

ct B1=ubl

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
ubl	Specifies the new user break 1 character. This character must differ from the characters currently defined for AB, BS, B2, CN, CT, EB, and EL.

The interruption sequence is discussed in section 17. The BR command determines whether the BREAK key can also function as the B1 character.

This command does not apply to bisynchronous 2780 and 3780 terminals.

(B2) - TERMINATION OR USER BREAK 2 CHARACTER

The B2 command specifies the character that, when entered as the only character in a logical line (termination character followed by the message transmission key), causes program termination.

Format:

ct B2=ub2

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
ub2	Specifies the new user break 2 character. This character must differ from the characters currently defined for AB, BS, B1, CN, CT, EB, and EL.

The termination sequence is discussed in section 17.

This command does not apply to bisynchronous 2780 and 3780 terminals.

(CH) - DISPLAY TERMINAL CHARACTERISTICS

The CH command displays some of the terminal's current characteristics.

Format:

ct CH

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.

The CH command produces a display in the following format:

TC=tc,BS=bs,CN=cn,AB=ab,B1=ubl,B2=ub2,EL=el,EB=eb

where the variable portions appear as ASCII characters if printable, mnemonics (for example, STX) if nonprintable, or N/A if the attribute is not applicable.

(CI) - CARRIAGE RETURN IDLE COUNT

The CI command specifies the number of idle characters to be inserted into the output stream after a carriage return.

Format:

ct CI=ci

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
ci	Specifies the new carriage return idle count. This value can be from 0 to 99 or the string CA. A value of CA restores the carriage return idle count to the default value.

When the network produces a carriage return, the network outputs the specified number of idle characters before outputting the next line. This allows time for the carriage return function and ensures that characters are not lost because of printing attempts during the carriage return operation.

This command only applies to asynchronous terminals.

(CN) - CANCEL CHARACTER

The CN command specifies the character used to cancel the logical line currently being input.

Format:

ct CN=cn

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
cn	Specifies the new cancel character. This character must differ from the characters currently defined for BS, Bl, B2, CT, EB, and EL.

When you enter the cancel character as the last character in a line, the entire logical line in progress is cancelled. The system responds to a cancel line character by printing *DEL* on the next line and positioning the carriage to the beginning of a new line. The BR command determines whether the BREAK key can also function as the CN character.

This command does not apply to bisynchronous 2780, 3270, and 3780 terminals.

(CP) - CURSOR POSITIONING AFTER INPUT

The CP command determines whether or not the system sends the terminal a response to the line feed key, the EL character, and the EB character.

Format:

ct CP=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Indicates whether to send a response to the terminal or not. A value of Y indicates yes and N indicates no.

This command applies only to asynchronous terminals (except 2741s). If CP=Y and IN=KB are selected, the system responds to the line feed key by sending a carriage return and responds to an EL or EB character by sending the cursor positioning response specified by the EL or EB command. If IN=BK is selected or you are at an X.25 PAD terminal, the system responds only to the EB character by sending the cursor positioning response selected by the EB command.

(CT) - NETWORK CONTROL CHARACTER

The CT command specifies a network control character for the terminal. When you enter this character as the first character of a logical line, it signals that what follows is a terminal definition command or a special command that the network forwards to your application with a preemptive status.

Format:

ct CT=char

<u>Parameter</u>	<u>Description</u>
ct	Represents the current network control character for your terminal.
char	Specifies the new network control character. This character must differ from the characters currently defined for AB, BS, Bl, B2, CN, EB, and EL.

If you enter the CT command in a line with multiple terminal definitions, the new value char does not take effect until the complete line is processed.

(DL) - DELIMITERS FOR SINGLE-MESSAGE TRANSPARENT INPUT MODE

The DL command specifies the delimiters that terminate single-message transparent input mode. When you initiate transparent input mode (IN=X, IN=XK, or IN=XP), the network reads your input and sends it to the system without translation until it encounters a delimiter. If your terminal's parity is set to even, odd, or zero (PA=E, O, or Z), the parity bit is stripped (set to zero) before the characters are sent to the host, although no

translation occurs. If your terminal's parity is set to none or ignore (PA=N or I), the parity bit is not stripped. See the NAM/CCP Terminal Interfaces Reference Manual for more information on transparent input and parity. The first delimiter encountered terminates transparent input mode.

Format:

ct DL=Xxx,Ccount,TO

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
Xxx	Specifies the 2-digit hexadecimal code (xx) of the character you want as a delimiter. The network does not send the character as part of the input data. When you specify a hexadecimal code above 7F, you must have previously set your terminal's parity to none (PA=N) or ignore (PA=I); otherwise the network cannot detect the code when it occurs in a transparent mode message. If the terminal's parity is set to none, the 8-bit hexadecimal code input from the terminal is matched against the 8-bit hexadecimal code specified as the delimiter (xx). If the terminal's parity is set to ignore, the bottom 7 bits of the terminal input are matched against the bottom 7 bits of the specified delimiter (xx).
Ccount	Specifies a decimal value (count) from 1 to 4095† that functions as a character count delimiter.
TO	Specifies that a timeout of between 200 and 400 milliseconds will be a transparent mode delimiter.

You can select any of the three types of delimiters: character delimiter (specify Xxx), character count delimiter (specify Ccount), and timeout delimiter (specify TO).

The parameter values for DL are order-independent and optional. However, you must specify at least one of the parameter values. You can omit trailing commas if you do not specify all three types of delimiters.

Entering this command with any number of new delimiters cancels all transparent mode input delimiters already in effect, including those specified with the XL command.

The DL command applies only to asynchronous and mode 4 terminals. Terminal class 4 is configured with the RETURN key as the only transparent mode terminator and mode 4 terminals are configured with

†These values are valid within the network. However, unless changed during installation, the maximum logical input line length in IAF (160 characters) limits you.

the SEND or ETX key as their only transparent mode terminator. For X.25 devices, the PAD forwarding signal always acts as a transparent mode terminator in addition to any other options selected.

(EB) - END-OF-BLOCK CHARACTER

The EB command defines the end-of-block character and defines the cursor-positioning response to that character. The EB character acts as the message transmission key for terminals operating in block mode (refer to the IN command).

Format:

ct EB=eb,cpr

<u>Parameter</u>	<u>Description</u>								
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.								
eb	Specifies the new end-of-block character. You can specify the following values for eb:								
	<table border="1"> <thead> <tr> <th><u>eb</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>char</td> <td>Selects the new EB character. This character must differ from the characters currently defined for AB, BS, Bl, B2, CN, and CT.</td> </tr> <tr> <td>EB</td> <td>Selects the default end-of-block character or character sequence.</td> </tr> <tr> <td>EL</td> <td>Selects the current logical end-of-line character.</td> </tr> </tbody> </table>	<u>eb</u>	<u>Description</u>	char	Selects the new EB character. This character must differ from the characters currently defined for AB, BS, Bl, B2, CN, and CT.	EB	Selects the default end-of-block character or character sequence.	EL	Selects the current logical end-of-line character.
<u>eb</u>	<u>Description</u>								
char	Selects the new EB character. This character must differ from the characters currently defined for AB, BS, Bl, B2, CN, and CT.								
EB	Selects the default end-of-block character or character sequence.								
EL	Selects the current logical end-of-line character.								
cpr	Specifies the cursor-positioning response to the EB character. You can specify the following values for cpr:								

<u>cpr</u>	<u>Description</u>
CR	Send a carriage return to the terminal.
LF	Send a line feed to the terminal.
CL	Send a carriage return and a line feed to the terminal.
NO	Send no response.

The two EB parameters (EB=eb and cpr) are optional and order-independent. If you omit one, the system retains its previous value.

This command is valid only for asynchronous terminals, except for paper tape devices and 2741 terminals. The end-of-block character for mode 4 terminals is ETX and is not changeable. The X.25 PAD forwarding signal is the default for X.25 PSN terminals and always acts as an end-of-block. If you specify the char form of the eb parameter, it is only effective if it occurs as the last character in a packet sequence.

The CP command enables or disables the cursor-positioning response specified by the cpr parameter.

(EL) - END-OF-LINE CHARACTER

The EL command defines the logical end-of-line character (carriage return) and defines the cursor-positioning response to that character. It also defines the message transmission key for asynchronous terminals not operating in block mode (refer to the IN command).

Format:

ct EL=e1, cpr

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
e1	Specifies the new logical end-of-line character. You can specify the following values for e1:
<u>e1</u>	<u>Description</u>
char	Selects the new EL character. This character must differ from the characters currently defined for AB, BS, B1, B2, CN, and CT.
EB	Selects the current end-of-block character or character sequence.
EL	Selects the default logical end-of-line character.
cpr	Specifies the cursor-positioning response to the EL character. You can specify the following values for cpr:

<u>cpr</u>	<u>Description</u>
CR	Send a carriage return to the terminal.
LF	Send a line feed to the terminal.

<u>cpr</u>	<u>Description</u>
CL	Send a carriage return and a line feed to the terminal.
NO	Send no response.

The two EL parameters (e1 and cpr) are optional and order-independent. If you omit one, the system retains its previous value.

This command does not apply to 2741, HASP, 3270, and bisynchronous 2780 and 3780 terminals; the cpr parameter does not apply to mode 4 or X.25 PSN terminals; and the char specification does not apply to mode 4 terminals.

The CP command enables or disables the cursor-positioning response specified by the cpr parameter.

(EP) - ECHOPLEX MODE

The EP command enables or disables the echoing of input characters back to the terminal.

Format:

ct EP=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) input echoing.

Some terminals can perform their own character echoing, such as terminals with a HALF/FULL duplex switch. If the switch is in the HALF position, set the EP value to N, and the terminal echoes the input. If the switch is in the FULL position, set the EP value to Y and the network echoes the input.

You can use the EP command and full-duplex mode as a security measure. If you are in full-duplex mode and EP is Y during login, the system echoes all of your input except your password. This is not true, however, if you enter your user name and password on the same line.

This command only applies to asynchronous terminals (except 2741s) but is not valid for X.25 PSN terminals.

(FA) - FULL ASCII INPUT MODE

The FA command enables or disables the full-ASCII input mode.

Format:

ct FA=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) the full-ASCII input mode.

In full-ASCII input, all 128 ASCII characters, including NUL, DEL, STX, LF, and all BLK and message-forwarding signals (which are normally discarded), are sent to the host. Exceptions to this are the following characters:

- NULs, DELs and LFs when encountered as the first character of an input line or as the character following a message-forwarding signal
- CRs that follow LFs when inputting from paper tape
- DC3s that follow the EL character when inputting from paper tape
- DC1s and DC3s when output control (OC=Y) is selected

In full-ASCII mode, the network does not recognize the terminal functions associated with the mnemonics B1, B2, CN, AB, and BS. The network does recognize terminal definition commands and other commands beginning with the CT character.

If SE is in effect when you select FA or if FA is in effect when you select SE, FA takes precedence and SE will not be effective until FA is turned off. If full-ASCII mode is enabled when you select transparent input mode, the network suspends the full-ASCII mode until the transparent input mode terminates.

This command only applies to asynchronous and mode 4 terminals.

(HC) - HOST CONNECTION

The HC command sets or changes the host connected to your terminal. The host availability display (HAD) shows the available hosts (refer to the HD command for more information on the HAD). This command disconnects you from your current host and establishes a new host connection. This command is equivalent to HS or HN followed by TM.

Format:

ct HC=hostname

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
hostname	specifies the name of the host to which you want your terminal connected. The valid entries appear in the HAD.

The host selected is used until either you change it or until your terminal is disconnected from the host.

(HD) - DISPLAY OF HOST NODES

The HD command controls when the system displays the host availability display (HAD), which shows all paths you can select to connect your terminal to a host. If your computer system is a multihost system, your site may choose to show the HAD whenever you disconnect from a host unless you inhibit the display.

Format:

ct HD=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Represents Y or N. Y enables and N disables the automatic displaying of the HAD.

If you specify HD=Y, you get a display of the HAD immediately and at any time you disconnect from the host (for example, after you enter the TM command or after you log out of an application without switching to another application). You also get a header and trailer line along with the HAD (refer to the TM command).

The HAD has the following format:

```

message          CONTROL CHARACTER=ct
NPU NODE=npu    TERMINAL NAME=terminal

HOST            NODE            SELECTED/
                NODE            CONNECTED
                NODE            STATUS
host1          node1          condition1   status1
host2          node2          condition2   status2
.               .               .               .
.               .               .               .
.               .               .               .
hostn          noden          conditionn   statusn

```

The variable items in the display have the following descriptions.

<u>Item</u>	<u>Description</u>
message	A message describing the status of your current host connection.
ct	Network control character currently defined for your terminal.
npu	The node number of the Network Processing Unit (NPU) your terminal is connected to.
terminal	The name of your terminal.
host _i	Represents the 1- to 7-character name of a host. The field may be blank.

<u>Item</u>	<u>Description</u>
node ₁	Represents a node number (1-31).
condition ₁	Indicates that you have selected node ₁ (S), that you are connected to the host using node ₁ (C), or that node ₁ has been automatically selected and the connection to the host is in progress (SA).
status ₁	Represents AVAILABLE or NOT AVAILABLE.

(HN) - HOST NODE SELECTION

The HN command sets or changes the host or the path (node) to a host through which your terminal is connected. This command does not disconnect you from your current host connection. You must log out or disconnect with the Secure Login command or TM terminal definition command before you can connect to the new selection.

Format:

ct HN=node

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
node	Specifies a node number. The default is the node the network processing unit is presently using to access its supervisor (CS).

The host availability display (HAD) lists all nodes your terminal can use. Refer to the HD command for a description of the HAD.

(HS) - HOST SELECTION

The HS command selects a host which is connected to your terminal. The host availability display (HAD) shows the available hosts (refer to the HD command).

Format:

ct HS=hostname

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
hostname	Specifies the name of the host to which you want your terminal connected. The valid entries appear in the HAD.

Your terminal uses the path to host with the least traffic. The path selected is used until either you change it or until your terminal is disconnected from the host.

(IC) - FLOW CONTROL FOR INPUT DEVICES

The IC command specifies whether the terminal recognizes DC3s as a stop signal and DC1s as a resume signal for terminal input.

Format:

ct IC=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) input control.

Input control applies to asynchronous terminals, including paper tape devices (except 2741 and X.25 PSN terminals).

(IN) - INPUT DEVICE AND TRANSMISSION MODE

The IN command identifies the input device and the transmission mode of the device. As the input device, you can specify one of the following:

- A keyboard.
- A paper tape reader.

As the input transmission mode, you can specify either:

- Normalized line mode (one message or logical line per terminal transmission block).
- Normalized block mode (one or more logical lines or messages collected into a block before it is transmitted).
- Transparent mode (input is not edited or translated by the network).

Format:

ct IN=in

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
in	Specifies the input device and transmission mode. You can specify the following values for in:

<u>in</u>	<u>Description</u>
BK	Selects block mode as from a form terminal and normalized input mode.
KB	Selects a keyboard in character or line mode and normalized input mode.
PT	Selects paper tape and normalized input mode.
X	Selects the current input device and transparent input mode.
XK	Selects the keyboard and transparent input mode.
XP	Selects paper tape and transparent input mode.

When the IN value is XK, XP, or X, terminal input is sent to the host in transparent mode blocks. Transparent mode continues until the system encounters a previously defined transparent mode terminator.

When the IN value is XK and transparent mode terminates, the IN value automatically reverts to KB. When the IN value is XP and transparent mode terminates, the IN value automatically reverts to PT. When the IN value is X and transparent mode terminates, the IN value automatically reverts to the value that was previously in effect.

When the IN value is PT or XP, the network sends the X-ON code to start the paper tape reader automatically after the end of a message is output to the terminal and the message empties the terminal's output queue.

The IN command is not applicable to HASP, 3270 or bisynchronous 2780 and 3780 terminals. 2741 terminals can only select IN=KB, IN=X, and IN=XK. Mode 4 terminals, which are always in block mode, and X.25 PSN terminals cannot select PT or XP mode.

(L) - LINE FEED IDLE COUNT

The LI command specifies the number of idle characters to be inserted into the output stream following a line feed.

Format:

ct LI=value

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.

<u>Parameter</u>	<u>Description</u>
value	Specifies the new line feed idle count. This value can be from 0 to 99 or the string CA. A value of CA restores the line feed idle count to the default value.

The line feed idle count is similar to the carriage return idle count (CI), except that the idle characters are output after a line feed instead of after the carriage return.

This command only applies to asynchronous terminals.

(LK) - LOCKOUT OF UNSOLICITED MESSAGES

The LK command determines whether unsolicited messages from the network operator appear at your terminal.

Format:

ct LK=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to lock out unsolicited messages. A value of Y locks out the messages and a value of N allows the messages to appear.

This command is not valid for bisynchronous 2780 and 3780 terminals.

(MS) - MESSAGE TO NETWORK OPERATOR

The MS command sends a message to the network operator.

Format:

ct MS=message

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
message	Specifies your message to the operator. You can use any string of 50 characters or less.

If you are having problems with the network (such as difficulty connecting to IAF), you can communicate these problems to the network operator using the MS command.

(OC) - FLOW CONTROL FOR OUTPUT DEVICES

The OC command specifies whether the terminal generates DC3s as a stop signal to the network and DC1s as a resume signal for terminal output.

Format:

ct OC=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) output control.

If the network receives a DC3 while output control is enabled, it suspends output until it receives a DC1 or until you send another block of input.

Output control applies to asynchronous terminals, including paper tape devices (except 2741s).

(OP) - OUTPUT DEVICE SELECTION

The OP command specifies the type of output device.

Format:

ct OP=op

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
op	Specifies the output device. You can specify the following values for op:
<u>op</u>	<u>Description</u>
DI	Specifies a display device.
PR	Specifies a printer.
PT	Specifies a paper tape punch.

You can punch a tape in any mode, but the network does not provide the proper X-OFF characters unless OP has a value of PT.

If you select OP=PR, the network does line folding and page bounding for asynchronous and X.25 PSN terminals. If OP=DI, the network assumes the terminal performs these functions itself.

This command only applies to asynchronous terminals (except 2741s). OP=PT does not apply to X.25 PSN terminals.

(PA) - PARITY PROCESSING

The PA command specifies the parity processing to be performed by the network.

Format:

ct PA=pa

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
pa	Specifies the type of parity processing. You can specify the following values for pa:

<u>pa</u>	<u>Description</u>
E	Set parity bit for even parity during output, and check for even parity and set parity bit to zero on input.
N	In transparent mode, do not set or check the parity bit because it is part of the input or output data. In normalized mode, set the parity bit to zero on input and output.
O	Set parity bit for odd parity during output, and check for odd parity and set parity bit to zero on input.
Z	Set parity bit to zero on output and check parity bit against zero on input.
I	In transparent mode, do not set or check the parity bit because it is part of the input or output data. Ignore the parity bit when checking for transparent input mode delimiters. In normalized mode, set the parity bit to zero on input and output.

This command only applies to asynchronous terminals (except 2741s). For X.25 PSN terminals, the parity bit is set to 0 and not checked for input.

(PG) - PAGE WAITING

The PG command enables or disables page waiting at the terminal during output. If you enable page waiting, the terminal stops at the end of each output page for your acknowledgement (a null input line) before output continues.

Format:

ct PG=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Specifies whether to enable (Y) or disable (N) page waiting.

When you enter a null input line to get the next page of output, the null input line has no other meaning.

Page waiting does not apply to bisynchronous 2780 or 3780 terminals or to HASP terminals.

(PL) - PAGE LENGTH

The PL command establishes the maximum number of physical lines that can be printed as one page. If OP has a value of PR, the network automatically positions the carriage to the top of the form after PL lines have been output.

Format:

ct PL=pl

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
pl	Specifies the page length. You can specify 0 or any value from 8 to 255. PL=0 selects an infinitely long page.

The PL value must be nonzero if you want to enable page waiting, since page waiting only occurs after the number of lines defined by pl-1 are output.

(PW) - PAGE WIDTH

The PW command establishes the maximum number of characters that the terminal prints on one output line. If OP has the value PR, the system automatically starts a new line after pw characters have been output.

Format:

ct PW=pw

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
pw	Specifies the page width. You can specify a value of 0 or any value from 20 to 255. PW=0 selects an infinitely long line.

(SE) - SPECIAL EDITING MODE

The SE command enables or disables the special editing variant of normalized input mode.

Format:

ct SE=option

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
option	Enables (Y) or disables (N) special editing operations.

When special editing is in effect, any backspace character entered is stored for transmission to the host. Similarly, a line feed entry produces a normal line feed operation at the terminal and the network stores the line feed character code for transmission to the host. A backspace followed by a line feed causes the system to output a bell and an exclamation point on an ASCII terminal or an inverted caret (v) on an APL terminal below the last character entered.

This command is only valid for asynchronous terminals but not for X.25 PSN terminals.

(TC) - TERMINAL CLASS COMMAND

The TC command specifies a terminal class for your terminal. The terminal class associates your terminal with a predefined set of terminal characteristics. To effectively use a terminal class, choose a terminal class whose default terminal definition corresponds to the characteristics of your terminal.

Format:

ct TC=tc

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
tc	Specifies the terminal class. The value tc can be from 1 to 17.

The following list associates terminal class values with terminal types:

<u>tc</u>	<u>Terminal Type</u>
1	M33, M35, M37, M38 teletypes
2	CDC 713-10, 722, 751-1, 752, 756
3	CDC 721
4	IBM 2741
5	M40 teletypes
6	Hazeltine 2000
7	Terminals conforming to the ANSI X3.64 standard (DEC VT100)
8	Tektronix 4010, 4014
9	HASP protocol, postprint format control
10	200 User Terminal
11	CDC 714-30
12	CDC 711-10
13	CDC 714-10/20
14	HASP protocol, preprint format control
15	CDC 734, 731-12, 732-12
16	IBM 2780
17	IBM 3780
18	IBM 3270

A terminal that is not shown as belonging to a terminal class may still be operational. The terminal can be assigned to a terminal class having similar characteristics and the terminal's definition can be modified as necessary to correctly define the operational characteristics of the terminal.

If your terminal is on an auto-recognition line and its terminal class is other than 1, 4, 10, or 13, you can use the TC command to change its terminal class to ensure proper operation.

If you include a TC command in a sequence of terminal definition commands, enter the TC command first as it resets all other terminal characteristics.

(TM) - TERMINATING A TERMINAL-HOST CONNECTION

The TM command severs the connection between your terminal and a host.

Format:

ct TM

Parameter Description

ct Represents the network control character for your terminal as defined by the CT command.

You can use the TM command if your application does not respond to your input. The system responds to the TM command in the following manner:

host status CONTROL CHARACTER=ct
had
prompt

The variable items have the following descriptions:

<u>Item</u>	<u>Description</u>
host status	Indicates the status of the host you have just specified or to which you are/were connected. You can get any one of the following messages:

- HOST AVAILABLE
- HOST BUSY
- HOST CONNECTED
- HOST DISCONNECTED
- HOST UNAVAILABLE
- NO HOST AVAILABLE
- NO HOST CONNECTED
- NO HOST SELECTED

ct Represents the network control character for your terminal as defined by the CT command.

had Represents the host availability display (HAD). The HAD appears only if you have enabled its automatic display (refer to the HD command).

prompt Represents an action the system wants you to take or a status message. You can get any one of the following prompts:

- ENTER ct HD TO SEE HOST STATUS
- ENTER ct HN=NN TO SELECT HOST
- ENTER INPUT TO CONNECT TO HOST
- INPUT DISCARDED
- READY FOR INPUT
- TERMINAL DISABLED BY NOP

(XL) - MULTIMESSAGE TRANSPARENT MODE

The network provides a transparent mode that allows you to remain in transparent input mode even after message blocks have been forwarded to the application. The XL command specifies the multimessage-forwarding signal and the terminator for transparent input mode. If your terminal's parity is set to even, odd, or zero (PA=E, O or Z), the parity bit is stripped (set to zero) before the characters are sent to the host, although no translation occurs. If your terminal's parity is set to none or ignore (PA=N or I), the parity bit is not stripped. See the NAM/CCP Terminal Interfaces Reference manual for more information on transparent input and parity.

Format:

ct XL=Xxx,Ccount,TO,Yyy

<u>Parameter</u>	<u>Description</u>
ct	Represents the network control character for your terminal as defined by the CT terminal definition command.
Xxx	Selects the character with the two-digit hexadecimal representation xx as the messageforwarding signal. If you do not specify a value yy, the occurrence of two successive characters defined by xx acts as a terminator for transparent input mode. When you specify a hexadecimal code above 7F, you must have previously set your terminal's parity to none (PA=N) or ignore (PA=I); otherwise the network cannot detect the code when it occurs in a transparent mode message.
Ccount	Selects a decimal value from 1 to 4095† as a character count that functions as a messageforwarding signal.
TO	Selects timeout as a transparent input mode terminator.
Yyy	Selects the character with the two-digit hexadecimal represen-

Parameter

Description

tation yy as a terminator for transparent input mode when it follows xx. When you specify a hexadecimal code above 7F, you must have previously set your terminal's parity to none (PA=N) or ignore (PA=I); otherwise the network cannot detect the code when it occurs in a transparent mode message.

Yyy, if specified, must follow the Xxx parameter; otherwise, it is ignored. The parameters are otherwise order-independent. You must specify at least one of Xxx or TO to select a transparent mode terminator. When it immediately follows the character defined by Xxx on input, the character defined by yy terminates transparent input mode. The characters defined by Xxx and Yyy can be identical or you can omit Yyy. In either case, two successive input characters defined by Xxx are required to terminate transparent mode.

The message-forwarding signals and mode terminators are not sent to the host.

This command cancels all transparent mode terminators and delimiters defined by the DL command or a previous XL command. You can select multmessage transparent input mode by using the XL command in conjunction with the IN command and selecting IN=X, IN=XK, or IN=XP.

This command does not apply to HASP and bisynchronous terminals.

†These values are valid with the network. However, unless changed during installation, the maximum logical input line length in IAF (160 characters) limits you.

Table G-1. Default Terminal Definitions

Name	Description	Terminal Types																		
		TTY M33	CDC 713-10, 722, 731-1, 732, 736	CDC 721	IBM 2741	TTY M40-2	HAZ 2000	X3.64 DEC VT100	Tek 4000	X.25 any	HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PREZ	CDC 734	IBM 2780	IBM 3780	IBM 3270
TC	Terminal class	1	2	3	4	5	6	7	8	(0)	9	10	11	12	13	14	15	16	17	18
AB	Abort output block	CAN	CAN	CAN	(CAN	CAN	CAN	CAN	na	na	na	na	na	na	na	na	na	na	na
BF	Blocking factor	1	1	1	1	1	1	1	1	(1)	na	1	1	1	1	na	na	na	na	na
BS	Backspace character	BS	BS	BS	BS	none	BS	BS	BS	(1)	na	na	na	na	na	na	na	na	na	na
B1	User break 1	DLE	DLE	DLE	:	DLE	DLE	DLE	DLE	(1)	:	:	:	:	:	:	:	na	na	:(6)
B2	User break 2	DCA	DCA	DCA)	DCA	DCA	DCA	DCA	(1))))))))	na	na	:(7)
BR	Break as user break 1 (and cancel character)	N	N	N	na	N	N	N	N	(1)	na	na	na	na	na	na	na	na	na	na
CR	Carriage return idles	2	0	0	8	1	0	0	0	(1)	na	na	na	na	na	na	na	na	na	na
CI	Cancel character	CAN	CAN	CAN	(CAN	CAN	CAN	CAN	(1)	(((((((na	na	na
CP	Cursor positioning	Y	Y	Y	na	Y	Y	Y	Y	na	na	na	na	na	na	na	na	na	na	na
CT	Network control character	ESC	ESC	ESC	z	ESC	ESC	ESC	z	(1)	z	z	z	z	z	z	z	z	z	z
DL	Single-message transparent input mode																			
	Character delimiter	CR	CR	CR	KL	CR	CR	CR	CR	(1)	na	(4)	(4)	(4)	(4)	na	(4)	na	na	na
	Character count delimiter	2043	2043	2043	na	2043	2043	2043	2043	(1)	na	na	na	na	na	na	na	na	na	na
	Timeout delimiter	No	No	No	na	No	No	No	No	(1)	na	na	na	na	na	na	na	na	na	na
EB	End-of-block																			
	Character	EOT	EOT	EOT	na	EOT	EOT	EOT	EOT	(2)	na	(4)	(4)	(4)	(4)	na	(4)	na	na	na
	Cursor positioning	CL	CL	CL	na	CL	CL	CL	CL	na	na	na	na	na	na	na	na	na	na	na
EL	End-of-logical line																			
	Character	CR	CR	CR	KL	CR	CR	CR	CR	(1)	na	(5)	(5)	(5)	(5)	na	(5)	na	na	na
	Cursor positioning	LF	LF	LF	na	LF	LF	LF	LF	na	na	na	na	na	na	na	na	na	na	na
EP	Echoplex mode	N	N	N	na	N	N	Y	N	na	na	na	na	na	na	na	na	na	na	na
FA	Full ASCII input mode	N	N	N	na	N	N	N	N	na	N	N	N	N	na	N	na	na	na	na
LC	Input device flow control	N	N	N	na	N	N	Y	N	na	na	na	na	na	na	na	na	na	na	na
IX	Input device	KB	KB	KB	KB	KB	KB	KB	KB	na	KB	KB	KB	KB	na	KB	na	na	na	KB
LI	Line feed idles	1	0	0	1	3	3	0	0	(1)	na	na	na	na	na	na	na	na	na	na
LK	Lockout NOP messages	N	N	N	na	N	N	N	N	(1)	na	na	na	na	na	na	na	na	na	na
OC	Output device flow control	N	N	N	na	N	N	Y	N	na	na	na	na	na	na	na	na	na	na	na
OP	Output device	PR	DI	DI	PR	DI	DI	DI	DI	(1)	na	na	na	na	na	na	na	na	na	na
PA	Parity processing	E	E	E	na	E	E	E	E	(3)	na	na	na	na	na	na	na	na	na	na
PG	Page waiting	N	N	N	na	N	N	N	N	(1)	na	Y	Y	Y	Y	na	Y	na	na	Y
PL	Page length	0	24	30	0	24	27	24	35	(1)	0	13	16	16	16	0	13	0	0	24
PW	Page width	72	80	80	132	80	74	80	74	(1)	80	80	80	80	80	80	80	80	120	80
SE	Special editing mode	N	N	N	na	N	N	N	N	na	na	na	na	na	na	na	na	na	na	na
XL	Multimessage transparent input mode	No	No	No	No	No	No	No	No	na	No	No	No	No	na	No	na	na	na	na

(0) Terminals belong to terminal classes 1 through 3 or 5 through 8.
 (1) Same as for the terminal if it were not connected via a packet-switching network.
 (2) End-of-packet sequence (N bit is reset to zero).
 (3) The parity bit is ignored on input and is generated for output as for the basic terminal type.
 (4) ETX (resulting from the SEND key).
 (5) ESC A (resulting from the CARRIAGE RETURN key).
 (6) Program attention key 1 is a valid user break 1.
 (7) Program attention key 2 is a valid user break 2.

Table G-2. Parameter Ranges for Terminal Definition Commands (Sheet 1 of 2)

Name	Description	Terminal Types																		
		TTY M33	CDC 713-10, 722,731-1, 732,736	CDC 721	IBM 2741	TTY M40-2	HAZ 2000	X3.64 DEC VT100	Tek 4000	X.25 any	HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
TC	Terminal class	1	2	3	4	5	6	7	8	(0)	9	10	11	12	13	14	15	16	17	18
AB	Abort output block	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	na	na	na	na	na	na	na	na	na
BF	Blocking factor	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	(2)	na	0-2	0-2	0-2	0-2	na	0-2	na	na	0-2
BS	Backspace character	(1)	(1)	(1)	(1)	(1),(7)	(1)	(1)	(1)	(2)	na	na	na	na	na	na	na	na	na	na
B1	User break 1	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	(1)
B2	User break 2	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	(1)
BR	Break as user break 1 (and cancel character)	Y,N (Y,N)	Y,N (Y,N)	Y,N (Y,N)	na	Y,N (Y,N)	Y,N (Y,N)	Y,N (Y,N)	Y,N (Y,N)	(2) na	na	na	na	na	na	na	na	na	na	na
CI	Carriage return idles	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	(2)	na	na	na	na	na	na	na	na	na	na
CN	Cancel character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	na	(1)
CP	Cursor positioning	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	(2)	na	na	na	na	na	na	na	na	na	na
CT	Network control character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
DL	Single-message transparent input mode																			
	Character delimiter	0-FF	0-FF	0-FF	6D	0-FF	0-FF	0-FF	0-FF	(2)	na	03	03	03	03	na	03	na	na	na
	Character count delimiter	1-4095	1-4095	na	na	1-4095	1-4095	1-4095	1-4095	(2)	na	na	na	na	na	na	na	na	na	na
	Timeout delimiter	TO	TO	TO	na	TO	TO	TO	TO	(2)	na	na	na	na	na	na	na	na	na	na
EB	End-of-block																			
	Character	(1), EL, EB	(1), EL, EB	(1), EL, EB	na	(1), EL, EB	(1), EL, EB	(1), EL, EB	(1), EL, EB	(2)	na	EB	EB	EB	EB	na	EB	na	na	na
	Cursor positioning	(3)	(3)	(3)	na	(3)	(3)	(3)	(3)	(2)	na	No	No	No	No	na	No	na	na	na
EL	End-of-logical line																			
	Character	(1), EL, EB	(1), EL, EB	(1), EL, EB	na	(1), EL, EB	(1), EL, EB	(1), EL, EB	(1), EL, EB	(2)	na	EB, EL	EB, EL	EB, EL	EB, EL	na	EB, EL	na	na	na
	Cursor positioning	(3)	(3)	(3)	na	(3)	(3)	(3)	(3)	No	na	No	No	No	No	na	No	na	na	na

- (0) Terminals belong to terminal classes 1 through 3 or 5 through 8.
- (1) Any character from the ASCII 128-character set except any lowercase or uppercase alphabetic character, any digit 0 through 9, NULL, SOH, STX, =, DEL, or space.
- (2) Same as for the terminal if it were not connected by a packet-switching network.
- (3) CR, LF, CL, NO.
- (4) KB, PT, SK, X, XK, XP.
- (5) PR, DI, PT.
- (6) Same as for the terminal if it were not connected by a packet-switching network. Parity is ignored on input and is generated for output.
- (7) ASCII BS character code (8 hexadecimal) cannot be used.

Table G-2. Parameter Ranges for Terminal Definition Commands (Sheet 2 of 2)

Name	Description	Terminal Types																		
		TTY N33	CDC 713-10, 722, 751-1, 752, 756	CDC 721	IBM 2741	TTY NA0-2	HAZ 2000	X3.64 DEC VT100	Tek 4000	X.25 any	HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
EP	Echoplex mode	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	na	na	na	na	na	na	na	na	na	na	na
FA	Full ASCII input mode	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	(2)	na	Y,N	Y,N	Y,N	Y,N	na	Y,N	na	na	na
IC	Input device flow control	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	na	na	na	na	na	na	na	na	na	na	na
IN	Input device	(4)	(4)	(4)	KB, XK, X	(4)	(4)	(4)	X	na	X	X	X	X	na	X	na	na	na	na
LI	Line feed idles	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	0-99, CA	(2)	na	na	na	na	na	na	na	na	na	na	na
LK	Lockout messages	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	(2)	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	na	na	Y,N
OC	Output device flow control	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	na	na	na	na	na	na	na	na	na	na	na
OP	Output device	(5)	(5)	(5)	PR, DI	(5)	(5)	(5)	PR, DI	na	na	na	na	na	na	na	na	na	na	na
PA	Parity processing	Z,0, E,N	Z,0, E,N	Z,0	na	Z,0, E,N	Z,0, E,N	Z,0, E,N	Z,0, E,N	(6)	na	na	na	na	na	na	na	na	na	na
PG	Page waiting	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	(2)	na	Y,N	Y,N	Y,N	Y,N	na	Y,N	na	na	Y,N
PL	Page length	0,8-255	0,8-255	0,8-255	0,8-255	0,8-255	0,8-255	0,8-255	(2)	0	0,8-255	0,8-255	0,8-255	0,8-255	0	0,8-255	0	0	0,8-255	
PW	Page width	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	(2)	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	0,20-255	
SE	Special editing mode	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	na	na	na	na	na	na	na	na	na	na	na
TC	Terminal class input	1,2,3, 5-8	1,2,3, 5-8	1,2,3, 5-8	4	1,2,3, 5-8	1,2,3, 5-8	1,2,3, 5-8	1,2,3, 5-8	9	10,15	11-13	11-13	11-13	14	10,15	16	17	18	
XL	Multimessage transparent input mode																			
	Character delimiter	0-FF	0-FF	0-FF	2D	0-FF	0-FF	0-FF	0-FF	(2)	na	03	03	03	03	na	03	na	na	na
	Character terminator	0-FF	0-FF	0-FF	2D	0-FF	0-FF	0-FF	0-FF	(2)	na	na	na	na	na	na	na	na	na	na
	Timeout delimiter	T0	T0	T0	na	T0	T0	T0	T0	na	na	na	na	na	na	na	na	na	na	na
	Character count delimiter	1-4095	1-4095	1-4095	na	1-4095	1-4095	1-4095	1-4095	(2)	na	na	na	na	na	na	na	na	na	na

(0) Terminals belong to terminal classes 1 through 3 or 5 through 8.
 (1) Any character from the ASCII 128-character set except any lowercase or uppercase alphabetic character, any digit 0 through 9, NULL, SOH, STX, =, DEL, or space.
 (2) Same as for the terminal if it were not connected by a packet-switching network.
 (3) CR, LF, CL, MO.
 (4) KB, PT, BK, X, XK, XP.
 (5) PR, DI, FT.
 (6) Same as for the terminal if it were not connected by a packet-switching network. Parity is ignored on input and is generated for output.
 (7) ASCII BS character code (8 hexadecimal) cannot be used.

Faint, illegible text covering the majority of the page, likely bleed-through from the reverse side of the document.



You can transmit input to an application in normalized mode (also called character mode) or transparent mode (also called binary mode). In transparent input mode, the network does not convert any of your input but sends it directly to your application. Characters like LF and BS, which the network normally strips from the input stream, become part of the data sent to your application. Appendix G describes how you delimit and transmit input in transparent mode (refer to the DL and XL commands). The default input mode for all supported terminal classes is normalized mode, however. The remainder of this appendix describes how to delimit and transmit terminal input in normalized mode under the default conditions of the various terminal classes. (Appendix G also describes how you can change these defaults.)

PHYSICAL END-OF-LINE (LINE FEED)

In terminal classes 1 through 8, you can terminate a physical line of input with a line feed key. The network also terminates a physical line of output whenever the page width of a terminal is reached (refer to the PW command in appendix G). The network may respond to the line feed key by advancing the cursor or carriage to the beginning of the next line (refer to the CP command in appendix G).

This key is transparent to TAF. When the terminal is not in block mode, it performs the same function as the carriage return but does not initiate the transaction. Rather, it causes the message to be

stored in the communication block chain until the message transmission key is pressed.

LOGICAL END-OF-LINE (CARRIAGE RETURN)

You terminate a logical line of input with a carriage return key. A logical line can consist of one or more physical lines but not vice versa. The network may respond to the carriage return by advancing the cursor or carriage to the next line (refer to the EL command in appendix G).

MESSAGE TRANSMISSION

You transmit your input to the network with the message transmission key. For non-blockmode devices, the message transmission key is the carriage return key. Hence, for these terminal classes, logical lines are transmitted separately. When the delimiter for the logical end-of-line and transmission key are different (as with block mode devices, including PSN and mode 4 terminals), the terminal stores logical lines until you press the transmission key. It then sends the single transmission consisting of multiple logical lines to the network. The network sends each logical line separately to your application.

Table H-1 shows the keys used by the various terminal classes to perform the three functions just described.

Table H-1. Default Message Delimiter and Transmission Keys

Terminal Class	Function		
	Physical End-of-Line (Line Feed)	Logical End-of-Line (Carriage Return)	Message Transmission Key
1	LINE FEED	RETURN	RETURN
2	↓	RETURN	RETURN†
4	ATTN	RETURN	RETURN
5	NEW LINE	RETURN	RETURN
6	LF	CR	CR
7	LINE FEED	CARRIAGE RETURN	CARRIAGE RETURN
8	LF	RETURN	RETURN
9	Varies††	Varies††	Varies††
10	None	RETURN	SEND
11	None	RETURN	SEND
12	None	NEW LINE	ETX
13	None	NEW LINE	ETX
14	Varies††	Varies††	Varies††
15	None	NEW LINE	SEND
16	None	EM	EOT
17	None	EM	EOT
18	None	Varies	ENTER

†If in block mode, refer to terminal documentation for terminal key equivalences. Those listed are for character and line modes.
 ††Terminals operating under HASP protocol use different keys for this purpose.

DETECTION OF POTENTIALLY BLOCKED TASKS

I

TAF prevents situations from occurring in which the execution of a task is blocked because it is impossible under the circumstances to obtain the needed central memory space for the task, whether for an initial load or a memory increase. Several terms will be defined to aid in the description of how potentially blocked tasks are detected:

- Minimum size of the total task area

The total task area is the CM in which TAF will schedule tasks. It includes the space occupied by CM-resident tasks. The area extends from the first word address into which tasks can be loaded to the maximum field length of TAF. The size of this area fluctuates if the TAF/CRM Data Manager is used. The minimum size of this area occurs when the TAF/CRM buffer takes on its maximum size.

- Minimum size of the transient task area

The transient task area is that portion of the total task area not occupied by CM-resident tasks. The minimum size of this area is defined to be the minimum size of the total task area minus the sum of the MFLs of all CM-resident tasks.

One or more tasks will be considered to be potentially blocked under the following circumstances.

- If the initial field lengths of the CM-resident tasks are so large that they all couldn't be loaded together.
- If the sum of the MFLs for the CM-resident tasks exceeds the minimum size of the total task area.
- If the MFL of a non-CM-resident task exceeds the minimum size of the transient task area.

Following are the situations when potentially blocked tasks are detected and the actions that are taken.

- At TAF initialization. Potentially blocked tasks result in TAF being aborted.
- When TAF is informed, via the LIBTASK TT option, of the update of a task library attached by TAF and the library contains potentially blocked tasks, the update will be rejected.
- When an attempt is made to reduce the MFL of TAF via the K.MAXFL command to a value that results in potentially blocked tasks, the command will be rejected.

The preceding checking assumes that the maximum field length of TAF is set (by default, via K.MFL, or via K.MAXFL commands) to a value that can be attained.

CONFIDENTIAL - SECURITY INFORMATION

1. The following information is being furnished to you for your information only. It is not to be disseminated outside your agency.

2. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

3. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

4. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

5. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

6. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

7. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

8. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

9. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

10. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

11. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

12. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

13. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

14. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

15. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

16. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

17. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

18. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

19. This information is being furnished to you for your information only. It is not to be disseminated outside your agency.

This appendix describes the steps used to access a host and to use CDC network software. The login procedure can be divided into eight steps.

- Gathering information.
- Setting up your terminal.
- Connecting your terminal to the network.
- Identifying your terminal to the network.
- Selecting a host.
- Connecting to the selected host.
- Identifying yourself and your terminal to the host.
- Selecting a network application.

The first part of this appendix gives a brief description of the eight steps. The second part of the appendix gives a more detailed description of the same eight steps followed by additional general information. You may find the brief description of the steps sufficient to accomplish login at your site. If not, consult the detailed description. In addition to more detail, it provides shortcuts, the various options that are available at any given step, and failure conditions.

BRIEF DESCRIPTION OF ACCESS

STEP 1 - GATHERING INFORMATION

Find the answers to the following questions.

- Is your terminal hardwired to the system?
- Which key is the data transmission key for your terminal?
- If you have a dialup terminal, what telephone number must you dial?
- Into which of the following categories does your terminal fall?

Asynchronous terminals using X.25 protocol

CDC mode 4 synchronous terminals
Houston Automatic Spooling Program (HASP) terminals

Bisynchronous terminals like IBM 2780/3780

Bisynchronous terminals like IBM 3270

- Must you supply a family name, user name, password, and application name during login? If so, get the required information.
- Does your terminal use the ASCII character set and code?

STEP 2 - SETTING UP YOUR TERMINAL

- Turn power switch on.
- Load and initialize any software or controlware needed by your terminal.
- Set the terminal's duplex or echoplex switch to the correct position (typically HALF if your keyboard entries appear unduplicated on your console's screen or printer).
- Set the terminal's parity switch to its proper position.
- Set the line speed switch to its proper setting.
- Set the transmission mode to either the character or line mode position.
- Set the online/offline switch to the online position.

STEP 3 - CONNECTING YOUR TERMINAL TO THE NETWORK

Skip this step if your terminal is hardwired. Otherwise, determine which of the configurations in figures J-1, J-2, J-3, and J-4 most closely matches your terminal and follow the instructions in the figure.

1. Turn on coupler.
2. Pick up receiver.
3. Dial phone number.
4. Wait for high-pitched tone.
5. Fit receiver into coupler. Be sure you put the cord end of the receiver into the correct cup, as indicated on the coupler.

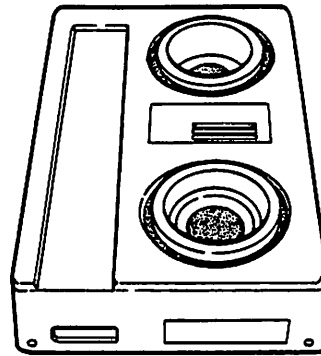


Figure J-1. Separate Acoustic Coupler

1. Pick up receiver.
2. Push button, such as ORIG button on Teletype model 33 or 35.
3. Dial phone number.
4. Wait for high-pitched tone.
5. Push button, such as ORIG button on Teletype model 33 or 35.
6. Replace receiver.

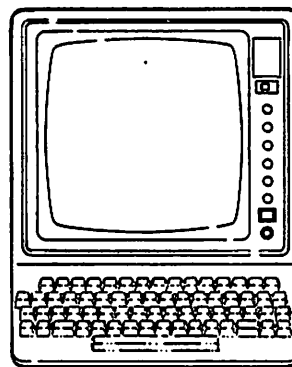


Figure J-2. Data Set Built into the Terminal

1. Pick up receiver.
2. Set switch to TALK or push TALK button.
3. Dial phone number.
4. Wait for high-pitched tone.
5. Set switch to DATA or push DATA button.
6. Replace receiver.

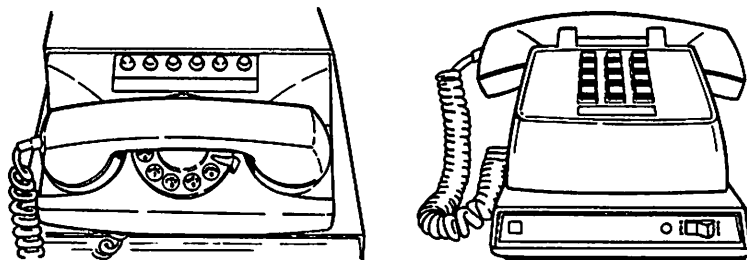


Figure J-3. Data Sets with Switches and Buttons

1. Pick up receiver.
2. Dial phone number.
3. Wait for high-pitched tone.
4. Fit receiver into coupler. Be sure you put the cord end of the receiver into the correct cup, as indicated on the coupler.

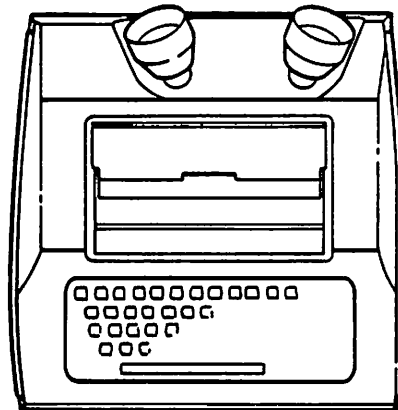


Figure J-4. Terminal with Built-in Acoustic Coupler

STEP 4 - IDENTIFYING YOUR TERMINAL TO THE NETWORK

If the network does not prompt you within a few seconds after you are connected to it (there might be a light on the terminal that indicates when your terminal is connected), perform the appropriate following steps.

- For asynchronous terminals not using X.25 protocol:
 - a. Press the key that transmits a carriage return. The network responds with two line feeds when it recognizes your line speed.
 - b. Type a closing parenthesis (not required if the terminal uses the ASCII character set and code) and press the carriage return key. When the network recognizes your character code set, it responds with a line feed.
- For asynchronous terminals using X.25 protocol:

Get the information from your site.
- For mode 4 synchronous terminals:

Press the data transmission key.
- For HASP and 2780/3780 bisynchronous terminals:

Send a /*CONFIG card image.

- For 3270 bisynchronous terminals:

No identification is necessary.

STEP 5 - SELECTING A HOST

If you have a choice of hosts or host paths (multi-host), the network displays a list of the hosts and host paths that are available and prompts you to make a selection. Otherwise, you can enter:

ct HD

to get a list of hosts. ct is your terminal's network control character (refer to the NOS Reference Set, Volume 3, System Commands).

Make your selection with one of the following commands.

ct HN=node
or
ct HS=hostname

node is the number of a host path; and hostname is the name of a host.

STEP 6 - CONNECTING TO THE SELECTED HOST

If your terminal was not automatically connected to a host, the network prompts you to connect. Enter a carriage return to make the connection.

STEP 7 - IDENTIFYING YOURSELF AND YOUR TERMINAL TO THE HOST

Once your terminal is connected to the host, the host displays a banner similar to:

```
WELCOME TO THE NOS SOFTWARE SYSTEM.  
COPYRIGHT CONTROL DATA 1978, 198X.
```

```
date time terminal name  
site identification NOS 2 PSR level number
```

Following this banner, you may be prompted for a family name, then a user name, then a password, and finally an application name (if there is no default). For each prompt, enter the appropriate information followed by a carriage return.

STEP 8 - SELECTING A NETWORK APPLICATION

If your login is successful, your terminal is placed under the control of the requested or default application, which then prompts you for a command.

DETAILED DESCRIPTION OF ACCESS

The following description covers in more detail the eight steps briefly described earlier. Although each of the steps is always performed when a terminal accesses the system, you might find that your network's administrators can make many of these steps unnecessary by having the software perform the steps for you.

STEP 1 - GATHERING INFORMATION

If you know the answers to the following questions, or can get the answers from people at your site, the text following each question will tell you which steps and corresponding subsections of this appendix you can skip. If you cannot answer the questions, read all of the subsections; each procedure description contains hints for answering the questions.

1. Is the terminal hardwired, or is it a dial-up terminal?

A hardwired terminal is connected directly to the network through a device called a modulator/demodulator (modem). A modem can be either a separate box, or built into the terminal. A dial-up terminal temporarily connects to the network through a telephone line and either an acoustic coupler or a telephone data set (a coupler or data set does the same things as a modem).

If you do not need to dial a telephone number to use the network, then your terminal is hardwired. (A nearby telephone does not necessarily mean the terminal is dial-up.)

2. Is your network configured for automatic recognition?

Your site can ask the network software to determine certain things about your terminal when the terminal joins the network. This process is called automatic recognition. [Terminals using X.25 packet-switching networks (PSN) cannot use automatic recognition.]

If your network is configured for automatic recognition, you must perform an extra step each time you connect your terminal to the network. If your terminal is not so configured, you may skip questions 3 and 4.

3. What protocol does your terminal use?

Your terminal must fit into one of the following categories.

- Asynchronous terminals that work like teletypewriters or IBM 2741 terminals.
- Asynchronous using an X.25 PSN through a packet assembly/disassembly (PAD) service; these terminals work like teletypewriters.
- CDC mode 4 synchronous; mode 4A terminals work like the CDC 200 User Terminal, while mode 4C terminals work like CDC 711 or 714 terminals.
- HASP terminals.
- Bisynchronous terminals that work like IBM 2780, 3270 or 3780 terminals.

If your terminal operating manual does not tell you what protocol your terminal uses, question 4 and the terminal classes shown in the NOS Reference Set, Volume 3, System Commands might help you answer this question. If your network is not configured for automatic recognition, skip question 4.

4. What class of terminal is it?

Many different kinds of terminals exist. The network software assigns terminals it supports to classes. If you know what the terminal class is, that will tell you the protocol, as follows:

<u>Class</u>	<u>Protocol</u>
1 thru 8	Asynchronous or X.25 asynchronous
9 and 14	HASP bisynchronous
10 and 15	CDC mode 4A synchronous
11 thru 13	CDC mode 4C synchronous
16 and 17	IBM 2780 and 3780 bisynchronous
18	IBM 3270 bisynchronous
Other	Site-defined

Terminals in classes 1 through 8 sometimes are called interactive terminals. A console has an input and output device that can be used for dialog with the network. Most consoles use a television-like screen or a typewriter-like mechanism to convey output information. Almost all consoles have a keyboard for entering information to the network. Some consoles might also use mechanisms such as magnetic tape cassettes or paper tape reader/punches for input or output.

Terminals in classes 9 through 18 sometimes are called remote batch terminals or remote job entry terminals. Such terminals have an interactive console and can also have batch devices the network can use for input only or for output only.

Figure J-5 shows a console with a display screen and a keyboard. Figure J-6 shows a console with a printer and a keyboard. Figure J-7 shows a remote batch terminal with a console, card reader, and a batch printer device.

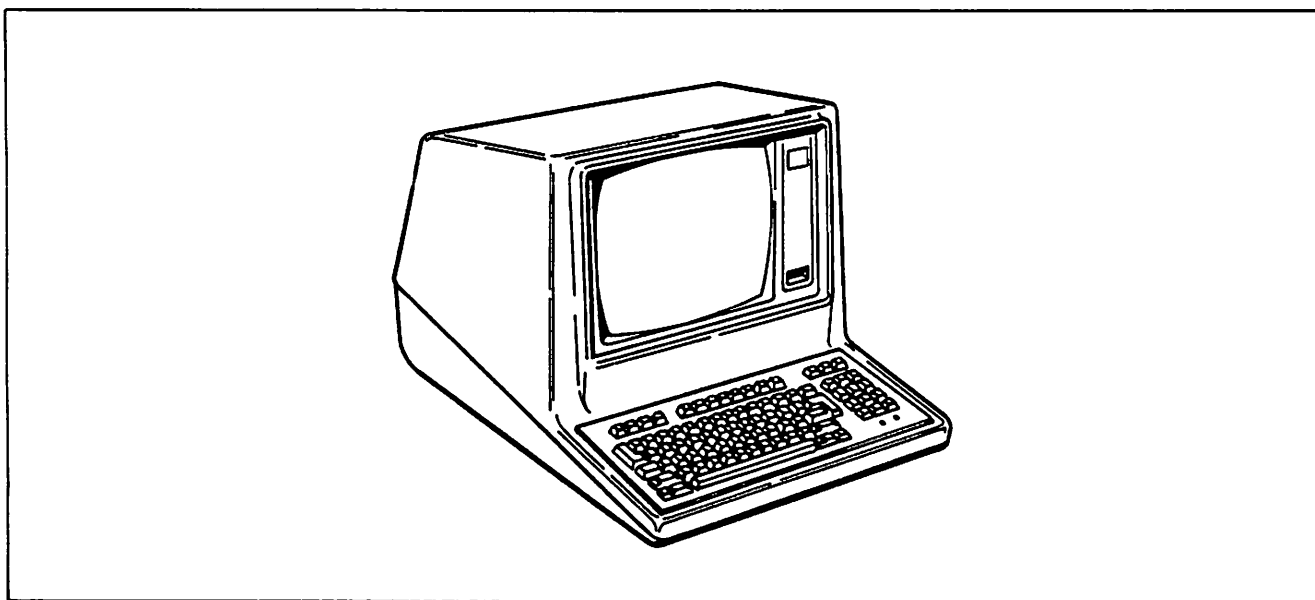


Figure J-5. Interactive Display Console

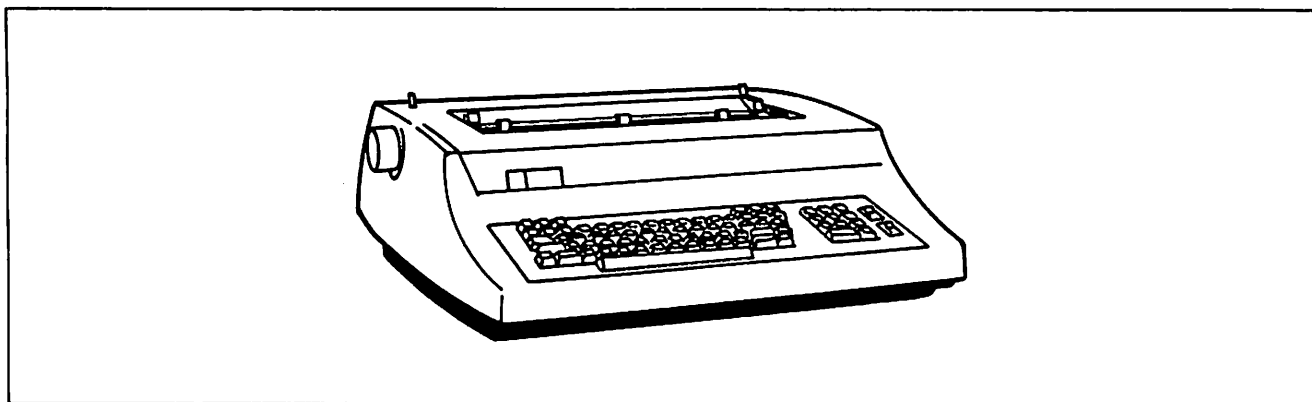


Figure J-6. Interactive Printer Console

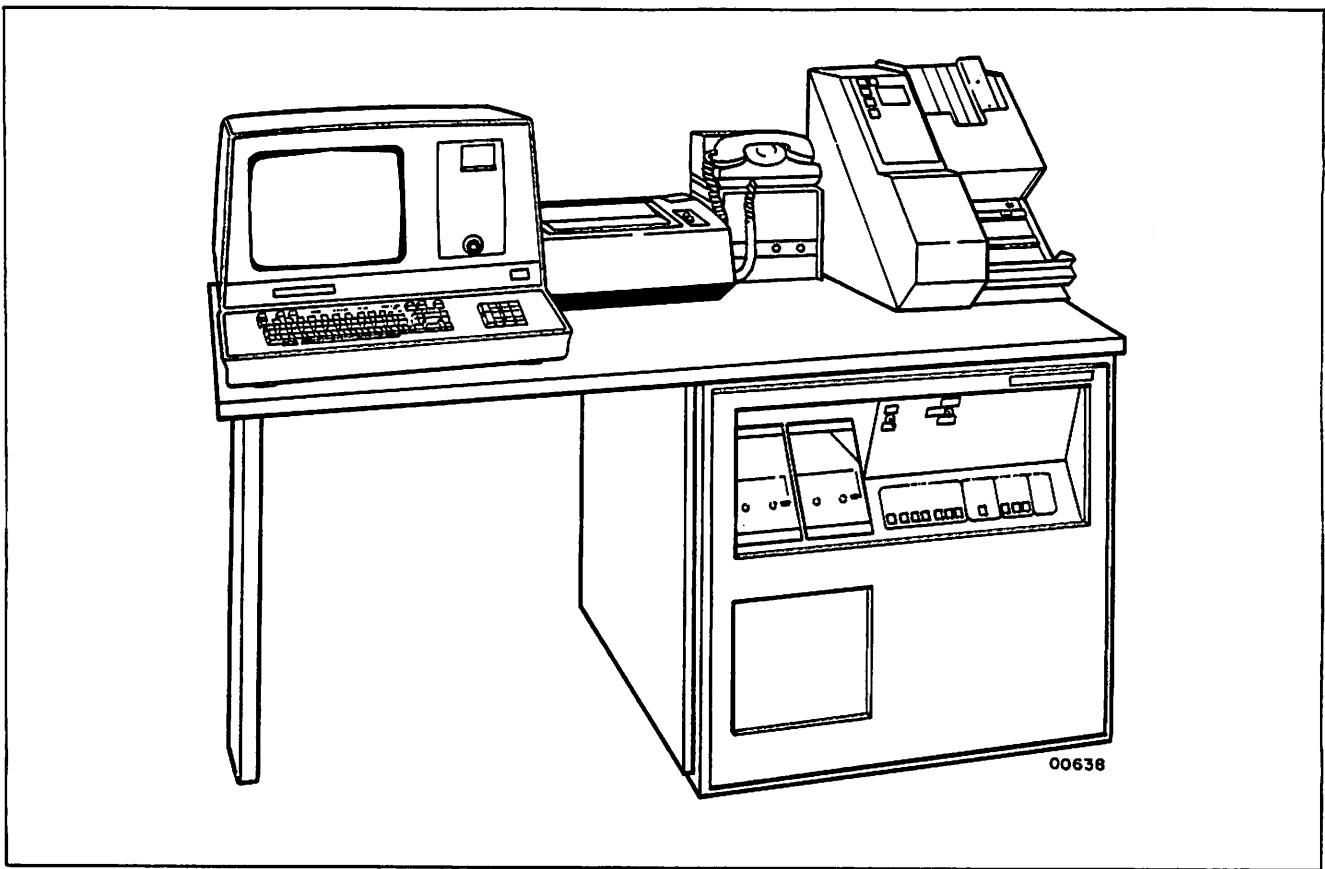


Figure J-7. Remote Batch Terminal

5. What key ends a line of data? What key sends data? (Does the terminal use block mode?)

You should know when data leaves your terminal so that you know when to expect a response from the network or the host computer. Your terminal operator's guide should tell you what key or keys send data from your terminal.

Terminals transmit information in one of the following ways.

- A character at a time (transmission occurs as you enter each character).
- A line at a time (transmission might occur when you end each line).
- Several lines at a time, called a block (transmission occurs when you direct the terminal to transmit).

Terminals that send data a character at a time are treated by the network as if they sent characters a line at a time. Until you end the line by entering a character or set of characters called an end-of-line character, you can backspace within the line and change data. When you enter the end-of-line character, the network processes the data.

The end-of-line character used can be changed by you, by a site administrator, or by an application. We represent the end-of-line character for such terminals by a carriage return.

This appendix assumes you enter an end-of-line character after each command.

Terminals that send data a character at a time usually send the end-of-line character when you use a key called the end-of-line key (remember, the character can be changed). The end-of-line key might be labeled CARRIAGE RETURN, RETURN, RET, NEWLINE, NEXT, ATTN, or CR.

Terminals that transmit a line at a time usually send the end-of-line character and transmit the line when you use the end-of-line key. If the character is changed, then you must enter the character as the last entry on the line before pressing the end-of-line key. The end-of-line key might be labeled CARRIAGE RETURN, RETURN, RET, NEWLINE, NEXT, ATTN, or CR.

Terminals that send data a block at a time do not transmit your data to the network when you press an end-of-line key. Until you press a key called the end-of-block key, you can position the cursor anywhere within

any unsent line and change data. This key might be labeled SEND, ETX, EOT, or XMIT. When you press the end-of-block key, the network processes each line of data as if it had been transmitted one line at a time.

The network expects each block from a block mode terminal to end with a character or set of characters called the end-of-block indicator. The end-of-block indicator used can be changed by you, by a site administrator, or by an application program. We do not show the end-of-block indicator in the examples.

Terminals that transmit a block at a time usually send the end-of-block character and transmit all lines in the block when you use the end-of-block key. If the character is changed, then you must enter the character at the end of the last line before pressing the end-of-block key.

Terminals that transmit blocks are sometimes said to work in batch mode; do not confuse this kind of batch mode operation with remote batch data transmissions. Remote batch data transmissions are simply transmissions of files from or to the host using batch devices, bypassing the terminal.

6. What telephone number is appropriate?

If your terminal is hardwired, you may skip questions 6 and 7.

Your network might use a specific telephone number for terminals that communicate at a certain speed, for terminals operating in block mode, or for all terminals using certain protocols.

The telephone number you use determines whether you need to complete some of the procedures. If you do not know what numbers you should use, get help from a site administrator.

7. Is your terminal automatically connected to a host?

If it is, skip question 8.

8. What is the name or node number for the host you need to use?

9. Will your terminal be automatically logged in?

If so, skip question 10.

10. What is the family name, user name, and password appropriate for you on the host system you will be using?

You can get this information from site administrators.

11. Is your terminal automatically connected to a network application?

If so, you will not have to select an application at login.

STEP 2 - SETTING UP YOUR TERMINAL

If your terminal is hardwired, you can probably skip most of this subsection; hardwired terminals are usually set up once and left in the appropriate condition. Before using this subsection, review the applicable terminal operator's manual, which describes the procedures for setting up your terminal. If you are using an X.25 terminal, follow the PAD access procedures supplied by the packet-switching network service.

Switch settings are important. The number, type, settings, and names of switches vary from terminal to terminal. Some terminals use buttons or toggles for switches; others use software option selections as switches.

If other people have used your terminal to connect to the network, the switches should be set correctly; do not change them. Directions for setting the switches may be posted near your terminal.

Here is a suggested procedure to follow when setting up your terminal:

1. Turn on the power switch.
2. Load and initialize any software or controlware needed by your terminal. You will find detailed information on this task in your terminal's operator manual.
3. Set the terminal's duplex or echoplex switch to the correct position (use HALF if you are not sure). If you wish to have the display of your login password suppressed, you must use full duplex. If nothing you enter appears on your console after you connect to the network, the network must display characters for your console. The NOS Reference Set, Volume 3, System Commands describes how to enable and disable character echoing by the network (refer to the EP command).
4. Set the terminal's parity switch to the proper position. If garbled output appears at your console after you connect to the network, the network must use a different parity choice for your terminal. The NOS Reference Set, Volume 3, System Commands describes how to change parity use by the network (see the PA command).
5. Set the line speed switch to a speed that matches one associated with the telephone numbers you were given. If you do not know the line speed you should use, set the switch to any position for X.25, mode 4 synchronous, or bisynchronous terminals; set the switch to 30 characters per second (300 bits per second) for asynchronous terminals.
6. Set the transmission mode switch to either the character or line mode position (you can change it later if you have a block mode terminal). For example: on a Viking 721, select the CHAR option; on a CDC 751, set the switch to the CHAR position; on a CDC 200 User Terminal, set the switch to the LINE position.

7. Set the online/offline switch to the position that permits online communications. For example: on a Viking 721, select the ON option of the LINE setting; on a CDC 713, turn off the LOCAL indicator switch; on a Teletype, set the LINE/OFF/LOCAL switch to the LINE position; on a CDC 200 User Terminal console, set the ATTENDED/UNATTENDED switch to the ATTENDED position.

You are now ready to connect your terminal to the network.

STEP 3 - CONNECTING YOUR TERMINAL TO THE NETWORK

To begin communication, your terminal must be physically connected to the network. If your terminal is hardwired, skip the remainder of this subsection; this subsection applies only to dial-up terminals.

A dial-up terminal uses either an acoustic coupler (asynchronous terminal) or a data set (asynchronous, X.25, bisynchronous, or synchronous terminal) to link itself to the network.

An acoustic coupler can be built into the terminal. Figure J-4 shows a terminal with a built-in acoustic coupler and gives directions to connect the terminal to the network.

An acoustic coupler can also be separate from the terminal. Figure J-1 shows an acoustic coupler separate from a terminal and gives directions to connect the terminal to the network.

A data set can be built into the terminal. Figure J-2 shows a terminal with a data set built in and gives directions to connect the terminal to the network.

A stand-alone data set can have either a switch labeled TALK and DATA or individual buttons labeled TALK and DATA. Figure J-3 shows both a data set with switches and a data set with buttons and gives directions to connect the terminal to the network.

After you dial the phone number, either you get a high-pitch tone or an operator answers. If an operator answers, ask for your terminal to be connected, wait for the tone, and then place the receiver in the coupler or set the data set switch or button to the on or data position. If you get a busy signal, wait and redial or try another number.

If your terminal has an indicator (sometimes marked DSR, DATA SET READY, or SYSTEM ACTIVE), it lights to let you know that the terminal is connected to the network. Wait approximately 2 seconds after the light comes on before entering data.

You are now ready for the next step. You might need to do one or all of the following:

- Identify your terminal to an X.25 network (this appendix does not describe this procedure).
- Identify your terminal to the CDC network.
- Select a host.

- Connect to the selected host.
- Identify yourself and your terminal to the host (log in).
- Select a network application.

The software usually sends you a message that helps you decide which step comes next. If nothing appears at your terminal within a few seconds, you probably need to identify the terminal to the network.

STEP 4 - IDENTIFYING YOUR TERMINAL TO THE NETWORK

Once you have established physical connection to the network hardware, you might need to identify the terminal to the network software.

This identification procedure, called automatic recognition, is not always needed. It does not apply to X.25 terminals. It does not apply if the communication line used by your terminal is not configured for automatic recognition.

If your terminal does not display any information within a few seconds of physical connection, you should complete one of the identification procedures described in the following subsections. The procedure you use depends on the terminal protocol, which you can determine if you know the terminal's type and/or class.

If you do not complete the identification procedure within the allowed time, the network disconnects a dial-up terminal. Hang up the phone and redial the number if you are using a dial-up terminal. You must restart the procedure for a hardwired terminal.

Procedure for Asynchronous Terminals

For an asynchronous terminal, which belongs in terminal classes 1 through 8, complete this procedure within 1 minute:

1. Wait about 2 seconds after the light that indicates your terminal is connected to the network comes on.
2. Press the carriage return key to identify the line speed used by your terminal. The network software responds with two line feeds.
3. If your terminal uses an APL, EBCD, or correspondence code character set, type a closing parenthesis; if your terminal uses an ASCII character set, you need not type in anything.
4. Press the carriage return key to identify the character and code set used by the terminal. The network software responds with a line feed.

The software sends you a message that helps you decide which step comes next. Turn to the subsection appropriate for the next step you must perform.

Procedure for HASP Terminals

HASP terminals belong either to terminal class 9 or 14. For automatic recognition of your HASP terminal, you might need to modify the signon block or enter a special statement.

The first information transmitted by a HASP terminal after it is connected to the network is called a signon block. A terminal might handle signon block transmission in one of several ways.

- Some terminals automatically transmit a signon block that begins with the characters:

```
/*SIGNON
```

These eight characters and the information following them imitate the card required by some IBM host systems for terminal identification. This transmission can occur without your knowledge or intervention if the information is built into the firmware or hardware.

- Some terminals require you to enter information for the signon block before they will transmit anything to the network; however, they only allow the block to contain information beginning with the eight characters /*SIGNON. These characters and other information might be required as:

-A command from the console.

-A card read from any card reader.

-A card read from a specific card reader.

- Some terminals require you to enter signon block information before they will transmit anything to the network, but the content of the signon block is left up to you. Whether the information can be entered from the console or from a card reader depends on the terminal.

The network software cannot use information from a /*SIGNON card image. When such a card image is received in a signon block, the signon block contents are discarded.

Instead, the network software uses a /*CONFIG statement for the automatic recognition procedure. You can send the /*CONFIG statement in the signon block when your terminal permits you to provide the contents of that block.

If the signon block does not contain the /*CONFIG statement and the terminal must use automatic recognition, a prompt requesting that statement is sent to the terminal console. When a prompt for the statement appears, enter the statement through a card reader.

The format of the /*CONFIG statement is:

```
/*CONFIG,ti,CO=co,CR=x,LP=x,CP=y,PL=z
```

Blanks are not allowed within the statement; a blank ends parameter processing. The parameters, which you can specify in any order, are:

Parameter

Description

ti

Terminal type:

POST HASP postprint (default); this is terminal class 9.

PRE HASP preprint; this is terminal class 14.

CO=co

Configuration ordinal, a decimal integer from 1 to 255; the default is 1. Use the configuration ordinal to select one of several terminal definitions defined for the line that describes a specific combination of additional characteristics. Site administration personnel can tell you the correct number to use.

CR=x

Identifies available card readers. The default is 1.

LP=x

Identifies available line printers. The default is 1.

CP=y

Identifies available card punches; y cannot equal z. The default is 1.

PL=z

Identifies available plotters; y cannot equal z. No default exists for z; you cannot omit PL=z if you have a plotter.

x, y, or z is a list of numbers specified by either:

```
1/2/ . . . /7
```

or:

ALL (all numbers from 1 through 7)

The numbers you use for x, y, and z must match the HASP stream numbers used for the corresponding devices within the workstation. Site administration personnel should have this information if you do not know what numbers to use.

For example:

```
/*CONFIG,PRE,CO=2,CR=2,LP=2/3,CP=4,PL=5
```

identifies your terminal as the second preprinting workstation defined for the communication line, having:

- A card reader and line printer on stream 2.
- Another printer on stream 3.
- A card punch on stream 4.
- A plotter on stream 5.

If no prompt for a /*CONFIG statement appears, you are ready for the next step.

The software sends you a message that helps you decide which step comes next. Turn to the subsection appropriate for the next step you must perform.

Procedure for Mode 4 Terminals

Mode 4 terminals belong to terminal classes 10 through 13 and 15. For automatic recognition, press the end-of-block key within 1 minute of physical connection.

You are ready for the next step. The software sends you a message that helps you decide which step comes next. Turn to the subsection appropriate for the next step you must perform.

Procedure for Bisynchronous Terminals

Bisynchronous terminals belong to terminal classes 16, 17, and 18. Enter a /*CONFIG statement to specify the type of terminal and the devices available at it.

The format of the /*CONFIG statement is:

```
/*CONFIG,ti,CO=co,CR,LP,CP=y
```

Blanks are not allowed within the statement; a blank ends parameter processing. The parameters, which you can specify in any order, are as follows:

<u>Parameter</u>	<u>Description</u>
ti	Terminal type: 2780 IBM 2780 (default); this is terminal class 16. 3780 IBM 3780; this is terminal class 17. 3270 IBM 3270; this is terminal class 18.
CO=co	The configuration ordinal, a decimal integer from 1 through 255; the default is 1. Use the configuration ordinal to select one of several terminal definitions defined for the line that describes a specific combination of additional characteristics. Site personnel can tell you the correct number to use.
CR	Indicates an available card reader; assumed if not specified.
LP	Indicates an available line printer; assumed if not specified.
CP=y	Indicates an available card punch; assumed not to exist if not specified. The =y portion is not allowed for 2780; for 3270 or 3780, the y portion is required and can be either 2 or 3 to correspond to the device selection character DC2 or DC3.

For example:

```
/*CONFIG,3780,CO=4,CR,LP,CP=2
```

identifies your terminal as a 3780 terminal having:

- A card reader and line printer.
- A card punch, which is selected by the character code DC2.

If no prompt for a /*CONFIG statement appears, you are ready for the next step.

The software expecting the next step sends you a message which helps you decide which procedure comes next. Turn to the subsection appropriate for the next step you must perform.

STEP 5 - SELECTING A HOST

You might need to select the host, which runs the software you want to use. This is not necessary if your installation personnel select a host path for your terminal when the network is configured.

You can select or change a site-defined host path after you connect your terminal to the network. The host path selected is used until either you change it or until your terminal is disconnected from the network.

More than one host path can be available to each host from a terminal. If one host path fails, you can select an alternate route to the host.

Selecting a Host

You can select a host or override a site-defined host selection with the following terminal definition command.

```
ct HS=hostname
```

ct represents the network control character for your terminal. For most asynchronous terminals this is ESC or %.

Each host may be configured with a unique name. This name (1 through 7 characters) is indicated by hostname.

For example:

```
%HS=ARHNOS
```

would select the path with the least traffic to the host called ARHNOS (% represents your terminal's network control character).

If you specify the name, your terminal uses the host path with the least traffic.

If you do not know the name you need, the host availability display (HAD) can show you what names are available. That display is described later in this appendix.

Once you have selected your host, you are ready for the next step.

Selecting a Specific Host Path

Use the following terminal definition command to select your own host path or to override the site-defined host path.

```
ct HN=nn
```

ct represents the network control character for your terminal. For most asynchronous terminals this is ESC or %.

Each access has a unique number, called a node number. The host node number (1 through 31) is indicated by nn.

For example:

```
%HN=2
```

would select access 2 (% represents your terminal's network control character).

If you omit the node number, your terminal uses the host path the network is using to communicate with its supervising host.

If you do not know the node number you need, the host availability display (HAD) can show you what numbers are available. That display is described in the following subsection. Each node number may have a name for its host, provided by your site to help you identify the corresponding host system on the HAD.

Once you have selected the host path, you are ready for the next step.

The software sends you a message that helps you decide which step comes next. Turn to the subsection appropriate for the next step you must perform.

Controlling the Display of Host Paths

The host availability display (HAD) (figure J-8) lists all host paths your terminal can currently use to gain access to the host and all hosts in the network. You can turn display of the HAD on and off or request the display with the following terminal definition command.

```
ct HD=option
```

ct represents the network control character for your terminal. For most asynchronous terminals this is ESC or %.

If the display mode is N (no), you receive only two lines at your terminal (the host status message and the prompt message) instead of the full display when you disconnect your terminal from a host.

If you enter HD=Y or HD, the full display is issued immediately when you disconnect your terminal from the host. You then have a choice of hosts and/or host paths to which you can request connection.

If you have a multihost system, typically the HAD appears when you first access the network. This appearance is controlled by site installation.

Display			
host status message		CONTROL CHARACTER = ct	
NPU NODE= xx	TERMINAL NAME= yyyyyy		
HOST	NODE	SELECTED/ CONNECTED	STATUS
host ₁	node ₁	condition ₁	status ₁
host ₂	node ₂	condition ₂	status ₂
.	.	.	.
.	.	.	.
host _n	node _n	condition _n	status _n
prompt message			

Figure J-8. Host Availability Display (Sheet 1 of 2)

<u>Field</u>	<u>Description</u>																											
host status message	Status of a given host, which can be one of the following: <table border="1"> <thead> <tr> <th><u>Message</u></th> <th><u>Meaning</u></th> <th><u>You Also See:</u></th> </tr> </thead> <tbody> <tr> <td>HOST CONNECTED</td> <td>You are connected to the host.</td> <td>prompt message d or e</td> </tr> <tr> <td>NO HOST SELECTED</td> <td>Host is available; you have not selected a host.</td> <td>prompt message b</td> </tr> <tr> <td>NO HOST AVAILABLE</td> <td>No hosts are available.</td> <td>prompt message a</td> </tr> <tr> <td>HOST UNAVAILABLE</td> <td>You have selected a host that is not up.</td> <td>prompt message a</td> </tr> <tr> <td>HOST BUSY</td> <td>Connection rejected by host.</td> <td>prompt message a</td> </tr> <tr> <td>HOST DISCONNECTED</td> <td>Connection to host terminated.</td> <td>prompt message c or e</td> </tr> <tr> <td>HOST AVAILABLE</td> <td>You have selected host that is up.</td> <td>prompt message c</td> </tr> <tr> <td>NO HOST CONNECTED</td> <td>You are not connected to a host.</td> <td>prompt message c</td> </tr> </tbody> </table>	<u>Message</u>	<u>Meaning</u>	<u>You Also See:</u>	HOST CONNECTED	You are connected to the host.	prompt message d or e	NO HOST SELECTED	Host is available; you have not selected a host.	prompt message b	NO HOST AVAILABLE	No hosts are available.	prompt message a	HOST UNAVAILABLE	You have selected a host that is not up.	prompt message a	HOST BUSY	Connection rejected by host.	prompt message a	HOST DISCONNECTED	Connection to host terminated.	prompt message c or e	HOST AVAILABLE	You have selected host that is up.	prompt message c	NO HOST CONNECTED	You are not connected to a host.	prompt message c
<u>Message</u>	<u>Meaning</u>	<u>You Also See:</u>																										
HOST CONNECTED	You are connected to the host.	prompt message d or e																										
NO HOST SELECTED	Host is available; you have not selected a host.	prompt message b																										
NO HOST AVAILABLE	No hosts are available.	prompt message a																										
HOST UNAVAILABLE	You have selected a host that is not up.	prompt message a																										
HOST BUSY	Connection rejected by host.	prompt message a																										
HOST DISCONNECTED	Connection to host terminated.	prompt message c or e																										
HOST AVAILABLE	You have selected host that is up.	prompt message c																										
NO HOST CONNECTED	You are not connected to a host.	prompt message c																										
ct	Network control character currently defined for your terminal.																											
xx	The number of the host node you are connected to.																											
yyyyyy	The terminal number of your terminal.																											
host;	The 1- to 7-character name of a host; if the network only has one host, this can be blank.																											
node;	The host node number used in selecting a path through the network; $1 \leq \text{node}_i \leq 31$.																											
condition;	S = selected, not connected C = connected, not selected SC = selected and connected SA = selected, attempting connection																											
status;	Either AVAILABLE or NOT AVAILABLE; a host is only available when it is connected to the network.																											
prompt message	Action to take in response to a host status message, which can be one of the following: <table border="1"> <tbody> <tr> <td>prompt message a:</td> <td>ENTER ct HD TO SEE HOST STATUS</td> </tr> <tr> <td>prompt message b:</td> <td>ENTER ct HN=nn TO SELECT HOST</td> </tr> <tr> <td>prompt message c:</td> <td>ENTER INPUT TO CONNECT TO HOST</td> </tr> <tr> <td>prompt message d:</td> <td>READY FOR INPUT</td> </tr> <tr> <td>prompt message e:</td> <td>TERMINAL DISABLED BY NOP</td> </tr> <tr> <td>prompt message f:</td> <td>INPUT DISCARDED</td> </tr> </tbody> </table>	prompt message a:	ENTER ct HD TO SEE HOST STATUS	prompt message b:	ENTER ct HN=nn TO SELECT HOST	prompt message c:	ENTER INPUT TO CONNECT TO HOST	prompt message d:	READY FOR INPUT	prompt message e:	TERMINAL DISABLED BY NOP	prompt message f:	INPUT DISCARDED															
prompt message a:	ENTER ct HD TO SEE HOST STATUS																											
prompt message b:	ENTER ct HN=nn TO SELECT HOST																											
prompt message c:	ENTER INPUT TO CONNECT TO HOST																											
prompt message d:	READY FOR INPUT																											
prompt message e:	TERMINAL DISABLED BY NOP																											
prompt message f:	INPUT DISCARDED																											

Figure J-8. Host Availability Display (Sheet 2 of 2)

STEP 6 - CONNECTING TO THE SELECTED HOST

Your site personnel might have configured your terminal so that it is automatically connected with a host. If so, the network continues to attempt connection indefinitely either until the connection is completed or until you stop it by entering `CT`. You can tell the network is attempting to connect you when the HAD shows a SA status for any host. If your terminal is automatically connected to a host, you may skip this subsection. If not, you will have to ask for the connection to be made.

You can ask to be connected to the selected host by entering a carriage return. If the carriage return is not preceded by an empty line, the information is discarded (unless it is a valid terminal definition command).

If your connection is rejected, you are notified. If you select another host path you must make another entry to complete a connection. The network software physically disconnects dial-up devices if you do not attempt connection to a host within 2 minutes of completing the previous step.

Once you are connected to a host, you are ready for the next step.

STEP 7 - IDENTIFYING YOURSELF AND YOUR TERMINAL TO THE HOST

The step that identifies you and your terminal to the host system is called login. Your terminal is identified to the host system by assigning it a family name, a user name, and a password. The host can prompt you for each of these things.

Your site can set up your terminal connection so that part or all of your login is done automatically. If so, you might not see the corresponding prompt and you need not enter the information.

Your site might also require use of a secure login initiation sequence, prior to login, to ensure that you send your user identification to the correct host software. If required, the sequence contains a site-selected security character and the sequence will be recognized by the system at any time. Depending on your terminal type, the secure login initiation sequence is entered in one of the following ways.

- At an asynchronous terminal (not 2741), press the interactive interrupt key (BREAK or ATTN) and type the character at a 2741 terminal, press the NL key twice. Then enter the ATTN key, the character, and press the NL key.
- At an X.25 terminal, press the interactive interrupt key, type the character, and press the message transmission key.
- At a mode 4 terminal, clear the screen, press SEND, clear the screen again, type the character, and press the end-of-block key.
- At a HASP terminal, type the character and press the end-of-line key.
- At a 2780 or 3780, enter `/*` and the character from the console or a card reader.
- At a 3270, clear the screen, press the program attention 1 key, type the security character, and press the ENTER key.

This action momentarily disconnects you from the host. You might need to reconnect your terminal to the host (if connection is not automatic). Now you are guaranteed that you are sending your login entries to the correct host software.

Refer to figure J-9 for a sample login from an asynchronous terminal. In the example, the terminal has not been configured by site personnel for automatic login. Everything you would typically enter is shown in lowercase.

```
WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, 198X.

85/03/01. 09/51.06. T05A50
CDC NETWORK OPERATING SYSTEM NOS 2
FAMILY:
USER NAME: jrc3069
PASSWORD: janc
/T05A50 - APPLICATION: iaf
```

Figure J-9. Sample Login from an Asynchronous Terminal

Automatic Login

The family name or user name assigned to your terminal for an automatic login can be:

- A mandatory value.
- A default value.
- A primary value.

If a mandatory value is assigned, you do not receive a prompt and cannot use any other value.

If a default value is assigned, you are prompted and you can either: enter an empty line as a response to the system prompt if you want either the default family or user name to apply, or respond to the prompt with another value.

The default family name value assigned to your terminal for automatic login can be different from the default value used by the host for a family name. You can override the login default even when you do not know the system default; this procedure is described in the following subsection.

If a primary value is assigned, you do not receive a prompt the first time login occurs while you are connected to the host. Subsequent logins (refer to Restarting Login Identification) do prompt you. You can override a primary value for user name by entering a different value during an abbreviated login. Abbreviated login is described in a later subsection.

Login Dialog

Login begins when the system displays the following lines.

```
WELCOME TO THE NOS SOFTWARE SYSTEM.  
COPYRIGHT CONTROL DATA 1978, 198X.
```

Then a line appears with the date, time, and terminal name; for example:

```
83/12/28. 15.23.58. TERM201
```

Next is a line your site supplies that identifies the host system used; something similar to:

```
CDC NETWORK OPERATING SYSTEM NOS 2
```

The next line is either of the following prompts.

FAMILY:

This indicates that no mandatory or primary family name has been assigned to your terminal. Enter the 1- to 7-character name of the storage device that contains your permanent files and press the carriage return. If you want to use the default login family name, just press the carriage return after the FAMILY prompt appears. You can also enter a value of 0 to override a preassigned login default family name and use the system default family name. If you have no preassigned default family name, carriage return selects the system default family name.

USER NAME:

This indicates that no mandatory or primary user name has been assigned to your terminal. Enter the user name you were given. The user name, which can contain any combination of digits, letters, and asterisks, identifies you as the terminal operator.

The next prompt appears only after a USER NAME prompt has appeared. This final prompt is either:

PASSWORD:

or

```
PASSWORD:  
XXXXXXX
```

The system attempts to preserve your password's secrecy. If you have set your terminal at full duplex (refer to the NOS Reference Set, Volume 3, System Commands), your password will not be displayed when you enter it. You will not see what you type in. At some terminals, the network overprints several characters on a line and asks you to type your password on that line, as shown by the second PASSWORD prompt. After creating this row of overstruck characters, the cursor moves back to the first character.

Enter the password currently associated with your user name. If you must enter the user name, you also must enter the password. No default exists for a password.

You must respond to each prompt within 2 minutes. If you take too long to respond, this message is displayed at your terminal.

TIMEOUT.

Your terminal is then disconnected from the host.

If you enter an unacceptable family name, user name, or password, you receive either the following message:

IMPROPER LOGIN, TRY AGAIN.
FAMILY:

if no mandatory family name is assigned to your terminal, or:

IMPROPER LOGIN, TRY AGAIN.
USER NAME:

if a mandatory family name is assigned. Reenter all needed login information, beginning with the requested parameter, regardless of when the error occurred.

You must spell each login entry correctly. If you make a mistake, such as typing the letter O for the numeral 0 or the number 1 for the letter l, your login will not be successful. Try again when the prompting sequence restarts.

You are allowed four consecutive chances to complete the login procedure. If you make four unsuccessful attempts to complete login, this message appears at your terminal:

USER RETRY LIMIT.

Your terminal is then disconnected from the host. If this happens, check the spelling of all of the entries. If you spelled everything correctly, contact a site administrator.

When the family name, user name, and password are accepted by the host, you must next connect your terminal to an application, if there is no default network application.

STEP 8 - SELECTING A NETWORK APPLICATION

Your site may be set up so that your terminal is connected to the IAF application automatically. Or your site may connect your terminal automatically to another application, depending on the installation. If so, you might not see the corresponding prompt. If not, you must complete this step yourself.

You are allowed several chances to make a correct entry. However, if you fail to connect successfully after four consecutive attempts, your terminal is automatically disconnected from the host. If this happens, check the spelling of the name you used. If the spelling was correct, contact a site administrator.

Automatic Connection

Your host may permit you access only to one network application, or an application can be connected

automatically. Your host site can configure your connection for either possibility by preassigning an automatic connection application name.

A preassigned automatic application connection can be either mandatory or primary for the connection or it can be mandatory for the user. If either you or your connection has a preassigned mandatory application, you are not prompted, you are automatically connected to that application, and you cannot use any other application. If a mandatory application program is not running in the host, you are logged out and disconnected from that host.

If a primary value is assigned, you are not prompted for the initial connection request after identifying your terminal to the host, and connection to that program occurs automatically. If that connection is not successful or if you disconnect from that application program, you have a choice of actions; you can select another application or disconnect from the host.

To override a primary automatic application connection on the first attempt to select an application, name a different application in an abbreviated login entry (described later in this appendix) following a login prompt.

When a primary value is assigned, you are prompted on subsequent attempts to select an application, such as when you are switching applications. When you are prompted to enter an application name and you want to be connected to your primary application, enter an empty line as a response.

Manual Connection

This procedure begins when you see the following prompting message.

terminalname - APPLICATION:

The terminalname variable on this line is the same terminal name as the one on the first line of your login sequence. If your terminal is automatically logged into a network application, you do not receive this prompt.

When you receive the APPLICATION prompt, enter the letters and digits that identify the network application you want to access; for example, IAF for the Interactive Facility. You can access CDC-written applications from the following list that are installed at your site:

- Remote Batch Facility (enter RBF).
- Interactive Facility (enter IAF).
- Transaction Facility (enter TAF).
- Terminal Verification Facility (enter TVF).
- Message Control System (enter MCS).
- PLATO (enter PNI, which stands for the PLATO-NAM Interface).

Other applications, written at your site, might be available.

You must respond to the prompt within 2 minutes. If you take too long to respond, this message is displayed at your terminal.

TIMEOUT.

Your terminal is then disconnected from the host.

You must spell your entry correctly. If you make a mistake, such as typing the letter O for the numeral 0 or the letter l for the number 1, you receive the following error message:

```
ILLEGAL APPLICATION, TRY AGAIN.  
terminalname - APPLICATION:
```

The terminal name is indicated by terminalname. You can also receive this message if your user name is not permitted access to the application you named. If you did not spell the name correctly, try again. If you did spell the name correctly, disconnect your terminal and contact a site administrator.

If you make four unsuccessful attempts to enter an application name, this message appears at your terminal.

APPLICATION RETRY LIMIT.

Your terminal is then disconnected from the host.

This step is complete when the application you request connects to your terminal. The application you asked to use might print an identification line next.

Whether the program sends such a message or not, you can use commands recognized by that application. If you need to disconnect your terminal from the application, the application probably has a command for that purpose. If the program has no such command, the subsection called Disconnecting from a Host describes a command you can use.

ABBREVIATED LOGIN AND APPLICATION CONNECTION

You can shorten the standard login and application connection procedures by entering all of the required information at once. For example, in response to the FAMILY prompt, you can enter the following information on the same line.

```
FAMILY: familyname,username,password,application
```

Separate all entries with a comma. The entries are order-dependent. Use a comma to indicate any default or unused entries.

You do not receive any of the remaining login selection prompts if you use this abbreviated login procedure, unless you omit a required entry or supply an entry that is not recognized. Your password entry is not protected by the system when you use an abbreviated login procedure.

Here are some examples. To log in and connect to the Interactive Facility application IAF with a family name of SYSTEMA, a user name of RLS4525, and a password of PASS123, enter:

```
SYSTEMA,RLS4525,PASS123,IAF
```

To use the site-defined login default family name, enter:

```
,RLS4525,PASS123,IAF
```

To override your site-defined login default family name with the host's default family name, enter:

```
O,RLS4525,PASS123,IAF
```

To use a login default family name and user name defined by the site for your terminal, enter:

```
,,,IAF
```

To also use a default application name defined by the site for your terminal, enter:

```
,,,
```

You can supply combined entries in response to the FAMILY, USER NAME, and PASSWORD prompts. Although you do not need to complete the remaining sequence on the same line, you must enter the information in the correct order: family name, user name, password, and application (if there is no default). For example:

```
FAMILY: systema  
USER NAME: rls4525,pass123  
terminalname - APPLICATION: iaf
```

If you supply a user/password combination that is overridden by a mandatory specification, this message is displayed.

VOLUNTEERED USER/PASSWORD IGNORED.

If you offer an application name that is overridden by a preassigned mandatory application, the following message is displayed.

VOLUNTEERED APPLICATION IGNORED.

SWITCHING TO A DIFFERENT APPLICATION

Unless you have a mandatory network application assigned for your terminal, you can transfer your terminal's connection from the current application to another application. There are two ways to do this, depending on the commands used by the current application.

Direct Switches

You can switch in one step by entering a switch command recognized by the current application. The following switch commands are recognized by CDC-written network applications:

<u>Application</u>	<u>Command</u>
RBF	IAF
IAF	BYE,application
MCS	LOGIN,application

For example, if your terminal is connected to RBF, you can gain access to IAF by entering the IAF command.

Do not enter any more input until the application acknowledges it has processed your switch command. Otherwise, you may lose the input.

Indirect Switches

If your current application does not have a command for a direct switch, you must disconnect your terminal from the current application with a command recognized by that application. Then you must request another application connection from the host. The following disconnection commands are recognized by CDC-written network applications:

<u>Program</u>	<u>Command</u>
RBF	END
IAF	BYE,
MCS	END
TAF	EX.LOGT
TVF	END

For example, you can switch your terminal connection from RBF to an application other than IAF by entering the END command. You then receive a prompt for a new application name. Do not enter any more input until the application acknowledges it has processed your disconnection command. Otherwise, you might lose the input.

Dialog for Direct and Indirect Switches

After disconnecting the application, the system indicates that a connection switch is in progress by issuing this message.

```
application - CONNECT TIME hh.mm.ss.
```

The application that was just disconnected is indicated by application and the time that passed while the terminal was connected to the program is indicated by hh.mm.ss.

If you used a switch command, you receive an identifying message from your new application. If you used a disconnection command, you receive another prompt for an application name.

```
terminalname - APPLICATION:
```

unless the terminal is permitted access to only one application. If so, you are disconnected from the host.

If you enter the name of another application in response to the APPLICATION prompt, you are connected to that application.

RESTARTING LOGIN IDENTIFICATION

There are many reasons why you might want to restart login identification and the subsequent application connection. Perhaps you have several user names available to you, each with permission to use different host resources. Or perhaps someone else left the terminal connected to an application you do not need, and you do not want to use their user name.

There are three ways to restart login, depending on the commands recognized by the current application and installation options.

Direct Restarts

You can restart host identification by entering a login command recognized by the current application. The following login commands are recognized by CDC-written network applications:

<u>Application</u>	<u>Command</u>
RBF	LOGIN
IAF	HELLO
MCS	LOGIN

For example, if your terminal is connected to IAF, you can restart login by entering the HELLO command.

Do not enter any more input until the application acknowledges it has processed your command. Otherwise, you may lose the input.

Indirect Restarts

If your current application does not have a command for a direct switch, you must disconnect your terminal from the current program with a command recognized by that application. Then you must request restart of login from the host. The disconnection commands recognized by CDC-written network applications are described earlier in this section under Switching to a Different Application.

For example, you can restart your terminal login from RBF by entering the END command. You then receive a prompt for a new application name. Do not enter any more input until the application acknowledges it has processed your disconnection command. Otherwise, you might lose the input.

Secure Login Restarts

If your site has selected the secure login initiation option and defined a security character, the sequence may be entered at any time to immediately disconnect from the current application and initiate the user login procedure.

After entering the disconnection command, you receive another prompt for an application name:

```
terminalname - APPLICATION:
```

unless the terminal is permitted access to only one application program. If so, you are disconnected from the host and can restart login by reconnecting.

If you receive the APPLICATION prompt, you use one of the following commands to ask the host system to restart login:

```
HELLO
```

```
LOGIN
```

DISCONNECTING FROM A HOST

You might want to disconnect your terminal from a host for several reasons, for example:

- You want to disconnect from the network properly, without leaving behind an indication that something went wrong; disconnecting from a host is the first step in disconnecting from a CDC network.
- You are allowed access to more than one host system, and you want to transfer your terminal's connection from the current host to another host.

There are four ways to do this, depending on the commands used by the current application.

Disconnecting with Application Commands

You can disconnect in one step by entering a host disconnection command recognized by the current application. The following host disconnection commands are recognized by CDC-written network applications:

<u>Program</u>	<u>Command</u>
RBF	LOGOUT or LOGOFF
IAF	BYE or LOGOUT or GOODBYE
MCS	BYE or LOGOUT

For example, if your terminal is connected to IAF, you can disconnect from the host by entering the BYE command.

Disconnecting with Host Commands

If your current application does not have a command for direct disconnection, you must first disconnect your terminal from the current program with a command recognized by that application. Then you must request host disconnection from the host. Application disconnection commands are described earlier in this section under Switching to a Different Application.

After disconnecting the application, the system indicates that a connection switch is in progress by issuing this message:

```
application - CONNECT TIME hh.mm.ss.
```

The application that was just disconnected is indicated by application and the time that passed while the terminal was connected to the program is indicated by hh.mm.ss.

You then receive another prompt for an application program name:

```
terminalname - APPLICATION:
```

unless the terminal is permitted access to only one application. If so, you are disconnected from the host anyway.

The following host disconnection commands are valid responses to the APPLICATION prompt on CDC host systems:

BYE

LOGOUT

For example, you can disconnect your terminal from RBF by entering the END command. You then receive a prompt for a new application name and enter BYE.

Disconnecting with a Network Software Command

You normally end a connection with a host by using a command to disconnect you from an application and/or a command to log out of the host. Events might make it impossible for you to do either. When this happens, you can end your terminal's connection to the host by entering this terminal definition command.

```
ct TM
```

ct represents the network control character. For most asynchronous terminals this is ESC.

The network responds to your TM command with a new host availability display. You can use the TM command if an application does not respond to your entries.

Reconnecting to a Host or Connecting to a Different Host

For a single host system, after your terminal is disconnected from the host, this message is printed.

```
HOST DISCONNECTED. CONTROL CHARACTER = ct
```

ct is the network control character. This message is followed by the prompting message.

```
ENTER INPUT TO CONNECT TO HOST
```

You can then attempt another connection to the same host by sending any input you want. If you have a multihost system, typically the HAD will be displayed between these two messages. To reach a different host, you must complete the procedures for Selecting a Host and Connecting to the Selected Host described earlier in this appendix.

Disconnecting from One Host and Connecting to a Different Host

You can disconnect from one host establish a connection with another host using the single command

```
ct HC = hn
```

ct is the network control character, and hn is the name of the desired host.

INTERRUPTIONS

Events can sometimes interrupt your input or output. These interruptions fall into two categories.

- Interruptions for communication with the network operator.
- Temporary suspensions in communication.

Communicating with the Network Operator

The term network operator includes any operator who has access to the network, including the host operator. Any network operator can send and receive messages that involve you.

Receiving Messages

The network operator can send short messages to a terminal. The message appears at your terminal in this format.

```
FROM NOP . . . message text
```

These messages can appear at any time a new line of output is possible at your terminal. For example, the network operator might want to inform all connected terminals that communication is going to be suspended intentionally with this message.

```
FROM NOP . . . SHUTDOWN IN 5 MIN
```

Sending Messages

You can respond to such messages by entering the following terminal definition command.

```
ct MS=message
```

ct represents the network control character. For most asynchronous terminals this is ESC.

The message text must not exceed 50 graphic characters. Because the network operator can be located at a terminal that does not support many special characters, you should avoid characters not in the ASCII 63-character subset described in appendix A of the NOS Reference Set, Volume 2, Guide to System Usage.

You can also begin dialog with the network operator. For example, if you need to use a remote batch printer that has been logically disabled, the entry:

```
%MS=PLEASE ENABLE LP=M4C555
```

causes the message PLEASE ENABLE LP=M4C555 to appear at the network operator's terminal, prefixed by the terminal name of your terminal. The % character represents the network control character for your terminal.

Preventing Messages

Sometimes it is important that output to your terminal does not include unplanned messages from the network operator. You can prevent delivery of messages from the network operator by locking them out with the following terminal definition command.

```
ct LK=option
```

ct represents the network control character. For most asynchronous terminals this is ESC.

After you enter this command with the option Y, any message to your terminal from the network operator is discarded by the network software. Messages are discarded until you again allow delivery by using the command with the option N.

Site administrators can configure your terminal so that messages are discarded as soon as the terminal is connected to the network. A network application can also change the LK option in use.

Suspensions of Communication

Sometimes, message traffic in the network is so heavy that all storage is temporarily used up. Also, a host may be too busy to accept data. During these times, the network software takes steps that temporarily stop input until enough output has occurred to free additional storage.

Although storage may be low, sometimes the network cannot prevent you from entering data. Therefore, the network discards each message from your terminal and sends this message to your terminal.

```
WAIT . .
```

When the following message appears, you can reenter any information that was previously discarded.

```
REPEAT . .
```

If your terminal transmits more than one message at a time, you must determine which information you need to reenter. The application might have commands that can help you determine which messages it received.

Communication Failures

Events can occur that:

- Cause your application to fail.
- Disconnect your terminal from an application.
- Disconnect your terminal from the host.
- Disconnect your terminal from the network.

Application Failures

If an application fails after your terminal is connected, this message is displayed.

```
APPLICATION FAILED.  
application CONNECT TIME hh.mm.ss.  
terminalname - APPLICATION:
```

The length of time your terminal was connected to the application is specified by hh.mm.ss; application is the name of the application; and terminalname represents the terminal name the network uses to identify the terminal. You either can select another application or log out.

Disconnection from an Application and/or a Host

An application is informed of two events that you might want to take action on.

- Long periods of device inactivity.
- Pending network shutdown.

The network software informs an application if more than 10 minutes have elapsed without any communication between a device at your terminal and the application. An application can:

- Ignore this time interval.
- Ask you for continued dialog.
- Disconnect from your terminal or from the device with or without telling you.

The effect of device inactivity depends on the application with which the device is currently connected.

When shutdown of the network is pending, the network operator can enter a command that informs all applications of the pending event. Your application might take actions that either result in connection termination or application failure. It also might send you a message to warn you of the event.

If you receive a message from an application indicating that shutdown is pending, you should take steps immediately to save all data or files you are currently using. Then end connection with both the application and the host as soon as possible.

Disconnection from the Network

The network informs you that it has lost communication with the host software by displaying the host status message:

```
HOST DISCONNECTED. CONTROL CHARACTER = ct
```

followed by the prompting message:

```
ENTER INPUT TO CONNECT TO HOST
```

If you have a multihost system, the HAD will be displayed between these two messages.

If you are at a dial-up terminal and you do not attempt to connect to the same or a different host within 2 minutes, your terminal is disconnected from the network.

If communication between the network and the host software resumes, you must connect to the host and log in again but you do not need to access the network again (perform dial-up and auto-recognition procedures). These host communication failures occur either when the host has been shut down or when hardware fails in the communication path between the network and the host.

All network software conditions (that is, parameters that may be set with terminal definition commands) are retained across host disconnection except for transparent, full ASCII, and special edit input modes. Refer to the NOS Reference Set, Volume 3, System Commands for a discussion of these commands.

Failures in the communication path between the network hardware and your terminal are treated by the network software as if your terminal had failed. These failures can occur because of hardware problems, or maybe because you entered an input character during output. If you hold the interactive output interrupt key down for too long, a communication failure can occur. This key is identified in the NOS Reference Set, Volume 3, System Commands.

There are several tests for communication failures. If your terminal has a CTS, ON-LINE, or CARRIER indicator, the light goes out when a failure occurs. At other terminals, if a failure has occurred, requesting the host availability display produces no response from the network.

Batch input and output devices associated with the failed console stop running, and batch device connections are ended. The network software does not distinguish between dial-up and hardwired terminals when processing terminal failures.

Terminal failures require you to complete new access and login procedures to resume communication with the network. Terminal failures always cause disconnection when the terminal is on a dial-up line.

INDEX

- Accessing a host J-1
- Accessing transaction subsystem 1-4
- ACT 2-11
- After image recovery file 8-1
- AKY statement 4-1
- Alternate keys 4-2
- Application block header
 - Building 2-12
 - Examining 2-14
 - Modifying 2-12
- Application character type 2-11
- ARF 8-1
- Attach mode 4-2
- Audience 5
- Auto-input mode 1-3; 2-12
- Automatic recovery
 - Description 7-1; 8-1
 - Report 8-4,6
 - Requests
 - Application 2-32
 - System 2-38

- Backup directory file 8-2
- Batch
 - Interface 1-3; 5-1
 - Subsystem interaction 5-1
- Batch output 5-4
- Batch recovery 8-1
- Before image recovery file 8-1
- Begin-commit identifier 7-1,3
- Begin-commit sequence 7-1; 8-1
- BEGIN request
 - Data base 7-1,3; 8-1
 - TAF 2-28
- BLDABH request 2-12
- Blocked tasks I-1
- BRF 8-1
- BTASK system task 8-4
- BTRAN request 5-1; 7-1
- Build ABH 2-12
- BWAITINP request 2-28

- Call tasks 2-4
- CALLRTN 2-4
- CALLTRN request 2-38; 8-1
- CALLTSK 7-4,5
- CALLTSK request 2-6
- CDCS 1-2; 3-1; 4-4; 7-1,3
- CEASE request 2-28
- Chain 2-4
- Change ACT 2-11
- Change CMDUMP request values 6-1
- Change I/O limit 2-28
- Change task field length 2-3; 10-6
- Character sets A-1
- Character type 2-11
- CLOSE requests 7-2
- CMDUMP request 6-3
- COMMIT requests 7-1,3; 8-1

- Commit updates 7-3
- Communication block
 - Defined 2-1
 - Diagram 2-2
 - Dump 6-4
 - Load overflow data 2-10
 - Release overflow data 2-23
 - TAF-originated transactions 12-3,4
- Communication recovery file
 - Description 4-1,8; 7-1; 8-1
 - Error override 8-3
 - Multiple 8-4
 - Reformat 8-3
 - User messages
 - Read 2-33
 - Write 2-33
- Concurrent users 4-2
- Confirmation message 2-35; 7-1
- Create task library 10-7
- Create xxJ file 4-4
- CRF 4-1,8
- CRM file type 4-2
- CYBER Database Control System 1-1,2; 3-1; 4-4
- CYBER Record Manager Data Manager 1-1,2; 4-1,2

- Data base
 - Describe 9-1
 - Journal files 4-5
 - Maximum number of 4-7
 - Recovery 7-1; 8-1
- Data manager
 - Concurrency 1-1; 3-1
 - Requests 2-3
 - xxJ files 4-1,4
- Deadlocks 3-1
- Define terminal characteristics 2-16
- Delete tasks from task library 10-7
- Describe users and data bases 9-1
- Determine if user is logged in 2-31
- Determine subsystem for task 2-28
- Diagnostics B-1
- DIAL directive 9-1
- Disable transaction recovery 8-2
- DISPLAY statement 4-7
- DMS statement 4-7
- DSDUMP request 6-1
- Dump
 - Communication block 6-4
 - Exchange package 6-4
 - TAF field length 6-4
 - Task field length 6-3,4

- ECS 5
- End task 2-28
- Entering transaction subsystem 1-4
- Entry header, journal file 4-5
- ERO command 8-3
- Error messages B-1
- Establish system identifier 2-40

Examine ABH 2-14
Execute named task 11-2
Executing transaction 7-1
Extend field length 2-3; 10-6
Extended core storage 1-5
Extended memory 5; 1-5
Extract bit string from word 2-31
EXTRACT request 2-31

File

CRM type 4-2
Definitions E-1
Descriptions E-1
Index 4-2

Files

Journal 4-5
JOURO 4-5
NCTF1 4-8
Network 4-8; 9-1
Procedure 4-1
Recoverable 8-1
Recovery 4-8; 8-1
Reserved E-2
System 4-1
TCF 4-1,7; E-1
xxJ 4-1,2,4
xxJORN 4-4
xxTASKL 10-1

Forced write indicator 4-3
Format effectors 2-12,15
Free requests 7-1
FSE C-6

GETABH request 2-14
Global task dump limit 6-5
GTDL 6-5

Hardware configuration 8-2

Ignore task in replacement file 10-8
IIO request 2-28
Implement application 1-5
Index file 4-2
Initiate a run unit 2-38
Initial task 1-3; 11-1; 12-1
Initialize recovery files 8-2
Initiate new transaction chain 2-6
Input/output
Limit 2-28
Requests 2-8
Input overflow 2-10,23
Insert bit string into word 2-31
INSERT request 2-32
Installing TAF F-1
INT command 8-2
Interactive virtual terminal 1-3; 2-16
Interrogate TST 2-23
ITASK 1-4; 11-1; 12-1
ITL request 2-29
IVT 1-3; 2-16
IXN statement 4-2

Journal files 4-5
JOURNL request 4-5
JOURO file 4-5

K display commands 2-30; 4-8; 6-1,4; 8-2,3
K.DSDUMP command 6-4
K.DUMP command 6-5
K.DUMPLIM command 6-5
K.ERO command 8-3
K.INT command 8-2
K.REC command 8-3
K. statement 4-8
KDIS 11-1
KPOINT request 2-30
KTSDMP utility/request 6-4

Libraries

LIBTASK command 10-2
LIBTASK input directives 10-5
LIBTASK utility/command 10-1
Line feed and transmission keys H-1
List tasks on library 10-3
Load overflow blocks 2-10
LOADCB request 2-10
Lock conflict 7-2
LOGIN request 2-31
Login terminal to TAF 1-4
Logout task 11-1
Logout terminal from TAF 2-29; 11-1
LOGT request 2-29
LOGT system task 11-1

Macros

12-1
Manipulate user area of TST 2-26

Memory

Changing from tasks 2-3; 10-5
Dump requests 6-1
Dumps 6-1
Requests 2-3
Message abort 11-1
Modify ABH 2-12
Modify terminal characteristics 2-16
Move data from terminal to task 2-28
MSABT 11-1
Multimainframe 1-5
Multimainframe recovery 8-4
Multiple indexed files 4-2
Multiple RECOVER statements 8-3

NAM

1-3; 12-3
Named-transaction 7-1; 10-5
NCTF1 file 4-8
NDL 1-5
Network

Access Method 1-3; 12-3
Definition language 1-5
File origination 9-1
Files 4-8; 9-1
Interface 1-3
NETWORK statement 4-8; 8-3
NEWTRAN request 2-7

OFFTASK

11-2
On-line transaction processing 1-1
OPEN requests 7-2
Output from SUBMT jobs 5-4
Override I/O errors 8-3

Parameter
SEND request 2-18
Parameter table 2-25
Procedure file 4-1,4
Put recovery data on CRF 2-33,41

RA+1 calls 1-5; 2-1
RC parameter 7-1
RCTASK system task 11-2
Read data from terminal 2-8
Read recovery data 2-33
Read SECURE message 2-34
Record data on journal file 4-5
Record lock conflict 7-2
Record locks 4-2
RECOVER statement 4-8; 8-2,3
Recoverable files 8-1
Recovery
CDCS 7-1
CDCS failure 8-5
CRF file 8-3
Data base 7-1; 8-1
Disable 8-2
File identifier 4-8
Flag 8-4
General 7-1; 8-1
Hardware configuration 8-2
Initialize files 8-2
K display commands 4-8; 8-2
Multimainframe 8-4
Multiple data manager tasks 7-5
NAM failure 8-5
Override I/O errors 8-3
Overview 7-1; 8-1
Processing flow 8-4
Requests
Application 2-33
System 2-38
System failure 8-4
TAF/CRM Data Manager 7-1
TAF failure 8-4
Task failure 8-5
TCF file 8-3
Terminal 7-4; 8-5
Transaction 7-1; 8-1
Recovery indicator 10-5
Application 2-32
System 2-38
Release overflow blocks 2-23
RELSCB request 2-23
Requests
Data manager 2-3
Input/output 2-8
Journal file 2-3; 4-5
Memory 2-3
Memory dump 6-1
Message 2-4
Recovery
Application 2-32
System 2-38
System 2-3
Task
Control 2-28
Scheduling 2-4
Utility 2-30
Rerun request 2-32
Reserved files E-2
RESTRICT clause processing 3-1
Resubmit terminal input 2-32,40
Retrieve recovery data 2-33
RGET request 2-33
RPUT request 2-33

RSECURE request 2-34
RTASK system task 8-4; 11-2
Complete 2-35
Initate 2-38
Running transaction 7-1

Sample deck structures D-1
Schedule tasks 2-4; 10-7
Scheduling priority 10-7
SECURE message
Acknowledge 2-36
BTRAN 2-36
Description 7-3; 8-1
Retrieve 2-34
Write 2-35
SECURE request 2-35; 7-1,3; 8-1
Send message task 11-2
Send message to user 2-17
SEND request 2-17
Set GTDL 6-5
Set system identifier 2-40
Set time limit for task 2-29
Set transaction sequence number 2-40
Set user recovery bit 8-3
SETCHT request 2-11
Specify communication block location 2-28
SRERUN request 2-40
Start transaction using batch job 5-1
Start update sequence 7-3
STASK 11-2
Status of current transaction 2-37
Status of current user 2-37
Status of transaction 2-8
STRAN macro 12-1
Subcontrol points 1-5
Submit batch job from task 5-3
Submitting comments 5
SUBMT request 5-3
Subtransaction codes 12-1
Supervisory messages 2-11
Suspend processing 2-30
SYSMSG 11-2
System
Files 4-1
Journal files 4-5
Requests 2-3
Task library 10-1
Tasks 11-1

TABCON statement 4-7
Tables
Terminal status (TST) 9-2
TAF
Automatic recovery report 8-4,6
Configuration file 4-7; E-1
Control point 1-1,2
Installation F-1
Originated transactions 12-3
Procedure file 4-1,4
Termination 8-5
TAF/CRM Data Manager 7-1
TAFREC 8-4
TARO request 2-26
Task
Chain 2-4
Control requests 2-28
Defined 1-4; 2-1
Dump 6-1
Field length changes 2-3; 10-6
Library

- Create 10-7
- Defined 10-1
- Delete task from 10-7
- Ignore task in replacement file 10-8
- List tasks 10-3
- Update 10-7
- Pause 2-30
- Queue limit 10-6
- Queuing 10-1,6,7
- Residency 10-1
- Scheduling priority 10-7
- Scheduling requests 2-4
- Type 10-1
- Utility requests 2-30
- Task library report 10-3,4
- TASKLIB 10-1
- Tasks
 - Blocked I-1
 - Call 2-4
 - End 2-29
 - Schedule 2-4; 10-7
 - Time-originating 12-1
- TCF file 4-1,7; E-1
- TERMDEF request 2-16
- Terminal
 - Characteristics
 - Define 2-16; G-1
 - General 12-4,5
 - Modify 2-16; G-1
 - Definition commands G-1
 - Failure 7-4; 8-5
 - Network description file 9-1
 - Recovery 7-4
 - Status request 12-5
 - Status table
 - Figure of 9-2
 - Interrogate 2-23
 - Manipulate user area 2-26
 - Recovery bit 8-3
- Terminology and conventions 5
- TIMCNT macro 12-1
- Time-originating tasks 12-1
- TINVOKE request 2-40
- TMSG request 2-4
- TOTAL Data Manager 1-1,3; 4-4; 7-5
- TOTAL task area I-1
- TPSTAT request 2-28
- TRAN macro 12-1
- TRANCHK request 2-8
- TRANS 12-1

- Transaction
 - Codes 12-1
 - Coding 7-2
 - Defined 1-1
 - Named 7-1
 - Processing 1-4
 - Recovery 7-1; 8-1
 - Rerun 2-32
 - Subsystem
 - Entering 1-4
 - Interaction 5-1
 - TAF-originated 12-3
 - Transaction-id 7-3
 - Transaction sequence number 2-2,40
 - Transient task area I-1
 - Transparent mode 2-12
 - TRANT 12-1
 - TSIM request 2-23
 - TST 9-2
 - TSTAT request 2-37
 - TTOT 12-1
 - Type ahead 12-5

- Updating task library 10-7
- User recovery bit 8-3
- USER statement 4-8
- User status 12-5

- Wait for terminal input 2-9
- WAIT request 2-30
- WAITINP request 2-9
- Write data to CRF 2-33,41
- WSTAT request 2-41

- XTASK 11-2
- XTRAN system transaction 11-2
- xxJ files
 - CDCS Data Manager 4-4
 - Creating 4-4
 - CRM Data Manager 4-2
 - No data manager 4-4
 - TOTAL 4-4
- xxJORN file 4-4
- xxTASKL file 10-1

COMMENT SHEET

MANUAL TITLE: CDC TAF Version 1 Reference Manual

PUBLICATION NO.: 60459500

REVISION: C

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ STATE: _____ ZIP CODE: _____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

Please Reply

No Reply Necessary

CUT ALONG LINE

AA3419 REV. 4/79 PRINTED IN U.S.A.

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD

FOLD



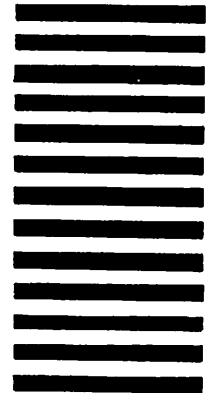
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112



CUT ALONG LINE

FOLD

FOLD