INTERNALS

Document Number TND-0205-10

The Network Director

North Ridge Software, Inc.

Special Notices

This document contains proprietary information associated with a generalized software product named **The Network Director**, which is a VTAM based terminal security and productivity product developed, maintained, and marketed by North Ridge Software, Inc.

Information contained herein that is associated with other proprietary products (as identified below) is also subject to copyright law and may not be reproduced without the express written permission of the appropriate company.

North Ridge Software, Inc.
12515 Willows Road N.E.
Suite 205
Kirkland, Washington 98034-8795
U.S.A.

North Ridge Software, Inc. can be contacted via any of the following mechanisms:

Telephone 425/814-9000
FAX 425/823-9636
InterNet support@nrsinc.com
Homepage http://www.nrsinc.com

North Ridge Software, Inc. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of fitness for any particular purpose.

Acknowledgements

References within this manual to the following products should be recognized as references to proprietary products and trademarks of the following firms:

Computer Associates	TOP SECRET, ACF2, UCC7, ROSCOE, EMAIL, IDMS/DC, TPX
IBM	ACF/VTAM, ACF/TCAM, NMPF, NPDA, VM/GCS, OS/VS1, NETVIEW, NLDM, NPA, CMS, MVS, DOS/VSE, CICS/VS, TSO, IMS, RACF, NPDA, and NCCF
Software AG	Com-plete
Sterling Software	VM/SECURE

Table of Contents

List of Illustrations

Introduction

The Network Director's *Internals* manual is intended for the individual who desires more information about how The Network Director operates internally.

Although it is not required, this manual assumes that The Network Director's Source Code is available to the reader. This manual also assumes that the reader is familiar with 370 ASSEMBLER code, Operating System fundamentals, and the VTAM Application Program Interface. The reader is referred to the appropriate IBM manuals for information in any areas that may require additional background.

The information present in this manual is not required knowledge to administer, manage, or utilize The Network Director. It is presented here in an effort to clarify the internal techniques used by The Network Director and to provide a knowledgeable individual a working knowledge of The Network Director's internal architecture.

Section Overviews

"Overview" on page 5 provides a quick description of the Source Library's layout and a more detailed description of the general topics covered in subsequent Sections.

"Functional Modules" on page 9 discusses the various executable modules (OS members, CMS files, or DOS books) that make up The Network Director. Each Module is briefly described and its general functional area is identified.

"Control Blocks" on page 113 identifies the Control Blocks used by The Network Director. Each major Control Block is described, its origin is identified, and its connections or links to other Control Blocks are discussed.

"Macros" on page 135 lists The Network Director ASSEMBLER Macros, their use, and identifies their individual purpose.

"Internal Facilities" on page 161 describes the internal structural and logical conventions that have been used during implementation of The Network Director. Internal storage management, basic dispatching and the External File are also described.

"Dump Analysis" on page 173 is a brief description of how to approach diagnosing problems within The Network Director that have created ABEND situations. Locating key control blocks and interpreting the function in control at the time of ABEND are emphasized.

"VTAM Characteristics" on page 177 identifies the manner with which The Network Director utilizes the VTAM Application Program Interface. This becomes more important as the VTAM Access Method itself undergoes changes as Access Method and Operating System evolution continue.

"Storage Estimates" on page 185 provides detailed information on how to compute the storage requirements for The Network Director. The individual control blocks and their relationships are discussed as well as providing formulas that can be utilized to compute specific operational characteristics.

"Installation Exits" on page 193 describes all the standard exit points that are available within The Network Director. This section can be used by an installation to tailor or extend facilities of The Network Director.

Finally, a Glossary of Terms and a manual Index are included to aid the reader in the use of this manual.

The Manual Set

This manual is one of a set related to The Network Director. The set consists of:

Number	Manual Title
TND-0201	General Information Manual
TND-0202	Network User's Guide
TND-0203	Network Administrator's Guide
TND-0204	Quick Reference Guide
TND-0205	Internals
TND-0206	Messages and Codes
TND-0210	Network Operator's Guide
TND-0219	Installation Guide
TND-0220	Single System Image
TND-0226	SecureNet Key Interface Reference
TND-0420	Version 4.2 Release Guide

Figure 1. The Manual Set

Each Network Director installation is provided with a complete set of base documentation for The Network Director. The base set consists of the *General Information Manual, Network User's Guide, Network Administrator's Guide, Quick Reference Guide, Internals, Messages and Codes, the Network Operator's Guide, Single System Image,* and the *Installation Guide*. Additional documentation is available, as requested.

4 Internals

Overview

The following sections each describe a specific portion of The Network Director's internal architecture. This section provides a quick overview of the technical aspects of The Network Director and its philosophy.

This manual attempts to identify the internals of The Network Director in sufficient detail to properly discuss how its architecture functions. However, the reader should note that the final authority on how The Network Director functions is the source code. It is typical for software subsystems to continually evolve to match requirements within the industry. While every effort to keep this manual current has been made, the reader should recognize that the documentation, publication, and distribution process may from time to time introduce a difference between what is documented here and what is actually within the software.

The Source Library

The Network Director's Source Code is distributed to Licensed sites as a single OS PDS, multiple CMS files, or a DOS source library. Regardless of operating environment, the source is a series of ASSEMBLER source Modules, Macros, and possible JCL.

A Network Director Module is an 370 ASSEMBLER CSECT containing executable instructions. It will be functional in differing environments, dependent upon its assigned task.

A Network Director Macro will contain either an ASSEMBLER DSECT generation or a traditional canned set of instructions for general use within Modules.

All Source elements on the source library are uniquely named. Each one begins with either the three characters "TND" or "NRS". The remaining five bytes allowed in the name are assigned to the specific module or macro to describe its function.

Each Source element begins with a prologue or a brief English description of the element's purpose, its input and output, and its process. Depending upon the element, it may or may not contain additional prose describing specific sections of the functions it accomplishes.

Technical Aspects

The Network Director is basically a fully reentrant software subsystem within the operating system. Certain functions and facilities within some environments (DOS, CICS, IMS) are not reentrant. This is true because of the environment itself and is not a characteristic introduced by North Ridge Software, Inc. policy. The Network Director's storage requirements will vary depending upon the environment and the size of the logical network being managed.

The individual CSECTs that make up the executable DIRECTOR are combined prior to execution by the standard Operating Environment Linkage Editor or Loader. Generally, The Network Director executes in problem program state and require no specialized authorization.

The Network Director makes use of the documented VTAM Application Program Interface and the Program Operator interface to accomplish all communications with VTAM.

The Network Director typically executes within the operating system within its own partition or address space, which may be Initiated or Started depending upon installation requirements.

Whenever a Network Director facility is required to provide code to modify other software subsystem's functions, the provided code attempts to take advantage of standard exits made available within the other software subsystem. North Ridge Software, Inc. is committed to utilizing documented interfaces. When a subsystem without sufficient documented interfaces is encountered (CICS SSI as an example) and a documented interface is later made available, North Ridge Software, Inc. will convert any existing Network Director code to the new documented interface when it is made available by the vendor.

Philosophy

The Network Director has multiple generalized philosophical approaches it has been designed around. They are listed in the following paragraphs in no particular order and are listed to provide the reader with a general concept of the philosophical approach used during design and implementation. This is done in the hope that the reader will be better equipped to manipulate and read the Source Code.

Operating System

All operating system services have been isolated into a limited number of functional CSECTs. This is also true of actual VTAM and VSAM functions. This includes the Operating System SVCs and Access Method dependent Macros. The rest of The Network Director interacts with these isolated CSECTs via internal TND Macros, which will provide the same parameter list to the Operating System CSECT regardless of environment (OS or DOS, etc).

This structure allows The Network Director to minimize operating system or VTAM release level exposures. When it does make use of external facilities, The Network Director localizes direct interface to simplify any changes that might be required by software subsystem upgrades (MVS/ESA, etc).

Modularization

The Network Director has also localized most of the internal facilities into specific modules. Each module is further broken up into smaller segments of code that are structured generally into a tree structure. This mechanism is further described in "Functional Modules" on page 9.

Logical Network

The Network Director builds the network image dynamically. This allows the Network Administrator to add, delete, and change definitions while the network is active. Philosophically, The Network Director is prepared to run 24 hours a day. Every attempt has been made to eliminate a requirement to restart The Network Director.

Storage

The Network Director does not require any specialized tuning in order to execute normally. Philosophically, The Network Director attempts to maintain general characteristics that allow each installation to utilize standard operating system facilities to manage The Network Director.

Storage management within The Network Director has been organized to be compatible with virtual paging systems. The executable nucleus itself is fully reentrant eliminating PAGEOUT operations (for OS and VM environments). The Network Director does not constantly rely upon GETMAIN/GETVIS for storage (see "Internal Facilities" on page 161), but manages storage itself. This allows The Network Director to pack control blocks, buffers, etc together within its partition or address space. This allows for improved page reference patterns within the partition or address space dispatching The Network Director.

Network Users

A major goal for The Network Director is to provide a simple, easy to use interface to the system for the network user. Every attempt has been made to minimize keystrokes and establish default values for the network user.

Within The Network Director, a network user sitting at a terminal asking for service will always be prioritized over an internal function that is not response time critical.

8 Internals

Functional Modules

This section generally discusses what a Network Director **Module** is and then proceeds into descriptions of each Module within the system.

Definition

A Module is a Network Director executable CSECT that provides a single function within The Network Director's environment. The Module may reside in The Network Director's partition/address space/virtual machine or in one or more target subsystems. Each Module corresponds with a single member, file, or book on the source library (called Source Element from this point on).

The functional Modules within The Network Director are generally organized into a conventional tree structure. The following figure presents an overview of how the highest level Modules are related within The Network Director's partition or address space.

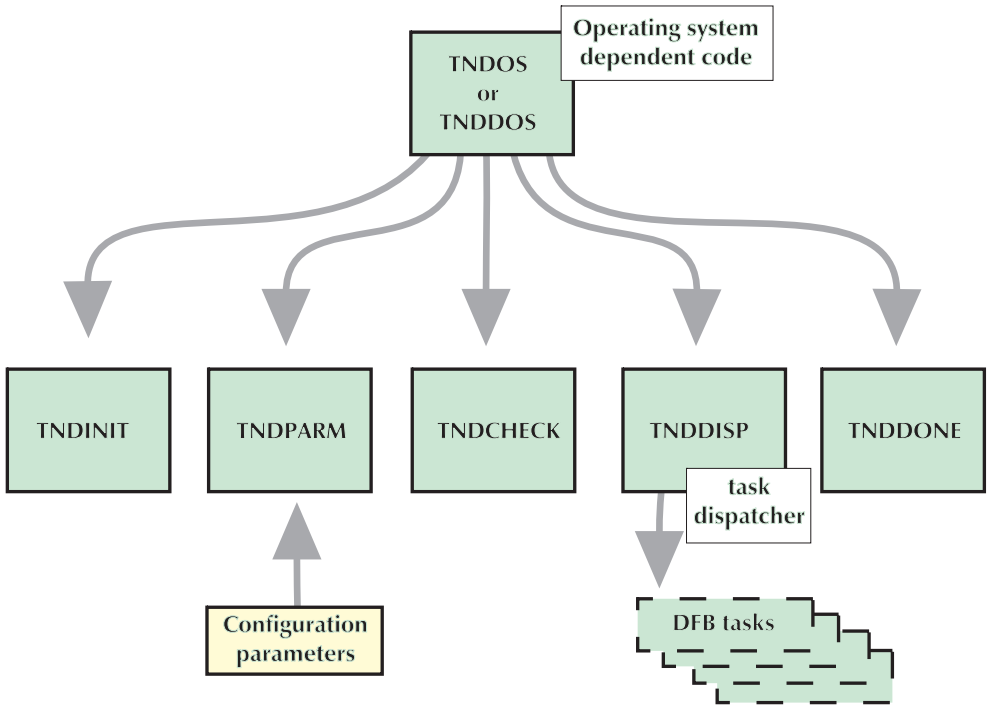

Figure 2. General Module Relationships

You should remain aware that this pictorial representation is intended to provide a general concept of which modules are at what level within the partition or address space. The actual technical interaction between the modules is somewhat different from that pictured.

The Modules not represented in the preceding figure have utility purposes defined by their individual purpose. They may be general routines that are callable at any point or may be extensions of the existing primary Modules.

Naming Conventions

Each source element is made up of a single CSECT and is internally structured into individual segments that are arranged in a traditional tree structure. All internal segments and ASSEMBLER labels are named *nnnnxyzz* where:

nnnn	is an acronym associated with the source element
x	is a numeric value associated with the major function of the Module that this segment or statement belongs to
y	is a numeric value associated with the minor function of the Module that this segment or statement belongs to
zz	is a numeric value assigning sequence to a label within a segment of the Module

Figure 3. ASSEMBLER Label Conventions

Modules will be identified by their acronym (DISP is the dispatcher). Major Segments will be in the form *nnnnx000* (DISP1000). Minor segments will be in the form *nnnxy00* (DISP1200). As an example, the tag DISP1235 is in the dispatcher's (TNDDISP) first major segment, second minor segment, and has the sequence number 35.

Major segments with a numeric segment number are always controlled by their owning segment, which is called the Primary Segment. The Primary Segment is identified with a major segment value of zero (0). Any segments without a numeric segment value are outside of the standard structure and may be called from anywhere within the Module, but not outside of the module.

Minor segments are always invoked from their owning major segment. No other entries are made to the minor segments or their specific labels.

Each Module is organized in an ascending order on the sequence numbers associated with segment and label tags.

Individual Module Descriptions

The following Module descriptions contain information about each CSECT shipped as a portion of The Network Director. A brief summary of what the module does is followed by a logical breakdown of the various routines within the program.

Relationships between the modules are implied by their Purpose. The general mechanism used to control the flow throughout The Network Director is described in "Internal Facilities" on page 161. This Section is dedicated to simply describing the individual Modules.

TNDACF2, ACF2 Interface

Provides the interface between The Network Director and the CA-ACF2 extended security package.

The Module's structure is as follows:

ACF21000 Invoke ACF2 to validate the password

- **ACF21100** Collect a reply for the ACF2 dialog
- **ACF21200** Update the System Directory
- **ACF21300** ACF2/VSE Initialization

ACF22000 Check for LOGON-MESSAGE=GENERAL situations

ACF23000 Logoff the user

ACF24000 Process the INHERIT function call

- **ACF24100** Delete the prior ACICB

ACF25000 Validate the SSE chain

- **ACF25100** Create the parameter list
- **ACF25200** Invoke ACF2 for Mass Interpret
- **ACF25300** Translate the results back to the SSE chain

ACF26000 Directory Build

ACF27000 Application FDE-NAME look up

- **ACF27100** Change offset to a Mini-LID

ACF28000 Check for presence of LID bit mask

ACF29000 Verify the Account Code

ACF29100 Invoke ACF2 via the SVC or ACFDIAG

ACF29200 Locate the ACCVT and ACUCB (ACF2/MVS only)

ACF29400 Issue a Message

TNDACQ, ACQUIRE=YES processor

Issues the SIMLOGON function for any devices that are to be acquired after the VTAM ACB has been opened (this is a result of the TERMINAL ACQUIRE=YES specification).

The Module's structure is as follows:

ACQ1000 Allocate the required RPL

ACQ2000 ACQUIRE=YES terminal processing

TNDACTN, Process the Action Bar for CUA Panels

Process all function requests from the CUA Action Bar

The Module's structure is as follows:

ACTN1000 Generate the Window constants

ACTN2000 Refresh the current screen

ACTN2100 Accumulate the screen buffers

ACTN7000 Allocate a new SWA and chain it to old one

ACTN8000 Free and unchain the newest SWA

TNDADMIN, Network Administration

Co-ordinates the various portions of the internal function known as Network Administration

The Module's structure is as follows:

ADM1000 Send the output queue display panel

- **ADM1100** Build fixed screen portions
- **ADM1200** Position within the Log buffer
- **ADM1300** Interpret a Log Buffer Entry
- **ADM1400** LOCATE subcommand search
- **ADM1500** Scan LBE for the Locate character string

ADM2000 Output the Log screen image

ADM3000 Process operator response

- **ADM3100** Validate actions
- **ADM3200** Invoke Program Operator
- **ADM3300** Invoke the VM Diagnose function
- **ADM3400** Invoke the DISPLAY processor
- **ADM3450** ACF2 command
- **ADM3460** SAVE command
- **ADM3470** RELOAD command
- **ADM3500** Operator services
- **ADM3550** SHOW command
- **ADM3600** Invoke interactive Deck manipulation
- **ADM3700** Check for Log buffer display commands
- **ADM3800** Invoke DUMP facilities
- **ADM3900** PREFIX command

ADM4000 Initialization logic

ADM9100 TNDCMPR for current prefix value

ADM9200 Point at next LBE in a forward direction

ADM9300 Point at next LBE in a backwards direction

TNDAPPC, LU 6.2 Interface Routine

Provides a generalized interface between The Director's internal functions and the VTAM defined 6.2 interface macros.

The Module's structure is as follows:

APPC1000 Initialize the RPL, etc.

APPC2000 OPRCNTL CNOS

APPC3000 DACTSESS

APPC4000 ACTSESS

APPC5000 RCVFMH5

APPC6000 RECEIVE

- **APPC6100** Send back confirmation we got some data

APPC7000 SEND

TNDBATCH, External File Maintenance

Manages all batch related External File activities.

The Module's structure is as follows:

BAT1000 Initialization

- **BAT1100** Open Log and Parms files
- **BAT1200** Create VSAM control blocks
- **BAT1300** Fake the Director"s PDA environment
- **BAT1400** OPEN the VSAM file

BAT2000 Acquire a Parameter

- **BAT2100** Validate the command
- **BAT2200** Validate the operand
- **BAT2300** Check for second operand =(first,second)

BAT3000 Display

- **BAT3100** Summary
- **BAT3200** Messages
- **BAT3300** HIX
- **BAT3400** Profiles
- **BAT3500** MIX
- **BAT3600** DMT
- **BAT3700** AIB
- **BAT3800** Info
- **BAT3850** LIST
- **BAT3900** Saved

BAT3950 Directory

BAT4000 Globals

BAT4100 VSAM-PASSWORD
BAT4200 OPSYS

BAT5000 Delete

BAT6000 Upgrade Info

BAT7000 Extract

BAT7100 Extract Info
BAT7200 NRS

BAT8000 Termination

BAT9000 VSAM related errors

BAT9100 Read a VSAM record

BAT9400 CLOSE the VSAM file

BAT9500 Print an output line

BAT9900 Print a heading line

TNDBAT1, External File Display Processor

Manages all batch related detail display activities

The Module's structure is as follows:

BAT1000 Display Messages=userid

BAT1100 Format a single message

BAT2000 Display Profile=userid

BAT2100 Format a profile record

BAT3000 Display Info=screen

BAT3100 Format a single info panel
BAT3200 Display the keywords for the panel

BAT4000 Display Mix=owner

BAT4100 Format a message index

BAT5000 Display Hix=owner

BAT5100 Format a Info index

BAT6000 Display Aib=name

BAT6100 Format a single Authentication block

BAT7000 Display Saved=setname

BAT7100 Format a single saved definition

BAT8000 Display Directory=name

BAT8100 Format a single directory entry
BAT8500 Display List=name
BAT8600 Format a single Distribution list entry

BAT9000 VSAM related errors

BAT9100 Read a VSAM record

BAT9300 Open the VSAM file

BAT9400 Close the VSAM file

BAT9500 Print an output line

BAT9600 Isolate a MDE/HDE line

BAT9700 Justify a single line

TNDBAT2, External File Batch Delete processor

Manages all batch related Delete activities to the External File.

The Module's structure is as follows:

BAT0100 Close and then Open the VSAM file

BAT1000 Delete Key=

BAT2000 Delete Messages=userid

BAT3000 Delete Profile=userid

BAT4000 Delete Dmt=message

BAT5000 Delete Info=screen

BAT6000 Delete Hix=hixlevel

BAT7000 Delete Mix=originator

BAT8000 Delete Authentication=element

BAT9000 VSAM related errors

BATA000 Delete Saved=(setname,version)
BATB000 Delete Directory=name
BATC000 Delete Lists=name

BAT9100 Read a VSAM record

BAT9200 Write a VSAM record

BAT9500 Print an output line

BAT9600 Delete a VSAM record

TNDBLD2, Selection Panel Application Qualifier

Generates the SSE entries from the APPLICATION lists and fully qualifies them through security packages and exits and then returns to TNDBUILD to create the Application Selection Panel (CUA or not).

The Module's structure is as follows:

BLD0200 Qualify the SSE, including security calls
BLD0300 Build SSEs from ADB chain for ACF2 rules

BLD1000 Set up the criteria

BLD2000 Add DEFAULTs to SSE list

BLD3000 Add TERMINAL elements to SSE list

BLD4000 Add USER items to the SSE list

BLD5000 Collect the assigned PFKEY values

BLD6000 SSE reprocessing (ACF2, EXIT, SEQUENCE)

BLD6100 Sort the SSE into proper SEQUENCE

BLD7000 Set up SWA values for TNDBUILD processing

BLD9000 Add authorized items (AAE) to SSE list

TNDBLOCK, Control Block Allocation

Allocates, formats, and properly chains individual Network Director control blocks (mostly anchored in the PDA).

The Module's structure is as follows:

BLK0100 Allocation additional save areas

BLK1000 Primary control block allocation

BLK1100 Access method control block
BLK1150 Application Definition Block
BLK1200 Dispatchable Function Block
BLK1250 External File Record
BLK1300 Group Definition Block
BLK1350 Node Initialization Block
BLK1400 Active Network Element
BLK1450 Pending Parameter Element
BLK1490 Resource Entry block
BLK1500 Request Parameter List
BLK1550 System Measurement Record
BLK1590 Network Distribution List
BLK1600 Terminal Definition Block
BLK1650 User Definition Block
BLK1690 Key Definition Block
BLK1700 Help IndeX
BLK1750 Help Data Element
BLK1790 Access Information Block
BLK1800 Message IndeX
BLK1850 Message Data Element
BLK1890 System Directory block
BLK1900 Profile Data Element
BLK1950 Director Message Text
BLK1990 Site Definition Block

BLK2000 Secondary Control Block Allocation

BLK2100 Authorized Application Element
BLK2150 Authorized Interval Element
BLK2200 Authorized Terminal Element
BLK2250 Authorized User Element
BLK2300 Device Command Element
BLK2350 Initial Data Element
BLK2400 Work Area Element
BLK2450 Screen Selection Element
BLK2500 Application Action Item
BLK2550 Authorized Group Element
BLK2600 Network Element List
BLK2700 Application Load Balance

TNDBUILD, Application Selection Panel Formatter

Creates an application selection panel image for the current user (ANE).

The Module's structure is as follows:

BLD0100 AUTOLOGOFF=RETURN logic
BLD0200 Acquire the device lock

BLD1000 Establish panel format

BLD4000 Insert SSE into panel

BLD4100 Set up the Status area value

BLD5000 COMMANDS=YES area

BLD8000 A device just returned from a subsystem

BLD8100 Check for ACF2 users password expiration

BLD9000 A new device here, establish a session

BLD9100 Evaluate the Bind image

BLD9200 Logoff the user

TNDCASE, Topsecret/VSE Interface Routine

This exit routine provides an interface between the Network Director and Topsecret/VSE (used to be known as Sentinel).

The Module's structure is as follows:

EXT1000 Attempt to signon the user

EXT2000 Interpret the return code

EXT3000 Format a successful signon message

EXT8000 Topsecret/VSE Logoff exit point (TNDEXT24)

EXT9000 Topsecret/VSE Selection Validation (EXT15)

TNDCHECK, Control Block Validation

Validates that the control blocks and environment set up by PARM supplied the mandatory fields to continue. CHECK also provides the last location to cross-validate (or relationally edit) control blocks prior to actually letting the Dispatcher get control.

The Module's structure is as follows:

CHEC2000 Validate DEFAULT references

CHEC3000 Validate GROUP references

CHEC4000 Validate TERMINAL references

CHEC5000 Validate USER references

CHEC6000 Generate appropriate DFBs

CHEC9100 Validate AAE chain

TNDCINIT, Cinit User Data Area Processor

Evaluates the contents of the Cinit Ru User Data Area.

The Module's structure is as follows:

INIT1000 Examine the Cinit Ru User Data Area

INIT2000 Process the SSX buffer that came in

INIT3000 Someone has just returned from a Site

TNDCMBD, Combined Operand Display Command

Handles all "combined" display functions. CMBD receives control from REPT and only after REPT has decided that the request is too tough for it to handle.

The Module's structure is as follows:

CMBD1000 Display ??,APPLICATIONS=

- **CMBD1100** Autologoff, Users, Terminals, and NELs

CMBD2000 Display ??,GROUPS=

CMBD3000 Display ??,PROFILES=

CMBD4000 Display ??,TERMINALS=

CMBD5000 Display ??,USERS=

CMBD6000 Display ??,NETWORK-ELEMENTS=

CMBD7000 Display ??,SUBAREAS= or NETID=

TNDCMDS, Selection Panel Command Processor

Executes a command provided from the Application Selection Panel's Command line

The Module's structure is as follows:

- **CMD0100** Page Up or Page Down command

CMD1000 Flash command

CMD2000 Disconnect, Drop, or RESET

CMD3000 Logmode or Setmode command

CMD4000 Dim

CMD5000 Return

CMD6000 Logon command

CMD7000 Owner command

CMD8000 GROUP Command

- **CMD8100** Validate GROUP requested against AGEs
- **CMD8200** Establish the new GROUP

CMD9100 Allocate a one time Broadcast message

TNDCMPRS, Output Data Stream Compression

Compresses the output 3270 data stream using Repeat to Address orders to replace consecutive characters.

The Module's structure is as follows:

CMP1000 Compress the terminal buffer

CMP9100 Translate a single SBA in the buffer

CMP9200 Translate a single SBA

TNDCOLOR, Output Buffer Color Support

Replaces special characters in the output terminal buffer to appropriate 3270 command orders to provide for extended color support.

The Module's structure is as follows:

COLR1000 Initialization

COLR2000 Translate a special symbol

- **COLR2100** Translate a color attribute (Color device)
- **COLR2200** Translate a color attribute for non-color

COLR3000 Handle a two byte order

COLR4000 Handle a three byte order

COLR5000 Handle a special count type order

COLR6000 End of output data

COLR7000 Handle the activator character

TNDCUAP, Common User Access Processor

Processes the Application Selection Panel in CUA mode.

The Module's structure is as follows:

CUAP1000 Translate CUA Panel

CUAP2000 Interpret function key

CUAP3000 Subwindow generation logic

- **CUAP3100** Add in a *-------* line width wide

CUAP4000 Find subwindow cursor selection entry

TNDDAILY, Daily Item Processor

Manages any activities for each day (or so) for a user

The Module's structure is as follows:

DLY1000 Check out Daily messages possible message

DLY1100 Could be a message about Messages

TNDDECK, Network Administration Interactive Keyword Manipulation

Interprets the PPE already validated by ADMIN and processes the implied meaning within the Parameter Deck type definitions.

The Module's structure is as follows:

DECK3000 GLOBALS processing

DECK3100 GLOBALS Flag Processing

DECK8000 Check Network Element Authorization

TNDDELE, Message Delete Functional Routine

Deletes a message from the Message Facility

The Module's structure is as follows:

DELE1000 Update the Originator"s MIX

DELE2000 Delete the message

DELE3000 Update the other element"s MIX

TNDDISP, The Dispatcher

Manages the instruction cycles available to The Director. Passes control amongst the various DFBs as they become eligible for dispatch. DISP also maintains the actual Wait List and issues the real Operating System WAIT operation.

The Module's structure is as follows:

DISP1000 Initialization

DISP2000 Dispatch scan of the DFBs

- **DISP2100** Evaluate DFB for dispatch
- **DISP2200** Dispatch the eligible DFB
- **DISP2300** Add this DFB to the ECB Wait List
- **DISP2500** Prime the first time DFB for dispatch
- **DISP2600** Abend the DFB
- **DISP2700** The DFB has completed processing
- **DISP2800** Check for a Slowdown condition

DISP3000 Operating System Wait

- **DISP3100** Check for termination of Slowdown

DISP4000 Close the Terminal Access Method ACB

- **DISP4100** Reset ANE"s due to VTAM ACB closure

DISP5000 Open the Terminal Access Method ACB

- **DISP5100** ACQUIRE=YES terminal processing
- **DISP5200** Display the VTAM ACB vectors

DISP6000 Close the External File

DISP7000 Open the External File

TNDDONE, Termination Processing

Manages final termination of The Director as an operating system task.

The Module's structure is as follows:

DONE1000 Close the VTAM and External File ACBs

DONE2000 Produce shutdown statistics

DONE3000 Close the LOG

DONE4000 Freemain a bunch of storage

DONE5000 Close the TNDSAR medium (if necessary)

DONE9100 Freemain a single storage chain

TNDDSECT, Assembled DSECTs

Contains an expanded copy of all The Director's control block DSECTs and their assembled sizes, etc. This module is not executable. It's only purpose is to produce an Assembly listing of The Director's DSECTs (generally in an alphabetical order).

TNDDSI, Director System Interface

Provides a standard manner with which one Network Director can communicate with another one.

The Module's structure is as follows:

DSI1000 Set up the basic items

DSI2000 Establish connection with the other TND

- **DSI2100** Allocate the control blocks required

DSI3000 Ship the DSI request

- **DSI3100** Send the NDA over (SEND)
- **DSI3200** Get the answer (NCB returning)

TNDDUMP, Administrator Dump Services

Provides a mechanism with which an authorized Network Administrator can display main storage in hexadecimal format. This facility does not do any validity checking or protecting from S0C4 type abends. It relies upon the Administrator to be careful and the GLOBALS RECOVERY option to recover from any mistakes that might be made.

The Module's structure is as follows:

DUMP1000 Locate the specified address

- **DUMP1100** ADB
- **DUMP1150** UDB
- **DUMP1200** TDB
- **DUMP1250** GDB
- **DUMP1300** ANE
- **DUMP1350** HDE
- **DUMP1400** HIX
- **DUMP1450** PDE
- **DUMP1500** DMT
- **DUMP1550** MIX
- **DUMP1600** NIB
- **DUMP1650** DFB
- **DUMP1700** MDE
- **DUMP1710** NDL
- **DUMP1750** RPL
- **DUMP1800** EFR
- **DUMP1810** DIR
- **DUMP1850** PPE
- **DUMP1900** SMR
- **DUMP1910** KDB
- **DUMP1950** SDB
- **DUMP1975** AIB

DUMP2000 Build and write the output panel

- **DUMP2100** Format a single panel line
- **DUMP2200** STAE recovery routine

DUMP3000 Obtain the input "command"

- **DUMP3100** Parse the input command
- **DUMP3200** Up
- **DUMP3300** Down
- **DUMP3400** End or Quit
- **DUMP3500** Locate (get the next control block)
- **DUMP3600** Help
- **DUMP3700** Zap

DUMP4000 Check DUMP authorization

DUMP5000 Convert display Hex to Binary

TNDEDATS, Extended Data Stream Read Partition Query Logic

Issues the Read Partition Query to a device and then interprets the response from the device.

The Module's structure is as follows:

EDAT1000 Issue the Write Structured Field RPQ

EDAT2000 Parse a Query Reply buffer

- **EDAT2100** X"8181" usable area vector
- **EDAT2200** X"81A6 Implicit Partition
- **EDAT2300** X"8186" Color reply
- **EDAT2400** Set display area size
- **EDAT2500** Retain the RPQ response (WSF=KEEP)
- **EDAT2600** We"re setting a new screen size

TNDEDIT, The Editor

Manages all the processes associated with the full screen editing operations that are associated with Messages or Help panels.

The Module's structure is as follows:

EDIT1000 Set up the environment

EDIT2000 Build the Edit panel

- **EDIT2100** Set up the fixed area
- **EDIT2200** Insert a single EDL
- **EDIT2300** Write, then read the panel
- **EDIT2400** Reset EDL flag settings
- **EDIT2500** Possible update of reference Time and Date

EDIT3000 Parse through the input data

- **EDIT3100** Place the cursor, count the lines

EDIT4000 Process the prefix commands

- **EDIT4100** Parse prefix command
- **EDIT4200** Validate the prefix command

EDIT5000 Translate the PFKEY meaning

EDIT6000 Process a primary command

TNDEDIT1, Editor Prefix Command Processor

Processes the prefix commands associated with a single Edit Line (EDL).

The Module's structure is as follows:

EDT11000 A (After)

EDT12000 B (Before)

EDT13000 C (Copy)

EDT14000 D (Delete)

EDT15000 DD (Delete range)

EDT16000 I (Insert lines)

EDT17000 M (Move)

EDT17500 R (Replicate)

EDT18000 CC (Mark Copy range)

EDT19000 MM (Mark Move range)

EDT19200 Acquire a basic EDL

EDT19300 Move a block of lines

EDT19400 Copy a block of lines

EDT19500 Copy a single line After

EDT19600 Move a single line After

EDT19700 Copy a single line Before

EDT19800 Move a single line Before

EDT19900 Validate block operations have a target

TNDEDIT2, Editor"s Primary Command Processor

Processes the commands entered on the Editor's primary command line.

The Module's structure is as follows:

EDT21000 Change

- **EDT21100** Change the located string

EDT22000 Copy

- **EDT22100** Search a MME for the copy target

EDT23000 Save, End or File

- **EDT23100** Locate a MDE, etc. to process
- **EDT23200** Fill up the data area
- **EDT23300** Update the originator"s MIX
- **EDT23400** Verify even number of activator characters
- **EDT23500** Save a HDE

EDT24000 KEY

EDT25000 Locate

EDT28000 Split

EDT29100 Parse the input arguments (Locate/Change)

TNDEDIT3, Editor, Primary Commands

Processes the primary commands detected by EDIT.

The Module's structure is as follows:

EDIT6000 Process a primary command

- **EDIT6100** Bottom or Top
- **EDIT6200** Invoke EDIT2 to process this command
- **EDIT6300** Up or Down
- **EDIT6400** Format
- **EDIT6500** Help
- **EDIT6600** Prefix
- **EDIT6700** Profile
- **EDIT6800** Quit
- **EDIT6850** Reset
- **EDIT6900** Ruler
- **EDIT7100** Show Directory
- **EDIT7200** Distribution List

TNDEDIT4, Editor, Time and Date Update Logic

Updates disk based message indexes with the viewed time.

The Module's structure is as follows:

EDIT1000 Update an individual MIX

TNDEFLD, TSO Pre-prompt Exit

This Pre-prompt exit is available for installation usage to bypass TSO/E's insistence upon a prompt panel. This should simplify (support?) the Director's SSI concept for TSO.

TNDEHIX, Edit Help Index Elements

Receives control as a function of an authorized help user that issued the "EDIT HIX.....' command from the command line.

The Module's structure is as follows:

EHIX1000 Set up the basic environment to work in

EHIX2000 Present the HIX modification panel

EHIX3000 Process the input (Alter the HIX)

EHIX3100 Check out PFKEY usage
EHIX3300 Update the HIX

TNDEXT01, Sample Exit 01 (Security)

This exit provides an example of how a Network Director security exit could be coded. It check to see if a user attempting to logon is in a table and if the password provided is acceptable.

The Module's structure is as follows:

EXT1000 Locate a table entry for the userid

EXT2000 Check for special conditions

TNDEXT04, Sample Exit 04 (Log Review)

This sample exit looks at each message as it is being issued by a Network Director module via TNDISSUE and can edit the text that is placed onto the terminal screen or into the LOG for subsequent display.

TNDEXT06, Sample Exit 06 (Operator Input)

This sample exit detects an input command called REFRESH from the IUCV interface and will cause a Selection Panel to be rewritten to the Network Element named in the REFRESH command (It can be a LUname or a userid). This will be done only if the user/device is currently connected to The Network Director and has a Application Selection Panel on it.

TNDEXT08, Sample Exit 08 (Administration Preview)

This exit provides a sample of how a Network Director exit can be coded that implements a new local command. The exit looks for the Network Administrator entering a "?" and replies with a message contained within the exit

The Module's structure is as follows:

EXT1000 Display survey detail records

EXT2000 Display survey detail records

EXT2100 Format numeric value for the panel

TNDEXT12, Sample Exit 12 (CLSDST PASS)

This sample exit provides an example of how to implement a couple of specialized functions when a terminal user has just selected a subsystem.

The Module's structure is as follows:

EXT1000 Check the basic selection

EXT2000 Check for CMS

TNDEXT15, Sample Exit 15 (Selection Review)

This sample restricts certain terminals to only certain Applications. This exit has a table of devices that are restricted and then a table that are valid for the identified devices.

TNDEXT19, Sample Exit 19 (Group Assignment)

This exit provides an example of how a Director exit can be coded to take advantage of dynamically assigning GROUPs within the Director based on an external facility. In this case, ext19 extracts some bytes our of the ACF2 logonid record (specifically, the UID string) for the purpose of setting a Director GROUP relationship.

TNDEXT22, Sample Exit 22 (Broadcast Review)

This exit provides an example of how a Director exit can be coded to place a message into the LOG each time a dynamically originated message is shipped to a device.

TNDEXT24, Sample Exit 24 (Logoff)

This exit sends a special set of characters to a 7171 device when it logs off to cause it to disconnect.

TNDEXT25, Sample Exit 25 (Authentication)

Provides an example of "additional authentication" associated with a user logging onto the system.

The Module's structure is as follows:

EXT1000 Prompt the operator

EXT1100 Generate the challenge

EXT2000 Compute the response

TNDEXT26, Sample Exit 26 (Initialization)

This routine causes a "USER" DFB associated with EXT33 to be allocated within The Network Director's dispatching environment. When the DFB is dispatched, TNDEXT33 will receive control from TNDDISP and operate as a DFB task.

TNDEXT31, Sample Exit 31 (Return)

This exit displays a theoretical "survey" questionaire to a terminal user every so often. The responses are then stored into the External File for later use.

The Module's structure is as follows:

EXT1000 Display the survey panel

EXT2000 Process survey response

TNDEXT33, Sample Exit 33 (User DFB)

This sample DFB exit wakes up periodically to determine if a specific VM based user has received any "mail".

TNDFKEYS, Establish correct Function Key values

Determines which Function Key area should be displayed on the device, based upon the Options or setting requested by the controlling CSECT.

The Module's structure is as follows:

FKEY1000 Search function key table for constants

FKEY2000 Look for a RESOURCE for this definition

TNDFLASH, Generate volume output to a device

Generates alternating output (LOGO and ID-LOGO) to one or more devices to test throughput, etc.

The Module's structure is as follows:

FLSH0100 Command authorization check

FLSH1000 Attach FLASH DFB for actual device I/O

FLSH2000 Possible FLASH output to a device

FLSH3000 Attach terminal oriented FLASH DFB to do I/O

FLSH4000 Flash an individual device

FLSH4100 Get the RPL we need
FLSH4200 Write the output buffer
FLSH4300 Completed the FLASH interval

FLSH5000 Parse the FLASH command

TNDFORM, FORMAT Function

FORM manages the conversion from MDE/HDE type format to EDL type format for the Editor functions. The FORMAT command is also accomplished by processing a EDL chain through this module. FORMAT is implied by a conversion request.

The Module's structure is as follows:

FORM1000 Convert MDE/HDE to EDL chain

- **FORM1100** Isolate a line in the MDE/HDE
- **FORM1200** Create a single EDL
- **FORM1300** Check for a continuation MDE
- **FORM1400** New item, generate some new lines

FORM2000 Format EDL chain

- **FORM2100** Locate a range to format
- **FORM2200** Locate a token
- **FORM2300** Right and left justify a line
- **FORM2400** Rechain the EDLs together
- **FORM2500** Insert a token

TNDGLOB, GLOBALS Processing Routine

Is called to initialize various secondary processes based upon the setting of GLOBALS flags in the PDA

The Module's structure is as follows:

GLOB1000 VM Initialization items

- **GLOB1100** Topsecret/VM Initialization

GLOB2000 MVS Initialization items

- **GLOB2100** ACF2/MVS Initialization
- **GLOB2200** IFAUSAGE REGISTER Director for MULC

GLOB3000 Initialize the LOG buffer

GLOB4000 Get the basic OS information

TNDGMM, Network Director/CICS SSI Routine

Replaces the CICS provided DFHGMM to implement automated signon concepts (SSI) within CICS. This routine is intended to operate with CICS 2.1 or lower.

The Module's structure is as follows:

GMM1000 Locate TNDTABLE in the TCT

GMM2000 Search TNDTABLE for this specific LU

GMM3000 Invoke the signon program

GMM4000 Return this LU to VTAM

GMM5000 Start the first transaction

GMM6000 House cleaning the TNDTABLE

GMM7000 Extract the LOGON msg from the CINIT

TNDGMMSA, CICS/ESA SSI Routine

This module replaces the CICS/ESA provided DFHGMM at locations interested in implementing the Director's SSI concepts within CICS/ESA 3.2.1 or higher.

The Module's structure is as follows:

GMM3000 Invoke the Sign-on program

> **GMM3100** Invoke the old style SNP
> **GMM3200** Sign the user on via the SIGNON command

GMM4000 Return this device to VTAM (Signon failed)

GMM5000 Start the first transaction

GMM7000 Extract the Logon message from CINIT

TNDGMSG0, IMS SSI Greeting Messages Exit

NRSGMSG0 intercepts the DFS3650 message that is sent to the device after successful signon and initiates an INITIAL-COMMAND, if appropriate. This is done by entering the Communications Analyzer directly after properly obtaining a new input buffer from IMS via DFSPOOL.

The Module's structure is as follows:

GMSG1000 Routine Initialization

GMSG2000 DFS3650 handling

- **GMSG2100** Locate the TNDTABLE
- **GMSG2200** Garbage collection in TNDTABLE
- **GMSG2300** Locate a TBL for this device
- **GMSG2400** Clean up current buffers, etc.
- **GMSG2500** Allocate a new input buffer
- **GMSG2600** Setup for return to IMS, then back to GMSG0
- **GMSG2700** Initiate the new transaction

GMSG3000 DFS2467 Processing

GMSG4000 DFS3649 Processing

GMSG9100 Free the WRK Area

TNDIDMS, IDMS Single System Image

This routine is designed to operate in IDMS/DC to automate the process associated with The Network Director's SSI concepts. TNDIDMS links to RHDCSNON to accomplish the signon so that all standard IDMS/DC facilities apply.

The Module's structure is as follows:

IDMS1000 Simulate the BYE task

IDMS2000 Invoke the IDMS signon program

IDMS3000 Return this device to VTAM

IDMS4000 Start the first transaction

TNDINFO, Info Command Processor

oversees the operation of the generalized INFO or HELP facility. This is the TNDHELP or TNDINFO internal application.

The Module's structure is as follows:

INFO1000 Initialization

INFO2000 Check for PFkey and prompt type request

- **INFO2100** Translate Pfkey meaning

INFO3000 Display direct request

INFO4000 Text string request

- **INFO4100** Find a text string token
- **INFO4200** Translate the text word

INFO5000 Implied command

INFO6000 Process Info primary command

- **INFO6100** End or Up
- **INFO6200** Down
- **INFO6300** Edit or Delete
- **INFO6400** Parse the Find or Locate command
- **INFO6500** Search for the character string
- **INFO6600** Top

INFO7000 Display the output screen

TNDINFO2, Info Output Processor

Generates the output panel from the Info/Help facility.

The Module's structure is as follows:

INFO1000 External File is down or no Info around

INFO2000 Build and write the Info panel

- **INFO2100** Set up various counters and variables

INFO3000 Collect the input data

INFO4000 Create the EDL chain

INFO5000 Create the text string

TNDINIT, Initialization

Sets up The Network Director's basic internal environment. This includes establishing the PDA, the storage pools, and opening up the External files.

The Module's structure is as follows:

INIT1000 Allocate the PDA, storage pools, AMODE

INIT2000 Open the Parameter Deck input file

INIT3000 Open the output TNDLOG file

TNDINPUT, Receive Any DFB Logic

Executes as an independent DFB (one per RPLS=) and is capable of receiving any input transmission intended for the Director. After input, the virtual RPL is checked for RPLDATA bits in RPLCNTDF to determine if normal input is to be processed or if special logic is required.

The Module's structure is as follows:

INP0100 Check out discarded input

INP1000 Acquire input buffer and RPL

INP2000 Issue Receive Any

INP2100 Look at the sense codes that came in
INP2200 Try to locate the ANE for this input

INP3000 Interpret non CONTROL=DATA input

INP3100 Produce non data type message
INP3200 Cancel was just received
INP3300 Quiesce at end of chain (QEC)
INP3400 Release Quiesce (RELQ)
INP3500 Lustat
INP3600 Request shutdown (RSHUTD)
INP3700 Chase

INP4000 Create an ANE

INP4100 Try to locate the ANE again

INP5000 Post an existing DFB awake

INP6000 Create a new DFB to process the input

INP7000 Handle input from an inactive device

INP8000 Handle just a response from the other LU

INP9000 Automatic release LU from Inactive list

INP9100 Simlogon a device

INP9200 Put the device onto the Inactive List

INP9300 Issue RESETSR for a session

INP9400 Acquire the ANE lock

INP9500 Release WAEBUF work i/o area

TNDINQA, Inquire Application Status

Issues the appropriate sequence of instructions to establish the availability of a single APPLICATION.

The Module's structure is as follows:

INQA1000 Interrogate VTAM application status

INQA1100 Inquire Appstat

INQA2000 Obtain available via VTAM SPO interface

INQA2100 Issue the SPO command

TNDINTCD, IntelliCard Authentication Interface

Provides the process to validate the user entering the system against the IntelliCard device and algorithms.

The Module's structure is as follows:

ITCD1000 Prompt the operator with the challenge

ITCD1100 Generate the challenge

ITCD2000 Compute the proper response

TNDINTX0, IMS SSI Initialization Exit

This allocates the TNDTABLE via IMS Callable Services for subsequent usage by the TNDLGNX0 and TNDGMSG0 routines. The address passed back to IMS that points in storage actually points at a list of addresses. The Director's IMS routines use the first word in this list to point at the TNDTABLE and the other words can be used by other User Installation exit routines.

The Module's structure is as follows:

INTX1000 Obtain space for the TNDTABLE

INTX2000 Set up table of table addresses

TNDKEYS, Pfkey Interpretation

Provides a standard routine to interpret the meaning of any particular PFKEY and its value from the Network Element's Profile (PDE).

The Module's structure is as follows:

KEYS1000 Process a PA key value

KEYS2000 Interpret a PF key

TNDLGNX0, IMS SSI Logon Exit

This routine collects the information originating from outside the IMS system and passed to it via the CINIT RU User Data Area. If the information string present contains SSI data that was sent from The Director, we collect and store it in the TNDTABLE for subsequent processing by DFSGMSG0 and reformat the input buffer for proper processing by IMS.

The Module's structure is as follows:

LGNX1000 Validate the CINIT for SSI/SSX format

LGNX2000 Allocate TNDTABLE

LGNX3000 Place SSI data into TNDTABLE

LGNX4000 Reformat the Signon data buffer for IMS

LGNX5000 Locate the TNDTABLE

TNDLIST, Distribution List Co-ordinator

Co-ordinates the loading of user associated distribution lists.

The Module's structure is as follows:

LIST1000 Parse the incoming LIST command

LIST2000 Load all the Lists for this user

LIST2100 See if the NDL is already in storage

LIST3000 Invoke SHOW to process the list items

TNDLOCAT, Control Block Location

Scans a control block chain and locates the first block whose name matches that provided by the caller. This routine is invoked via the TNDFIND macro.

The Module's structure is as follows:

LOCT1000 Dispatchable Function Block

LOCT2000 External File Records

LOCT3000 Group Definition Blocks

LOCT4000 Active Network Elements

LOCT5000 Terminal Definition Blocks

- **LOCT5100** Evaluate TDB Netid and Subareas

LOCT6000 User Definition Blocks

- **LOCT6100** Evaluate UDB Netid and Subareas
- **LOCT6200** Help IndeX
- **LOCT6400** Help Data Element
- **LOCT6600** Message IndeX
- **LOCT6800** Message Data Element

LOCT7000 Profile Data Element

- **LOCT7100** PDE Look Aside Buffer
- **LOCT7200** Director Message Text
- **LOCT7400** Site Definition Blocks
- **LOCT7600** Access Information Blocks
- **LOCT7800** Application Definition Blocks
- **LOCT7900** System Directory

LOCT8000 Key Definition Blocks

- **LOCT8500** Resource Blocks
- **LOCT8900** Network Distribution Lists

TNDLOG, Log and Storage Queue Services

Places Director messages on the output TNDLOG file listing. Also maintains the central storage queue. Individual message can also be written to the operator's console, depending upon the GLOBALS WTO= operand and the message class attribute of the message.

The Module's structure is as follows:

LOG1000 Locate requested message text

LOG2000 Replace symbolic text

- **LOG2100** Compress the blanks
- **LOG2200** Insert the message number and time
- **LOG2300** Eliminate any PASSWORD= values

LOG3000 Check for extended message function

- **LOG3100** Write to Operator"s console
- **LOG3200** Abend logic
- **LOG3300** Write To Log (WTL) logic

LOG4000 Insert into Log Storage Queue

- **LOG4200** End of storage queue, re-chain the LBEs
- **LOG4300** Insert a single LBE

LOG5000 Write the message onto the output TNDLOG

LOG6000 Look for Administrators in Monitor mode

LOG7000 Watch for message loops

LOG8000 Send a message via VM SMSG

TNDLOGON, Logon and Logoff Processor

Handles the logging on and off activities associated with a Network Element. Co-ordinates the security package interfaces.

The Module's structure is as follows:

LOGN1000 Logoff Processing

LOGN2000 Locate the TDB and UDB items that apply

- **LOGN2100** Set up the GROUP connection
- **LOGN2200** NEW-PSWD=VALIDATE logic
- **LOGN2300** Validate the password value
- **LOGN2400** Secondary Authentication?
- **LOGN2500** Validate the Terminal characteristics
- **LOGN2600** Check allowable Ids
- **LOGN2700** Get the password from a LU1 device

LOGN4000 Validate the USER characteristics

- **LOGN4100** Check the allowable TIME and DAY intervals
- **LOGN4200** Check authorized terminals
- **LOGN4400** See if the Userid is already in use

LOGN6000 Failed Logon attempt

- **LOGN6100** Inactive List for TRIES

LOGN7000 Logon was successful

LOGN8000 Validate the Account code

LOGN9100 Validate AIE based information

LOGN9200 Validate ATE based information

LOGN9300 Record Logon event

LOGN9400 Release some MX pool storage

LOGN9500 Compute the security system duration

TNDLOST, VTAM LOSTERM Processor

Manages all DFB task processing associated with the LOSTERM and RELREQ exit. This DFB is created in VTAM IRB code to manage work under a DFB task instead of in the IRB.

The Module's structure is as follows:

LOST1000 Locate the ANE

LOST2000 CLSDST the device

LOST3000 Clean up any residual stuff

TNDLTRM, Line Terminal Buffer Processing

Accepts as input a formatted 3270 buffer as built by the Director. Processes the output buffer and attempts to simulate the requested action within the limits of capability of a line oriented (LU1) type device.

The Module's structure is as follows:

LTRM1000 Build physical output buffer

LTRM1100 Shuffle the buffer

LTRM2000 Build physical input buffer with SBA"s

LTRM2100 Process incoming escape sequences

LTRM3000 Clear the screen

TNDMAIN, Primary Operating System Task Control

Passes control between the primary operating system level tasks. This is the basic logic flow for the entire Director.

The Module's structure is as follows:

MAIN1000 Display environment related items

TNDMAINT, Environment Maintenance

Is a system DFB that performs duties associated with the External File maintenance and timer type expirations.

The Module's structure is as follows:

MNT1000 External File Initialization

- **MNT1100** DMT initialization
- **MNT1200** Broadcast Initialization
- **MNT1300** System Directory Initialization

MNT2000 External File Updates

- **MNT2100** Director Message Text
- **MNT2200** Profile Data Element
- **MNT2300** Network Distribution Lists
- **MNT2400** Message Data Element
- **MNT2500** Help IndeX
- **MNT2600** Help Data Element
- **MNT2700** Access Information Block
- **MNT2800** Control Blocks
- **MNT2900** System Directory Entries

MNT3000 Message Expiration Scan

- **MNT3100** In storage MDE scan
- **MNT3200** Check for oldest message on External File

MNT4000 Check Authorization

MNT5000 External File Initialization

- **MNT5300** Obtain a MDE record
- **MNT5400** Update or make a MIX

MNT9100 Update or add a record to the External File

MNT9200 Release MX pool storage

MNT9300 Obtain a record from the file

MNT9400 Delete one MDE

MNT9500 Delete a record from the External File

TNDMNTR, Network Monitor

Monitors the status of various portions of the logical network on a periodic basis (the interval is set via GLOBALS TIMER=)

The Module's structure is as follows:

MNTR1000 Evaluate Application status

- **MNTR1100** Interrogate VTAM application status
- **MNTR1200** Process application status change
- **MNTR1300** Notify network of status change
- **MNTR1400** Check application availability (DAY/TIME)
- **MNTR1500** Get USERVAR value
- **MNTR1600** Map ADB to SDB for status
- **MNTR1700** Evaluate load balanced Application

MNTR4000 Check USER availability (DAY/TIME)

MNTR5000 Check TERMINAL availability (DAY/TIME)

MNTR6000 Check GROUP availability (DAY/TIME)

MNTR7000 Interrogate Site Definition status

MNTR9000 A quick look at Daily items

MNTR9100 Log this network element off

MNTR9200 Check DAY and TIME values

MNTR9300 Allocate an RPL and NIB

MNTR9400 Reset the daily News bit (possibly)

TNDMNT1, Environment Maintenance (Control Blocks)

is a sub-program of MAINT and is called if there are any control blocks that need to be written to the External File.

The Module's structure is as follows:

MNT1000 Globals

MNT2000 Default

- **MNT2100** Default Fixed Fields
- **MNT2200** Default Applications
- **MNT2300** Move a Label
- **MNT2400** Default Logo
- **MNT2500** Default Id-logo

MNT3000 Application

- **MNT3100** Application Fixed Fields
- **MNT3200** Application Actions
- **MNT3300** Application Initial-data
- **MNT3400** Application Days and Time
- **MNT3500** Application Privilege
- **MNT3600** Application Balance

MNT4000 User

- **MNT4100** User Fixed Fields
- **MNT4200** User Applications
- **MNT4300** User Terminals
- **MNT4400** User Time and Days
- **MNT4500** User Logo=
- **MNT4600** User Subareas
- **MNT4700** User Terminals

MNT5000 Invoke MNT2 for TERMINAL, SITE, GROUP

MNT6000 Profile

MNT9100 Update/Add a record to the External File

MNT9200 Refresh the buffer to maximum size

MNT9300 Check for trailing records

MNT9400 Check record buffer for available space

TNDMNT2, Environment Maintenance (Control Blocks)

is a sub-program of MNT1 and is called to determine if any Terminal, Site, Key or Group control block should be saved to the External File.

The Module's structure is as follows:

MNT1000 Terminal

- **MNT1100** Terminal Fixed Fields
- **MNT1200** Terminal Applications
- **MNT1300** Terminal Applications
- **MNT1400** Terminal Time= and Days=
- **MNT1500** Terminal Logo=
- **MNT1600** Terminal Applications
- **MNT1700** Terminal Groups=

MNT2000 Site

- **MNT2100** Site Network-elements

MNT3000 Group

- **MNT3100** Group Fixed Fields
- **MNT3200** Group Applications
- **MNT3300** Group Terminals
- **MNT3400** Group Network-elements
- **MNT3500** Group Time and Days
- **MNT3600** Group Logo

MNT4000 Keys

- **MNT4100** Keys Fixed Fields

MNT5000 Resource

MNT9100 Update/Add a record to the External File

MNT9200 Get and Delete record from VSAM file

MNT9300 Check for trailing records

MNT9400 Check record buffer for available space

TNDMNT3, Storage Balancing

is a sub-program of MAINT and is called to accomplish logic associated with storage balancing algorithms.

The Module's structure is as follows:

MNT5000 STORAGE-BALANCE Logic

- **MNT5100** Scan a single GSA chain for information
- **MNT5200** Free a storage element

TNDMSG, Message Facility Processor

Co-ordinates the various portions of the Message Facility

The Module's structure is as follows:

MSG1000 Send the Primary Messages Menu

- **MSG1100** Build the suffix items
- **MSG1200** Build fixed panel portions
- **MSG1400** Process the MME

MSG2000 Output the panel we produced

MSG3000 Process the operator response

- **MSG3100** Validate the operator"s actions
- **MSG3200** Verify the Site combinations
- **MSG3300** Process a single activity
- **MSG3400** Check out the PFKEY and Command area
- **MSG3500** Process a primary command
- **MSG3600** Scan the MDE we"re working with
- **MSG3700** Point at the MIX entry (if disk based)

MSG4000 Create the MMEs for this user

MSG5000 Termination logic

MSG9200 Validate the MMECUR pointer

MSG9300 Acquire a MX pool SSE

MSG9400 Free a storage area

TNDMSGS, Internal Message Text

Contains the message text and attributes for The Director until the External File is active and any locally modified messages are loaded.

TNDMSG1, MSG Message Element Construction

Establishes the MME used for the primary message menu

The Module's structure is as follows:

MSG1000 Create the MME for this network element

- **MSG1100** Process storage MDE"s
- **MSG1200** Collect MIX based messages
- **MSG1500** Acquire a MME

MSG2000 Establish MME order

MSG9300 Build a single MME entry

MSG9400 Check for GROUP NETWORK-ELEMENT Inclusion

TNDMSG2, Process Message Actions

Executes the Primary Message Menu actions

The Module's structure is as follows:

- **MSG0100** Prepare for a new created message

MSG1000 Deleting a message

MSG2000 Extending a message"s expiration time

- **MSG2100** Update the destination"s MIX with new date

MSG3000 Edit an existing message

MSG4000 Print a message

MSG5000 Send a message

- **MSG5100** Locate the distribution list
- **MSG5200** Send a copy of the base message
- **MSG5300** Delete the base message

MSG6000 View a message

MSG7000 Redirect a message

- **MSG7100** Create a new MDE for Redirect
- **MSG7200** Update the originating MIX

TNDMVS, OS Operating System Interface

provides all the OS or GCS operating system services not provided by NRSMVS. This includes OS dependent services that are specific to The Director.

The Module's structure is as follows:

TNDOPEN Open a DCB
TNDCLOSE Close a DCB
TNDGET Acquire a record from a sequential file
TNDPUT Output a logical record to sequential file
TNDSTAE Establish and manage recovery logic
STAEEXIT STAE exit routine
TNDSMF Produce a SMF record
TNDCHK Issue CHECK against an RPL
TNDLOAD LOAD a program into virtual storage
TNDDELT Delete a program from virtual storage
TNDBLDL Locate (BLDL) a program in the library

TNDNRP, Network Request Processor

Handles requests within The Director's partition/address space that originated via the Network System Interface (NSI) via the LU0 type interface.

The Module's structure is as follows:

NRP1000 Establish the session

- **NRP1100** Refuse the NSI request (CLSDST)
- **NRP1200** Acquire the NCB
- **NRP1300** Accept the session (OPNDST ACCEPT)
- **NRP1400** Acquire a NIB, RPL, and ANE

NRP2000 Wait for the NDA to arrive

NRP3000 Process the NSI request

- **NRP3100** Send message
- **NRP3200** Stack command

NRP4000 Send the reply back

TNDNRP2, IUCV Network Request Processor

Handles requests within The Director's partition/address space that originated via the IUCV based Network System Interface.

The Module's structure is as follows:

NRP1000 Establish the session

- **NRP1100** Refuse the NSI request (CLSDST)
- **NRP1400** Acquire a NIB, RPL, and ANE

NRP2000 Wait for NDA to arrive

NRP3000 Process the NSI request

- **NRP3100** Send message
- **NRP3200** Stack command

NRP4000 Send the reply back

TNDNRP62, Network Request Processor for LU 6.2

Handles requests within the Director's partition/address space that originate via the Network System Interface (NSI) using the LU 6.2 communication mechanism.

The Module's structure is as follows:

NRP1000 Establish the session

- **NRP1100** Refuse the NSI request (CLSDST)
- **NRP1300** We"ll go with the conversation

NRP2000 Process the FMH5

- **NRP2100** Locate the NCB in the PIP data
- **NRP2200** Mapped conversation (CICS)
- **NRP2300** Unmapped conversation

NRP3000 Process the NSI request

- **NRP3100** Send message
- **NRP3200** Stack command

NRP4000 Send the reply back

NRP6000 Acquire a RPL and ANE

TNDNSI, Network System Interface

Provides a standard manner via LU0 with which an application program may communicate with The Director.

The Module's structure is as follows:

NSI1000 Validate request

- **NSI1100** Validate Send message request

NSI2000 Establish connection with The Director

- **NSI2100** Create the ACB, NIB, and RPL required
- **NSI2200** Initiate the VTAM session (OPEN, SETLOGON)
- **NSI2300** Request the session (REQSESS)
- **NSI2400** Finalize connection (OPNSEC)

NSI3000 Ship NSI request to the Director

- **NSI3100** Send NDA across (SEND)
- **NSI3200** Get the answer (NCB returning)

NSI4000 Terminate the session (TERMSESS)

NSI8000 NSEXIT (when NSI=NO over in The Director

NSI9000 SCIP (when session is established)

TNDNSIX1, NSI, Assembler example

Demonstrates how to utilize The Network Director's Network System Interface (NSI) to transmit a note to the message facility within The Network Director's address space.

TNDNSI62, LU 6.2 Network System Interface

Provides a standard manner with which an application program may communicate with The Director via LU 6.2 from a batch (TSO) environment.

The Module's structure is as follows:

NSI1000 Validate request

- **NSI1100** Validate Send message request

NSI2000 Establish connection with The Director

- **NSI2100** Create the ACB, NIB, and RPL
- **NSI2200** Initiate the VTAM session (OPEN, SETLOGON)
- **NSI2300** Set up the 6.2 basis (OPRCNTL)
- **NSI2500** Establish the session (ALLOC)

NSI3000 Ship NSI request

- **NSI3100** Send the NDA (SEND)
- **NSI3200** Get the answer back (NCB returning)

NSI4000 Terminate the session (DEALLOC)

NSI8000 ATTN exit

TNDNSXT, NSEXIT Processing Logic

Manages all DFB task processing associated with the NSEXIT VTAM exit. This DFB is created in VTAM's NSEXIT exit code.

The Module's structure is as follows:

NSXT1000 Locate the ANE

- **NSXT1100** Map the RU identification area

NSXT2000 NSPE Error Logic

NSXT3000 Notify Error Logic

NSXT4000 Clean Up Error Logic

NSXT5000 CLSDST the Logical Unit

NSXT6000 Clean up the residual stuff

NSXT7000 Interpret the sense codes

- **NSXT7100** Extended sense code descriptions

NSXT8000 NSPE and Notify sense check logic

TNDNTFY, Asynchronous Notification

Informs appropriate Network Elements of a change within the network (like Application status) or handles general Broadcast duties.

The Module's structure is as follows:

NTFY1000 Selection Status Update

- **NTFY1100** Check the SSE status
- **NTFY1200** Rewrite the panel
- **NTFY1300** Logoff residual users
- **NTFY1400** Auto-select Logic Check

NTFY2000 Broadcast message

- **NTFY2100** Check the TO= value in the MDE against ANE
- **NTFY2200** Rewrite the panel

NTFY3000 Synchronize Notify Tasks

NTFY9000 User Exit Interface

NTFY9100 Undim the device

NTFY9200 Count updates and conditional release lock

TNDNTWK, Network Monitor

Monitors the status of the active Network Elements on a periodic basis (the interval is set via GLOBALS TIMER)

The Module's structure is as follows:

NTWK2000 Reset statistics

- **NTWK2100** Add the current interval in
- **NTWK2200** Reset the interval counters

NTWK7000 Look at active Network Elements

- **NTWK7100** Check TERMINAL availability
- **NTWK7200** Check USER availability
- **NTWK7300** Check GROUP availability
- **NTWK7400** Reactivate Terminals
- **NTWK7500** Check STATUS-INTERVAL
- **NTWK7600** Evaluate DIM intervals
- **NTWK7700** Check if Defaults have changed
- **NTWK7800** Check if PROFILE has changed
- **NTWK7900** Check if APPLICATIONs have changed

NTWK8000 Refresh ANE based counts

- **NTWK8100** Produce APPLCNTS type SMRs
- **NTWK8200** Held RPL Garbage Collection
- **NTWK8300** Upgrade Iteration Counters (maybe)
- **NTWK8500** Reset control block flags

NTWK9000 Handle effected Network Element

NTWK9100 Log this element off

TNDOPCMD, Operator Command Processor

Accomplishes all processing associated with an operator command.

The Module's structure is as follows:

OPCM1000 Broadcast

OPCM2000 Cancel

- **OPCM2100** Cancel DFB
- **OPCM2200** Cancel Group
- **OPCM2300** Cancel Terminal
- **OPCM2400** Cancel User
- **OPCM2500** Cancel Network Element

OPCM3000 Close

- **OPCM3100** Close External
- **OPCM3200** Close Tam
- **OPCM3300** Close TNDSAR

OPCM4000 Hold

- **OPCM4100** Hold Application and/or Targets
- **OPCM4200** Hold DFB
- **OPCM4300** Hold Group
- **OPCM4400** Hold NSI
- **OPCM4500** Hold User
- **OPCM4600** Hold Terminal
- **OPCM4700** Hold Network Element
- **OPCM4800** Hold Site
- **OPCM4900** Hold AIB

OPCM5000 Open

- **OPCM5100** Open External
- **OPCM5200** Open Tam
- **OPCM5300** Open TNDSAR

OPCM6000 Release

- **OPCM6100** Release Application
- **OPCM6150** Release AIB
- **OPCM6200** Release DFB
- **OPCM6250** Release Held-appls
- **OPCM6300** Release Group
- **OPCM6400** Release NSI
- **OPCM6450** Release News
- **OPCM6500** Release User
- **OPCM6600** Release Terminal
- **OPCM6700** Release Network Element
- **OPCM6800** Release Site
- **OPCM6900** Release Inactive
- **OPCM6950** Release Monitor

OPCM7000 Simlogon

OPCM7500 Disconnect

OPCM8000 Stop

OPCM8100 Delete
OPCM8200 Delete Group
OPCM8300 Delete Terminal
OPCM8400 Delete User
OPCM8500 Delete Application
OPCM8600 Delete Profile
OPCM8700 Delete Network Element
OPCM8800 Delete Site
OPCM8900 Delete AIB
OPCM8950 Delete Keys

OPCM9000 Check authorization level

OPCM9100 Possible forced Logoff

OPCM9200 Check UDB for proper criteria

OPCM9300 Check TDB for proper criteria

OPCM9400 Check ANE for proper criteria

TNDOPER, Operator Services

Provides an interface for the operating system console(s)

The Module's structure is as follows:

OPER1000 Initialize Operator Communications

OPER1100 Set up the ANE for the Operator

OPER2000 Obtain input from the Operator

OPER2200 Try to connect Operator to Administration

OPER3000 Parse the input

OPER4000 Process the request

OPER4200 Invoke Program Operator
OPER4400 Invoke Display Processor
OPER4500 Invoke Operator Services
OPER4600 Invoke Deck Manipulation
OPER4700 Invoke Dump Storage facility
OPER4800 ACF2 Command

OPER5000 Issue the WTOR

TNDPARM, Initial Parameter Deck Processor

Interprets the Configuration Parameters and builds the appropriate control blocks and environment for the Director's Dispatcher.

The Module's structure is as follows:

PARM1000 Process a parameter statement

- **PARM1100** Get a record image
- **PARM1200** Invoke PARMS to parse it

PARM2000 Allocate and fill in a control block

- **PARM2100** Application
- **PARM2200** Default
- **PARM2300** Globals
- **PARM2400** Group
- **PARM2500** Profile
- **PARM2600** Terminal
- **PARM2700** User
- **PARM2800** Site
- **PARM2900** ACF2

PARM3000 Reload

- **PARM3100** Keys
- **PARM3200** Directory
- **PARM3300** Resource

PARM7000 Open the External File

TNDPARMS, Parameter Processing

Establishes a single location within The Director for processing incoming parameter statements. All interaction with the caller is via the PPE, which the caller must allocate and continue to pass to PARMS.

The Module's structure is as follows:

PRMS1000 Scan for and set Statement Id

- **PRMS1100** Parse along looking for the statement
- **PRMS1200** Validate the statement identifier
- **PRMS1300** Check use, etc.

PRMS2000 Set statement defaults

- **PRMS2100** Application
- **PRMS2120** ACF2
- **PRMS2150** Default
- **PRMS2200** Globals
- **PRMS2250** Group
- **PRMS2300** Profile
- **PRMS2350** Terminal
- **PRMS2400** User
- **PRMS2975** Site
- **PRMS2980** Save or Reload
- **PRMS2990** Logon
- **PRMS2995** Keys

PRMS3000 Parse a parameter operand

- **PRMS3100** Locate the keyword/positional value
- **PRMS3200** Locate the operand
- **PRMS3300** Manipulate the LOGO parameter

PRMS4000 Validate the parameter

- **PRMS4100** Abbreviation logic
- **PRMS4200** Length and use validation
- **PRMS4400** Numeric conversion
- **PRMS4500** DAY Conversion
- **PRMS4600** TIME Conversion
- **PRMS4700** PFKEY Conversion
- **PRMS4800** Handle a statement name (It's positional)
- **PRMS4900** Check out a List characteristic

PRMS5000 Process the parameter operand

TNDPARMX, Common Parsing Routines

Provides common routines for the parameter parsing that is specific to The Director. Typically used for common operands amongst the DEFAULT, GROUP, TERMINAL, and USER statements.

The Module's structure is as follows:

PRMX1000 Authentication

TNDPARM1, Application thru Profile PPE Completion

PARMS just finished parsing a single value for a single operand and would like us to fill in the PPE, as needed.

The Module's structure is as follows:

PRMO1000 Application

PRMO2000 Default

PRMO3000 Globals

PRMO4000 Group

PRMO4575 ATTRIBUTES=

PRMO5000 Profile

PRMO9100 Issue an error message

TNDPARM2, Terminal and User PPE Completion

PARMS finished parsing a single value for a single operand for the TERMINAL and USER statement and would like us to fill in the PPE, as appropriate.

The Module's structure is as follows:

PRMT1000 Terminal

PRMT1100 ACQUIRE=
PRMT1125 REJECT=
PRMT1150 APPLICATIONS=
PRMT1175 AUTOLOGOFF=
PRMT1200 AUTHORIZATION
PRMT1250 COMMANDS=
PRMT1275 CONFIDENTIAL=
PRMT1300 DAY=
PRMT1310 AUTHENTICATION=
PRMT1320 LOGMODE=
PRMT1330 WSF=
PRMT1340 DIM=
PRMT1350 FORMAT-ID=
PRMT1360 IDENTIFICATION=
PRMT1370 STATUS-INTERVAL=
PRMT1380 TRIES=
PRMT1390 RECOVERY=
PRMT1400 EXTENSION=
PRMT1425 SCOPE=
PRMT1450 GROUPS=
PRMT1470 MODE=
PRMT1500 LOGO=
PRMT1550 MESSAGES=
PRMT1575 ATTRIBUTES=
PRMT1600 PFKEYS=
PRMT1650 PROFILE=
PRMT1700 TIMES=
PRMT1800 TIMEOUT=
PRMT1850 USER=
PRMT1860 NETID=
PRMT1870 SUBAREAS=
PRMT1900 USERS=
PRMT1950 ID-AREA=
PRMT1960 CUA=

PRMT2000 User

PRMT2100 APPLICATIONS=
PRMT2110 PSWD-OPTIONS=
PRMT2125 ACQUIRE=
PRMT2150 MAXIMUM=
PRMT2160 AUTOLOGOFF=
PRMT2200 AUTHORIZATION=
PRMT2250 COMMANDS=
PRMT2300 DAYS=

PRMT2310 AUTHENTICATION=
PRMT2400 EXTENSION=
PRMT2410 ID-AREA=
PRMT2450 GROUPS=
PRMT2500 LOGO=
PRMT2525 SCOPE=
PRMT2550 MESSAGES=
PRMT2575 ATTRIBUTES=
PRMT2600 PFKEYS=
PRMT2650 PROFILE=
PRMT2700 TERMINALS=
PRMT2800 TIMES=
PRMT2860 NETID=
PRMT2870 SUBAREAS=
PRMT2900 TIMEOUT=
PRMT2910 FORMAT-ID=
PRMT2920 STATUS-INTERVAL=
PRMT2950 SELECTIONS=
PRMT2975 PASSWORD=
PRMT2980 CUA=

PRMT9990 Issue a message

TNDPARM3, Site through ACF2 Statement Completion

PARMS just finished parsing a single value for a single operand for the SITE statement, etc. and would like us to complete filling in the PPE.

The Module's structure is as follows:

PRMT1000 Site

PRMT2000 Save or Reload

PRMT4000 Show

PRMT5000 Acf2

PRMT6000 Logon

PRMT7000 Keys

PRMT7700 Process Keys Values

PRMT8000 Directory

PRMT9000 Resource

TNDPARM4, Globals PPE Completion

PARMS finished parsing a single value for the GLOBALS statement and we should fill in the PPE appropriately.

TNDPARM5, Display thru Dump PPE Completion

PARMS finished parsing a single value for a single operand for the DISPLAY thru STOP statement and would like us to fill in the PPE, as appropriate.

The Module's structure is as follows:

PRMF3000 Display

PRMF4000 Broadcast

PRMF5000 Cancel

PRMF6500 Hold

PRMF7000 Open

PRMF7100 Dump

PRMF8000 Release

PRMF9000 Simlogon

PRMF9100 Disconnect

PRMF9500 Stop

PRMF9600 Delete

PRMF9990 Issue a message

TNDPATCH, Primary Director Area

Provides the basis for the PDA in CSECT form.

TNDPO, VTAM Program Operator Interface

Accomplishes all interaction with VTAM through the Program Operator interface.

The Module's structure is as follows:

PO1000 PO Receive Operations

PO1100 Handle Internal DFB Operations
PO1200 Place the received command into the LOG

PO2000 Send a command

PO2100 Check for Authorization
PO2200 Put call on chain

TNDPRINT, Message Print Processor

Manages all actual printing of a message onto a 328x type LU. PRINT runs as it's own independent DFB and will terminate when the message has been completely printed.

The Module's structure is as follows:

PRNT1000 Connect to the ANE

PRNT1100 Locate the Printer
PRNT1200 Acquire the device

PRNT2000 Collect the message

PRNT3000 Print the message

PRNT3100 Establish the output buffer size/header
PRNT3200 Fill in the buffer
PRNT3300 Write, then wait for the response
PRNT3500 Allocate a one time Broadcast

TNDPRMS, Configuration Parameter Descriptions

Establishes a single location within The Director for describing all potential input parameters.

TNDPROF, Profile Processor

Provides a mechanism to collect the individual option fields into the ANE.

The Module's structure is as follows:

PROF2000 Establish ANE values

PROF2100 Establish PDE based information
PROF2200 Establish VTAM LU Information
PROF2400 Get UDB and UDB's GDB values into ANE
PROF2500 Put default values into ANE
PROF2600 Get TDB and TDB's GDB values into ANE
PROF2700 Set basic ANE pointers
PROF2800 Locate the Profile name
PROF2900 Set up the ANE for initialization

PROF3000 Set CUA override options

PROF9100 Set display area size

PROF9200 TNDFIND the PDE for this user

PROF9300 TNDISSUE a message

TNDRACF, RACF Interface

Provides the interface between The Director and IBM's RACF security product.

The Module's structure is as follows:

RACF1000 Invoke RACF to validate the password

- **RACF1100** Check for password expiration condition
- **RACF1200** VM/GCS password validation
- **RACF1500** Update the Director with RACF information

RACF2000 Evaluate the return code

RACF3000 Logoff the user

RACF4000 Look up the VM Group name

RACF5000 General initialization tasks

RACF6000 Validate APPLICATION PRIVILEGE

RACF7000 Issue any messages that were sent back

TNDRELD, The Reload Processor

Reloads specified control block images from the External File and establishes a predefined environment for The Director.

The Module's structure is as follows:

- **RELD2100** Globals
- **RELD2200** Default
- **RELD2300** Application
- **RELD2400** Users, Terminals, Sites, or Groups
- **RELD2700** Profiles

RELD8000 Check Reload Authorization

RELD9000 Check for spanned record

RELD9100 Issue a TND0727 message

TNDRELD1, The Reload Processor Extension

Reloads specified control block images into storage from the External File.

The Module's structure is as follows:

RELD1000 Users

- **RELD1100** User Fixed Length Fields
- **RELD1200** User Applications
- **RELD1300** User Terminals
- **RELD1400** User Time or Day
- **RELD1500** User Logo
- **RELD1600** User Subareas
- **RELD1700** User Groups

RELD2000 Terminals

- **RELD2100** Fixed length TERMINALS fields
- **RELD2200** TERMINALS Applications=
- **RELD2300** TERMINALS Users=
- **RELD2400** TERMINALS Date= Time=
- **RELD2500** TERMINALS Logo=
- **RELD2600** TERMINALS Subareas=
- **RELD2700** TERMINALS Groups=

RELD3000 Site

RELD4000 Group

RELD5000 Keys

RELD6000 Resource

RELD9000 Check for spanned record

RELD9100 Issue Message TND0380

TNDREPT, The Display Processor

Performs all processing associated with the Display command (overview, specific, or combined displays).

The Module's structure is as follows:

REPT1000 Display Application=

REPT2000 Display Group=

REPT3000 Display Profile=

REPT4000 Display Terminal=

REPT5000 Display User=

- **REPT5500** Display Saved=

REPT6000 Positional Display Operands

REPT7000 Combined Displays

- **REPT7500** Display Site=

REPT8000 Display Network-element=

- **REPT8100** RACF portion of Display Network-element=
- **REPT8200** Topsecret specific messages
- **REPT8300** System Directory Information

REPT9000 Check Authorization for this user

TNDREPT1, Positional Display Processor

Accomplishes processing associated with the Display command where the Display has only a positional parameter on it.

The Module's structure is as follows:

REPT1000 Display Applications

REPT1050 Display Profiles

REPT2000 Display Groups

- **REPT2500** Display Directory

REPT3000 Display Terminals

REPT3050 Display Counts

REPT4000 Display DFBs

REPT4050 Display Globals

REPT5050 Display Users

REPT6000 Display Messages, Notes, Memos, Broadcasts

REPT6050 Display Storage

- **REPT6100** Display Saved

REPT7000 Display Default

- **REPT7500** Display Ptfs
- **REPT7600** Produce the PTF display line
- **REPT7700** Display Modules
- **REPT7800** Display Exits
- **REPT7900** Display File-io

REPT8000 Display Network-elements

REPT8050 Display Sites

REPT9000 Display Inactive

REPT9050 Display Chains

TNDREPT2, Generalized Reporting Facilities

Functions as an extension of REPT and localizes reporting functions associated with various control blocks and display facilities.

The Module's structure is as follows:

REPT1000 Day and Time

- **REPT1100** Time display
- **REPT1200** Day display

REPT2000 Application Initial-data

- **REPT2500** Commands=

REPT3000 Applications=

- **REPT3500** Application Actions=

REPT4000 Messages=

- **REPT4500** Globals Events=

REPT5000 Pfkeys=

- **REPT5500** Authorization
- **REPT5600** First 5 Authorization values
- **REPT5650** Second group of 5 Authorization values
- **REPT5700** Third set of Authorizations
- **REPT5750** Final set of 5 Authorization values
- **REPT5800** Another final set????
- **REPT5900** Output the Authorization message

REPT6000 Profile=

- **REPT6500** Autologoff=

REPT7000 Terminals=

- **REPT7500** Access block

REPT8000 Selections=

- **REPT8500** Inactive List

REPT9000 Logo=

REPT9500 Users= or Network-elements=

TNDSAVE, The Save Processor

Saves specified control block chain(s) on the External File so that the environment that they represent can be restored at a later stage by using the RELOAD statement.

The Module's structure is as follows:

SAVE1000 Applications

SAVE2000 Users

SAVE3000 Terminals

SAVE4000 Sites

SAVE5000 Groups

SAVE6000 Keys

SAVE6500 Resources

SAVE7000 Profiles

SAVE8000 Check Save authorization

TNDSCAN, Boolean Scan Function

Performs generalized scan operations on data strings.

The Module's structure is as follows:

SCN1000 Scan the input and build a qualifier table

SCN2000 Scan the data to see if it qualifies

SCN3000 Free the qualifier table and reset pointer

SCN9000 Find next space in the data

SCN9100 Find the next non-space in the data

SCN9200 Find the next work element

SCN9300 Find the next function

SCN9500 Scan the actual data string for qualifiers

TNDSCRN, Logical Screen Processing

Places the Title, Broadcast, and Id areas into the output terminal transmissions. Also translates all logical SBA sequences to real ones. On input, the opposite occurs.

The Module's structure is as follows:

SCRN1000 Build a physical output buffer

- **SCRN1100** Translate a single SBA
- **SCRN1110** Translate a 14 bit SBA address
- **SCRN1200** Translate an Attribute
- **SCRN1300** Insert LOGO into Title area
- **SCRN1600** Add Identification Area
- **SCRN1700** Write the terminal buffer
- **SCRN1800** Check for possible symbolics in the LOGO
- **SCRN1900** Add the function key area

SCRN2000 Create logical input buffer

- **SCRN2100** Read the terminal
- **SCRN2200** Translate a single SBA in physical buffer
- **SCRN2250** Translate a single 14 bit SBA order
- **SCRN2300** Locate field in the logical input buffer
- **SCRN2400** Check for Id area modification
- **SCRN2500** Set up Broadcast pointers

SCRN3000 Logoff this network element

SCRN4000 Logon the new network user

SCRN5000 The user just timed out

SCRN6000 Simply clear the screen

SCRN7000 Requeue the device

SCRN8000 Write Structured Field

SCRN9000 Test for Line oriented device

SCRN9100 Try to obtain the device lock

SCRN9200 Issue a Network Director message

SCRN9300 Allocate a terminal buffer

TNDSECTY, Internal Director Password Checking

Provides the internal SECURITY=DIRECTOR password validation against the Access block contents.

The Module's structure is as follows:

SCTY1000 Locate the Access Information Block

- **SCTY1100** Collect the PSWD-OPTIONS

SCTY2000 Check out the provided password

- **SCTY2100** Successful logon

SCTY3000 Setting a new password now

- **SCTY3100** Check for minimum password length
- **SCTY3200** Minimum wait interval
- **SCTY3300** Duplicate password in generations?
- **SCTY3400** NO-PATTERN check for password value
- **SCTY3500** Non-repeating password validation
- **SCTY3600** New password has been set successfully

SCTY4000 User has successfully logged on

SCTY5000 Logon has just failed

SCTY9100 Issue a message

TNDSEL, Application Selection Panel Construction

Manages the device during the Application Selection Panel

The Module's structure is as follows:

SEL0100 Get the name into the SWA

SEL1000 Process incoming Selection panel input

SEL2000 Create a new Application Selection Panel

SEL3000 Send the output panel to the device

SEL3100 Check for Autoselect characteristics

SEL4000 Device timeout expired or user logged off

SEL4100 Logoff the user
SEL4200 Set the device defaults

SEL5000 Selection has been made

SEL6000 Send an error message to the device

SEL7000 Process a command line command

SEL9100 Allocate a one time Broadcast message

SEL9200 Issue a message

TNDSEL1, Selection Input Processor

Evaluates the incoming data stream from a device that had the Application Selection Panel on it.

The Module's structure is as follows:

SEL1000 Check Command line

SEL2000 Check Pfkeys

SEL3000 Modified field method?

- **SEL3300** Check for action characters

SEL4000 Cursor placement

SEL5000 Parse command line

- **SEL5100** Check for application name entry
- **SEL5200** Simulated Pfkey used?
- **SEL5300** Check for an internal command

SEL6000 Process queued command element

SEL7000 Check for authorized commands

SEL8000 Check Application Maximum and Concurrent

TNDSEND, Message Send Action

Initiate the transmission of a message from the origin to the destination.

The Module's structure is as follows:

SEND1000 Update the Originator"s MIX

- **SEND1100** Get rid of the In Process MDE

SEND2000 Update the destination(s) MIX(es)

SEND4000 Check file maintenance wake up time

SEND5000 Notify Active Network Elements

- **SEND5100** Set up Notify text message

SEND6000 Check on the Site specification

SEND7000 Send it to another Site

SEND8000 Figure expiration date and time

SEND9100 Update a MIX

TNDSHADB, Show Application Processor

Provides a mechanism for the full screen manipulation of the Application Definition Block.

The Module's structure is as follows:

SADB0100 Set defaults for a new ADB

SADB1000 Display the existing definition

SADB1100 Second level fields

SADB2000 Write, then read the device

SADB3000 Check for primary commands

SADB3100 Down
SADB3150 Up
SADB3200 End
SADB3300 Help
SADB3400 Profile
SADB3500 Quit or Cancel

SADB4000 Edit the individual fields

SADB4200 Validate the numeric fields
SADB4400 Validate the secondary fields
SADB4500 Check for duplicate definitions

SADB9100 Issue an error message

TNDSHAIB, Access Information Block Manipulation

Provides a mechanism for the full screen manipulation of the Access Information Block

The Module's structure is as follows:

SAIB1000 Display the current definition

- **SAIB1100** Fill in the algorithm/device name
- **SAIB1200** Optional parameters and comments

SAIB2000 Write, then read the device

SAIB3000 Check for primary commands

- **SAIB3100** Down
- **SAIB3150** Up
- **SAIB3200** End
- **SAIB3300** Help
- **SAIB3400** Profile
- **SAIB3500** Quit or Cancel

SAIB4000 Edit the individual fields

- **SAIB4100** Edit the device/algorithm field
- **SAIB4200** Collect the parameters
- **SAIB4300** Collect the comments (if around)

TNDSHANE, Show ANE Processor

Provides a mechanism for the full screen display and manipulation of the fields within the ANE

The Module's structure is as follows:

SANE1000 Display the current definition

SANE2000 Write, then read the device

SANE3000 Check for primary commands

- **SANE3100** Down
- **SANE3200** End
- **SANE3300** Help
- **SANE3400** Profile
- **SANE3500** Quit or Cancel
- **SANE3600** Color

SANE4000 Edit the individual input fields

SANE9100 Issue an error message

TNDSHBIT, Show Bit Field Sub-processor

Provides a generalized routine for the full screen collection of various bit settings amongst control blocks

The Module's structure is as follows:

SBIT1000 Display the existing field values

- **SBIT1100** User or Terminal Attributes
- **SBIT1200** Pswd-Options
- **SBIT1300** Commands
- **SBIT1400** Messages
- **SBIT1500** Default Attributes

SBIT3000 Replace the bits we just set

SBIT4000 Edit the input buffer

- **SBIT4100** Attributes
- **SBIT4200** Pswd-Options
- **SBIT4300** Commands
- **SBIT4400** Messages
- **SBIT4500** Default Attributes

SBIT7000 Write, then read the device

SBIT8000 Check for primary commands

- **SBIT8100** Down
- **SBIT8200** End
- **SBIT8300** Help
- **SBIT8400** Profile
- **SBIT8500** Quit or Cancel
- **SBIT8600** Up

SBIT9100 Issue a message

TNDSHCMP, Show Copy and Move Processor

Provides a mechanism within Show to re-arrange the control block chain.

The Module's structure is as follows:

SCMP1000 Set up the pointer location

SCMP2000 Invoke the appropriate function

- **SCMP2100** Copy ADB
- **SCMP2200** Copy AIB
- **SCMP2300** Copy Directory
- **SCMP2400** Copy GDB
- **SCMP2500** Copy KDB
- **SCMP2600** Copy NDL
- **SCMP2700** Copy PDE
- **SCMP2800** Copy SDB
- **SCMP2900** Copy TDB

SCMP3000 Copy UDB

- **SCMP3100** Copy Resource

SCMP4000 Move the control block

SCMP9000 Build ALB chain

SCMP9100 Build AAI chain

SCMP9200 Build AIE chain

SCMP9300 Build IDE chain

SCMP9400 Build NEL, ATE, or AUE chains

SCMP9500 Copy some miscellaneous block of R3 length

SCMP9600 Copy a CB block of R3 length

SCMP9700 Build AAE chain

TNDSHCOL, Show Chains Sub-processor

Provides a generalized routine for the full screen collection of various second level control block chains.

The Module's structure is as follows:

SCOL1000 Display the existing field values

- **SCOL1100** Applications and Pfkey values
- **SCOL1200** Logo and Resource
- **SCOL1300** Terminals, Users, or Network elements
- **SCOL1400** Authorization
- **SCOL1500** Time and Days
- **SCOL1600** Actions
- **SCOL1700** Initial-data
- **SCOL1800** Subareas
- **SCOL1900** Rotate and Balance

SCOL2000 Move control block chain to look aside area

SCOL3000 Return look aside area back to CB chain

SCOL4000 Edit the input buffer

- **SCOL4100** Applications and Pfkey values
- **SCOL4200** Logo
- **SCOL4300** Terminals/Users/Network-elements/Privilege
- **SCOL4400** Authorization
- **SCOL4500** Times and Days
- **SCOL4600** Actions
- **SCOL4700** Initial-data
- **SCOL4800** Subareas
- **SCOL4900** Rotate and Balance input values

SCOL7000 Write, then read the device

SCOL8000 Check for primary commands

- **SCOL8100** Down
- **SCOL8200** End
- **SCOL8300** Help
- **SCOL8400** Profile
- **SCOL8500** Quit or Cancel
- **SCOL8600** Up

SCOL9100 Issue a message

TNDSHC01, Show Labels Sub-processor

Provides a mechanism for the full screen collection of the Header and Trailer label control blocks

The Module's structure is as follows:

SC011000 Display the current settings

SC012000 Edit the input

SC013000 Write, then read the device

SC018000 Check for primary commands

- **SC018100** Down
- **SC018200** End
- **SC018300** Help
- **SC018400** Profile
- **SC018500** Quit or Cancel

TNDSHDEF, Show Defaults Processor

Provides a mechanism for the full screen manipulation of the Default definitions.

The Module's structure is as follows:

SDEF1000 Display the current settings

- **SDEF1100** Second level fields

SDEF2000 Write, then read the device

SDEF3000 Check for primary commands

- **SDEF3100** Down
- **SDEF3200** End
- **SDEF3300** Help
- **SDEF3400** Profile
- **SDEF3500** Quit or Cancel

SDEF4000 Edit the individual fields

- **SDEF4100** Edit Profile Authorization
- **SDEF4200** Validate numeric fields
- **SDEF4300** Edit the other primary fields
- **SDEF4400** Validate the secondary fields

SDEF9100 Issue an error message

TNDSHDIR, Show Directory Processor

Provides a mechanism for the full screen manipulation of the System Directory.

The Module's structure is as follows:

SDIR0100 Set defaults for a new DIR

SDIR1000 Display the current settings

SDIR2000 Write, then read the device

SDIR3000 Check for primary commands

SDIR3100 Down
SDIR3150 Up
SDIR3200 End
SDIR3300 Help
SDIR3400 Profile
SDIR3500 Quit or Cancel

SDIR4000 Edit the individual fields

TNDSHDMT, Show Director Message Text

Provides the mechanism to manipulate the Director Message Text elements in a full screen manner

The Module's structure is as follows:

SDMT0100 Set default for a new DMT

SDMT1000 Display the current field values

SDMT2000 Write, then read the device

SDMT3000 Check for primary commands

SDMT3100 Down
SDMT3150 Up
SDMT3200 End
SDMT3300 Help
SDMT3400 Profile
SDMT3500 Quit or Cancel

SDMT4000 Edit the individual fields

TNDSHEXT, Show Exits Processor

Provides a manner to manage the user exits from a full screen panel

The Module's structure is as follows:

SEXT1000 Display the current field values

SEXT2000 Write, then read the device

SEXT3000 Check for primary commands

- **SEXT3100** Down
- **SEXT3200** End
- **SEXT3300** Help
- **SEXT3400** Profile
- **SEXT3500** Quit or Cancel

SEXT4000 Edit the individual fields

SEXT9100 Issue an error message

TNDSHGDB, Show Group Processor

Provides a mechanism to manipulate the GDB (Group) definition(s)

The Module's structure is as follows:

- **SGDB0100** Set up default values for a new GDB

SGDB1000 Display the current field values

- **SGDB1100** Second level fields

SGDB2000 Write, then read the device

SGDB3000 Check for primary commands

- **SGDB3100** Down
- **SGDB3150** Up
- **SGDB3200** End
- **SGDB3300** Help
- **SGDB3400** Profile
- **SGDB3500** Quit or Cancel

SGDB4000 Edit the individual fields

- **SGDB4100** Edit the Profile authorization
- **SGDB4200** Validate the numeric fields
- **SGDB4300** Edit the other primary fields
- **SGDB4400** Validate the secondary fields
- **SGDB4500** Check for duplicate definitions

SGDB9100 Issue an error message

TNDSHGLO, Show Globals Processor

Provides a manner to manipulate the Globals definitions online and interactively.

The Module's structure is as follows:

SGLO1000 Display the current values

- **SGLO1100** The first panel
- **SGLO1200** Second panel fields

SGLO2000 Write, then read the device

SGLO3000 Check for primary commands

- **SGLO3100** Down
- **SGLO3200** End
- **SGLO3300** Help
- **SGLO3400** Profile
- **SGLO3500** Quit or Cancel

SGLO4000 Edit the individual fields

- **SGLO4100** Panel one, column one
- **SGLO4200** Panel two, edit second column
- **SGLO4400** Edit Globals authorization
- **SGLO4500** The second panel

SGLO9100 Edit a STORAGE-POOL value

TNDSHGL2, Show Globals Field Sub-processor

Provides a generalized routine for the full screen collection of the GLOBALS multiple occurring fields.

The Module's structure is as follows:

SHGL0100 Obtain work areas

SHGL1000 Display the existing field values

- **SHGL1100** COLORS
- **SHGL1200** EVENTS
- **SHGL1300** NETWORK-WAITS

SHGL3000 Replace the bits we just set

SHGL4000 Edit the input buffer

- **SHGL4100** COLORS
- **SHGL4200** EVENTS
- **SHGL4300** NETWORK-WAITS

SHGL7000 Write, then read the device

SHGL8000 Check for primary commands

- **SHGL8100** Down
- **SHGL8200** End
- **SHGL8300** Help
- **SHGL8400** Profile
- **SHGL8500** Quit or Cancel
- **SHGL8600** Up

SHGL9100 Issue a message

TNDSHKDB, Show Keys Processor

Provides a manner to manipulate the Keys Definition Block

The Module's structure is as follows:

SKDB0100 Set defaults for a new KDB

SKDB1000 Display the current field values

SKDB2000 Write, then read the device

SKDB3000 Check for primary commands

SKDB3100 Down
SKDB3200 End
SKDB3300 Help
SKDB3400 Profile
SKDB3500 Quit or Cancel

SKDB4000 Edit the individual fields

SKDB4500 Check for duplicate definitions

SKDB5000 Display Key values

SKDB6000 Edit Key values

TNDSHNDL, Show List Processor

Provides a manner to manipulate the Network Distribution Lists used within The Director.

The Module's structure is as follows:

SNDL0100 Set defaults for a new NDL

SNDL1000 Display the current field settings

SNDL1100 Second level fields

SNDL2000 Write, then read the device

SNDL3000 Check for primary commands

SNDL3100 Down
SNDL3150 Up
SNDL3200 End
SNDL3300 Help
SNDL3400 Profile
SNDL3500 Quit or Cancel

SNDL4000 Edit individual fields

SNDL4400 Validate secondary fields

SNDL5000 Initialization

TNDSHOP, Show Operands Processor

Provides a mechanism for the full screen manipulation of common definition operands

The Module's structure is as follows:

SHOP1000 Interpret a field

- **SHOP1100** Commands=
- **SHOP1120** Interpret a numeric value
- **SHOP1140** Selections=
- **SHOP1160** Messages=
- **SHOP1180** Autologoff=
- **SHOP1200** Authentication=
- **SHOP1300** Events=
- **SHOP1400** Format-id=

SHOP5000 Edit an input field

- **SHOP5100** Edit a numeric value
- **SHOP5120** Selections=
- **SHOP5140** User Id-area=
- **SHOP5160** Authentication=
- **SHOP5180** Autologoff=
- **SHOP5200** Commands=
- **SHOP5220** Messages=
- **SHOP5240** Terminal Id-area=
- **SHOP5300** Events=
- **SHOP5400** Format-id=

TNDSHOW, Full Screen Control Block Manipulation

Provides a generalized mechanism for the full screen manipulation of the various Director control block chains.

The Module's structure is as follows:

SHOW0100 Check Show authorization

SHOW1000 Locate the address

SHOW2000 Build and write the output panel

SHOW3000 Obtain the input "command"

- **SHOW3100** Parse the input command
- **SHOW3200** Up
- **SHOW3400** End or Quit
- **SHOW3500** Help
- **SHOW3600** Add
- **SHOW3700** Select, Locate, or Find
- **SHOW3800** Top
- **SHOW3900** Bottom

SHOW4000 Reset the SSE"s

SHOW5000 Check for and process the action characters

SHOW6000 Process non-list type items

SHOW8000 Build the Show list of items

SHOW9100 Call the detail process for selected item

SHOW9200 Issue a message

SHOW9300 Free some TP storage

SHOW9400 Allocate a terminal buffer

SHOW9500 Find a non blank character

TNDSHOW2, Show Primary Command Processor

Contains several of Show's primary commands and the logical process associated with them.

The Module's structure is as follows:

SHOW1000 Select, Locate, or Find

- **SHOW1100** Set up the comparison strings
- **SHOW1200** Compare this block

TNDSHOW3, Show Prefix Commands

Operates as an extension to Show and processes individual control block prefix commands.

The Module's structure is as follows:

SHOW5000 Check for and process action characters

- **SHOW5100** Delete
- **SHOW5200** Hold
- **SHOW5300** Release
- **SHOW5400** Dump
- **SHOW5500** Insert
- **SHOW5600** Set the next spot to work with
- **SHOW5700** Mark the Before or After location
- **SHOW5800** Set up the Copy or Move information
- **SHOW5900** Replicate a control block

SHOW6000 Check Copy or Move requests

SHOW9100 Call the detail process for selected item

SHOW9200 Issue a message

TNDSHOW4, Show Panel Display

Contains the logic that constructs the actual SHOW menu and selection list.

The Module's structure is as follows:

SHOW1000 Fill in the menu header

SHOW2000 Format a single panel line

- **SHOW2100** Network Element (ANE) processing
- **SHOW2200** Directory entry processing
- **SHOW2300** Abend processing

SHOW3000 Insert additional info into panel line

- **SHOW3100** Access Information Block
- **SHOW3200** Terminal Definition Block
- **SHOW3300** User Definition Block
- **SHOW3400** Group Definition Block

SHOW4000 Write the output panel

TNDSHPDE, Show Profiles Processor

Provides a mechanism for the full screen manipulation of the Profile Data Element

The Module's structure is as follows:

SPDE0100 Set defaults for a new PDE
SPDE0200 Check authorization

SPDE1000 Display the current field values

SPDE1200 Optional Parameters

SPDE2000 Write, then read the device

SPDE3000 Check for primary commands

SPDE3100 Down
SPDE3150 Up
SPDE3200 End
SPDE3300 Help
SPDE3500 Quit or Cancel
SPDE3600 Reset

SPDE4000 Edit the individual fields

SPDE4200 Collect the optional Parameters
SPDE4500 Check for duplicate definitions

TNDSHRES, Show Resource Processor

Provides a manner to manipulate the Resource definitions interactively.

The Module's structure is as follows:

SRES1000 Display the current field values

SRES2000 Write, then read the device

SRES3000 Check for primary commands

SRES3100 Down
SRES3200 End
SRES3300 Help
SRES3400 Profile
SRES3500 Quit or Cancel

SRES4000 Edit the individual fields

TNDSHSDB, Show Sites Processor

Provides a mechanism to manipulate the Site Definition Block interactively.

The Module's structure is as follows:

SSDB0100 Set the defaults for a new SDB

SSDB1000 Display the current field values

SSDB1100 Second level fields

SSDB2000 Write, then read the device

SSDB3000 Check for primary commands

SSDB3100 Down
SSDB3150 Up
SSDB3200 End
SSDB3300 Help
SSDB3400 Profile
SSDB3500 Quit or Cancel

SSDB4000 Edit the individual fields

SSDB4400 Validate the secondary fields
SSDB4500 Check for duplicate definitions

TNDSHSEL, CUA Selection Processor

Provides a manner to manipulate the selection panel in CUA mode.

The Module's structure is as follows:

SSEL1000 Build the selection panel

SSEL1100 Set reserve area
SSEL1200 Insert SSE information into buffer
SSEL1300 Create CUA Identification panel

SSEL2000 Process the input from the device

SSEL2100 Logoff the user from Function Key action

TNDSHSTA, Show Statistics Processor

Displays statistical information about TND's processing

The Module's structure is as follows:

SSTA1000 Initialization

SSTA2000 Generate the output panel

- **SSTA2100** General information
- **SSTA2200** GSA Chain Information
- **SSTA2300** Format the interval counters
- **SSTA2400** Application counters
- **SSTA2500** Format Storage Counts
- **SSTA2600** The total count formatting process
- **SSTA2700** Set the color for the data fields

SSTA3000 Write, then read the device

SSTA4000 Check for primary commands

- **SSTA4100** Down
- **SSTA4200** Up
- **SSTA4300** Help
- **SSTA4400** Profile
- **SSTA4500** Quit, Cancel, End

SSTA9300 Adjust and compute 1024 increments

TNDSHSTO, Show Storage Processor

Displays information about the status of TND's storage pool

The Module's structure is as follows:

SSTO1000 Summarize the current pools

- **SSTO1100** Add up a single chain"s information
- **SSTO1200** Format a single line

SSTO2000 Write, then read the device

SSTO3000 Check for primary commands

- **SSTO3100** Down
- **SSTO3200** Up
- **SSTO3300** Help
- **SSTO3400** Profile
- **SSTO3500** Quit, Cancel, End

SSTO4000 Edit the individual fields

- **SSTO4100** Check input from Summary Panel
- **SSTO4300** Scan SSE chain for selected element

SSTO5000 Map a single chain

- **SSTO5100** Format most of a single line
- **SSTO5200** Count the elements in a single GSA

SSTO6000 Map a single GSA pool element (MX or TP)

- **SSTO6100** Format a single line for a GSA entry

SSTO9100 Allocate an SSE to keep track of choice

SSTO9200 Allocate a new I/O buffer

SSTO9300 Free the SSE chain

TNDSHTDB, Show Terminals Processor

Provides a mechanism to manipulate the Terminal Definition Block interactively and via a full screen process.

The Module's structure is as follows:

STDB0100 Set the defaults for a new TDB

STDB1000 Display the current field values

STDB1100 Second level fields

STDB2000 Write, then read the device

STDB3000 Check for primary commands

STDB3100 Down
STDB3150 Up
STDB3200 End
STDB3300 Help
STDB3400 Profile
STDB3500 Quit or Cancel

STDB4000 Edit individual fields

STDB4100 Edit the Profile authorization
STDB4200 Validate the numeric fields
STDB4300 Edit the other primary fields
STDB4400 Validate the secondary fields
STDB4500 Check for duplicate definitions

STDB9100 Issue an error message

TNDSHTRA, Show Trace Processor

Provides a manner to manipulate the Trace criteria from a full screen panel

The Module's structure is as follows:

STRA1000 Display the current settings

STRA2000 Write, then read the device

STRA3000 Check for primary commands

- **STRA3100** Down
- **STRA3200** End
- **STRA3300** Help
- **STRA3400** Profile
- **STRA3500** Quit or Cancel

STRA4000 Edit the individual fields

STRA5000 Activate or inactivate the Trace facility

STRA9100 Issue an error message

TNDSHUDB, Show Users Processor

Provides the manner to manipulate the contents of the User Definition Block in a full screen mode

The Module's structure is as follows:

- **SUDB0100** Set defaults for a new UDB

SUDB1000 Display the current field values

- **SUDB1100** Second level fields
- **SUDB1200** Set up SHOP interpreted fields

SUDB2000 Write, then read the device

SUDB3000 Check for primary commands

- **SUDB3100** Down
- **SUDB3150** Up
- **SUDB3200** End
- **SUDB3300** Help
- **SUDB3400** Profile
- **SUDB3500** Quit or Cancel

SUDB4000 Edit the individual fields

- **SUDB4100** Edit the Profile authorization
- **SUDB4200** Validate numeric fields
- **SUDB4300** Edit the other primary fields
- **SUDB4400** Validate the secondary fields

SUDB4500 Check for duplicate definitions

SUDB9100 Issue an error message

TNDSMSG, CP SMSG Services

Provides an interface for the VM or CP SMSG or MSG facility

The Module's structure is as follows:

SMSG1000 Initialize SMSG communications

SMSG1100 Set up the ANE for the OPERSMSG
SMSG1200 Process external interrupts

SMSG2000 Obtain input from CP via IUCV

SMSG2100 Try to connect via the ANE
SMSG2200 Try to connect via the Terminal
SMSG2300 Wait for Post or process another element
SMSG2400 Attach a DFB to process this NSI request

SMSG3000 Parse the input

SMSG4000 Process the request

SMSG4100 Invoke the VM interface
SMSG4200 Invoke the Program Operator
SMSG4300 Save Command
SMSG4400 Display Command
SMSG4500 Invoke operator services
SMSG4600 Invoke Deck manipulation
SMSG4700 Dump Storage
SMSG4800 ACF2 Command
SMSG4900 Reload Command

SMSG5000 Reset IUCV

SMSG6000 Handle VMSECURE incoming message

SMSG6100 Place VMSECURE response into the log
SMSG6200 Handle TNDVMS responses

TNDSNK, SecureNet Key Authentication Interface

Provides the process to validate a user entering the system against Digital Pathways SecureNet Key Authentication device.

The Module's structure is as follows:

SNK1000 Installkey

- **SNK1100** Convert seed to octal

SNK2000 Buildsnk

- **SNK2100** Random
- **SNK2200** Desencrypt

SNK3000 Prompt the operator

SNK4000 Generate a random key

SNK8000 Pbits

SNK9000 Rotl

TNDSSI, SSI Data Area Formattor

Acquires and formats the INITIAL-DATA area that is passed to the selected APPLICATION.

The Module's structure is as follows:

SSI2000 Handle the IDE variables

SSI3000 Build the SSI buffer

SSI4000 Build the SSX buffer

- **SSI4100** Set my own site name (from GLOBALS SITE=)

TNDSTAT, System Measurement (SMF) Statistics Generation

Generates the appropriate external record information when the SMF= parameter has been turned on and the EVENTS operand has been activated.

The Module's structure is as follows:

STAT1000 Obtain a SMR

STAT2000 Fill in the SMR values

- **STAT2100** Set up 1.3.0 and beyond SMR header
- **STAT2200** Return Event
- **STAT2300** Logon Event
- **STAT2400** Logoff
- **STAT2500** Applstat
- **STAT2700** Infoupd
- **STAT2800** Applcnts
- **STAT2900** Msgsend
- **STAT3100** Msgdel
- **STAT3200** Msgprint
- **STAT3300** Msgview
- **STAT3400** Admincmd
- **STAT3500** Select

STAT6000 Write the SMR to SMF

TNDSTAT2, TNDSAR Statistics Generation

Generates the appropriate external record information when the ACCOUNTING=TNDSAR has been turned on.

The Module's structure is as follows:

STAT1000 Obtain a SAR

STAT2000 Fill in the SAR values

- **STAT2100** Set up the SAR Header for file writes
- **STAT2200** Return
- **STAT2300** Logon
- **STAT2400** Logoff
- **STAT2500** Applstat
- **STAT2600** Vtamerrs
- **STAT2700** Infoupd
- **STAT2800** Applcnts
- **STAT2900** Msgsend
- **STAT3100** Msgdel
- **STAT3200** Msgprint
- **STAT3300** Msgview
- **STAT3400** Admincmd
- **STAT3500** Select

STAT5000 Open the output file

STAT6000 Write the SAR to the external medium

TNDSTAT3, SAR Statistics Generation

Generates the proper external record information when the SMF= parameter and ACCOUNT=SAR has been turned on.

The Module's structure is as follows:

STAT1000 Obtain a SAR

STAT2000 Fill in the SAR values

- **STAT2100** Set up the SAR header
- **STAT2200** Return
- **STAT2300** Logon
- **STAT2400** Logoff
- **STAT2500** Applstat
- **STAT2600** Vtamerrs
- **STAT2700** Infoupd
- **STAT2800** Applcnts
- **STAT2900** Msgsend
- **STAT3100** Msgdel
- **STAT3200** Msgprint
- **STAT3300** Msgview
- **STAT3400** Admincmd
- **STAT3500** Select

STAT6000 Write the SAR to SMF

TNDSTCK, Store Clock Processor

Computes the current time and date from the value in the CPU's Time of Day clock (accessed via the STCK instruction)

The Module's structure is as follows:

STCK1000 Compute the number of years

STCK2000 Compute the month of the year

STCK3000 Compute the day of the month

STCK4000 Derive the hours, minutes, and seconds time

STCK5000 Format the derived info for our caller

TNDSTOR, Storage Management

Manages the dynamic storage pools acquire during execution.

The Module's structure is as follows:

STOR1000 Allocate storage

- **STOR1100** Control Block pool allocation
- **STOR1200** MiXed pool allocation
- **STOR1300** TeleProcessing allocation
- **STOR1400** Set the storage framing indicators

STOR2000 Recover space

STOR3000 Wait for storage

STOR4000 Abnormal termination for storage shortage

STOR5000 Storage release

- **STOR5100** Release CB pool storage
- **STOR5200** Release MX pool storage
- **STOR5300** Release TP pool storage

STOR9000 Reset GSA fields after storage allocation

STOR9100 Getmain storage from the operating system

STOR9200 Issue a message

STOR9300 Validate the byte map

TNDTIMER, Timing Services

Provides general timing services to the other DFB related tasks within the Director.

The Module's structure is as follows:

TIME2000 Compute the Stimer interval required

TIME3000 Issue the Timed Wait

TNDTRACE, Internal Trace Routine

Accomplishes internal module trace operations, as requested

The Module's structure is as follows:

TRA1000 Determine if we should trace current item

TRA2000 Check and build the alternate vector list

TRA3000 Reset the real vector with alternate area

TRA9000 Pass control to original routine requested

TNDTSS, TopSecret Interface

Provides the interface between the Director and the TopSecret security package.

The Module's structure is as follows:

TSS1000 Invoke TopSecret to validate the password

- **TSS1100** Prepare the RACROUTE Arguments
- **TSS1300** TopSecret/VM Password validation
- **TSS1400** Get miscellaneous TopSecret/VM information
- **TSS1500** Update the Directory with TopSecret info
- **TSS1600** Parse TopSecret/MVS Message Buffer

TSS2000 Evaluate return code

- **TSS2100** Format TopSecret/VM Messages

TSS3000 Logoff the user

- **TSS3300** TopSecret/VM Logoff

TSS4000 Look up the VM Group name

TSS6000 Validate the APPLICATION Privilege

- **TSS6100** Check out TopSecret/VM Privilege
- **TSS6300** Check out TopSecret/MVS Privilege

TNDVAR, Generalized Variable Support

Provides a mechanism to replace symbolic variables in a variety of locations (messages, Info panels, LOGOs, etc.) with execution time values.

The Module's structure is as follows:

VAR1000 Load the Control Blocks into the WAE

VAR2000 Load more WAE Control Block addresses

VAR3000 Locate a variable

- **VAR3100** Search the variable table for the entry
- **VAR3200** Process the variable"s VAR entry

VAR4000 Insert symbolic value

- **VAR4100** Alphanumeric processing
- **VAR4200** Format a packed date or time field
- **VAR4300** Halfword or Fullword processing
- **VAR4400** Hexadecimal processing
- **VAR4500** STCK type conversion
- **VAR4600** Isolate type specification
- **VAR4700** Binary fullword to milliseconds
- **VAR4800** Binary fullword as a decimal value
- **VAR4900** Adjust string for type specification length

VAR5000 Initialization

VAR6000 Special variable handling routines

- **VAR6100** CONNECT
- **VAR6200** PASSWORD
- **VAR6300** OPERANDS
- **VAR6400** INHERIT
- **VAR6500** SFPROTECT
- **VAR6600** SFDARK
- **VAR6700** LOGMODE
- **VAR6800** NEWS
- **VAR6900** DIRSTAT
- **VAR7100** ANESTAT
- **VAR7200** MORE
- **VAR7300** CUAOPTn
- **VAR7400** CUATITLE
- **VAR7500** PROFILE PARMS
- **VAR7600** TICKET
- **VAR7700** DATEC

VAR9100 Suppress leading blanks

TNDVM, VM Command (Diagnose) Interface

Accomplishes all interaction with CP through the DIAGNOSE X'08' interface.

The Module's structure is as follows:

VM1000 Check to see if we can Diagnose

- **VM1100** Command authorization check

VM2000 Set up the RESET command

VM3000 Get set to issue a command

VM4000 Issue the Diagnose

VM5000 Parse reply buffer and put it in the LOG

- **VM5100** Check for TopSecret/VM Message Information

VM6000 Check the Directory for Userid/Password ok

TNDVMSI, VM/Secure Interface

Provides the interface between The Director and the VM/Secure security package.

The Module's structure is as follows:

VMSI1000 Issue Diagnose A0 to validate password

- **VMSI1100** TNDVMS GETINFO from VM/Secure
- **VMSI1200** Change the password
- **VMSI1300** Check out the password warning interval

VMSI3000 Logoff the user

VMSI4000 Look up the VM Group name

VMSI5000 General initialization tasks

VMSI6000 Establish the VM/Secure version id

VMSI9100 Acquire the VM/Secure lock

VMSI9200 Make a Date into a magic number for compare

TNDVSAM, VSAM Interface

Provides a single location within The Director to manage all real invocations of the operating system's file access method.

The Module's structure is as follows:

VSAM1000 Locate an RPL

VSAM2000 Generate a control block

- **VSAM2100** Generate an ACB
- **VSAM2200** Generate an RPL

VSAM3000 Execute the request

- **VSAM3100** Open the ACB
- **VSAM3200** Close the ACB
- **VSAM3300** Get a record
- **VSAM3400** Update a record
- **VSAM3500** Erase a record
- **VSAM3600** Add a record

VSAM4000 Interpret the feedback codes

- **VSAM4100** Open error interpretation
- **VSAM4200** Close error interpretation
- **VSAM4300** Control block error interpretation
- **VSAM4400** Request error interpretation
- **VSAM4500** What macro, EFR experienced the error?

VSAM5000 Test for VSAM Close and issue Check macro

TNDVSI, VM Network System Interface

Provides a standard manner with which an application program may communicate with The Director from VM/CMS.

The Module's structure is as follows:

NSI1000 Validate the request

- **NSI1100** Validate Send Message request

NSI2000 Establish connection with The Director

NSI3000 Ship the NSI request

NSI4000 Terminate the session

TNDVTAM, VTAM Interface

Provides a generalized interface between The Director's internal functions and the VTAM Application Program Interface (API).

The Module's structure is as follows:

VTAM1000 Retry the operation

VTAM2000 Generate the control blocks

- **VTAM2100** Generate an ACB
- **VTAM2200** Generate an RPL
- **VTAM2300** Generate a NIB

VTAM3000 Execute the request

- **VTAM3100** Open ACB
- **VTAM3150** Close ACB
- **VTAM3200** Setlogon
- **VTAM3250** Reqsess
- **VTAM3300** Inquire
- **VTAM3400** Opndst
- **VTAM3500** Simlogon
- **VTAM3600** Clsdst
- **VTAM3700** Sendcmd
- **VTAM3720** Opnsec
- **VTAM3750** Rcvcmd
- **VTAM3770** Termsess
- **VTAM3780** Resetsr CA
- **VTAM3800** Send
- **VTAM3900** Receive

VTAM4000 Interpret Feedback Codes

VTAM8000 LU 6.2 Function?

VTAM9100 Wait for RPLECB or a time interval

VTAM9300 Check the RPL

VTAM9400 Wait for RPLECB to be posted

VTAM9500 SHOWCB RPL Contents

VTAM9900 Issue the message in R2

TNDVTAMX, VTAM Asynchronous Exits

Contains all the asynchronous routines ued by The Director for interaction with VTAM. This is logically an extension of the TNDVTAM routine, who sets this in place via EXLST.

The Module's structure is as follows:

VTMX1000 ATTN exit

VTMX5000 TPEND

    VTMX5500 RPL Exit Routine

VTMX6000 Release Request

VTMX7000 LOGON Exit

    VTMX7100 Scan CINIT area for LOGMODE name, etc.

VTMX8000 LOSTERM

VTMX9000 Network Services Exit

VTMX9500 SCIP Exit

TNDVTAM1, VTAM Send Interface

Provides the generalized manner with which The Director writes device output via the VTAM API.

The Module's structure is as follows:

VTAM3000 Send an output buffer

    VTAM3800 Send

VTAM9100 Wait for RPLECB or a time interval

VTAM9200 EXECRPL

VTAM9300 Check the RPL

VTAM9400 Wait for RPLECB to be posted

TNDVTAM2, VTAM Error Interpretation

Interprets any error that may have occurred during a VTAM API based operation.

The Module's structure is as follows:

VTAM1000 Interpret sense fields

- **VTAM1100** Extended sense code descriptions

VTAM4000 Interpret feedback codes

- **VTAM4100** Open errors
- **VTAM4200** Close errors
- **VTAM4300** Manipulative macro errors
- **VTAM4400** Request errors

VTAM5000 APPC based error processing

VTAM9100 Locate the LU name

TNDWCMD, Action Bar Command Processor

Processes the command functions that are based upon the CUA Action Bar.

The Module's structure is as follows:

WCMD1000 Which window should we present?

- **WCMD1100** Present the available commands
- **WCMD1200** Present the Logon panel

WCMD2000 Write, then read the device

WCMD3000 Check for primary commands

- **WCMD3100** Down
- **WCMD3200** End
- **WCMD3300** Help
- **WCMD3400** Profile
- **WCMD3500** Quit or Cancel

WCMD4000 Edit the individual fields

- **WCMD4100** Edit command window
- **WCMD4200** Process Logon fields

WCMD9000 Add a subfield to the command window

WCMD9100 Issue an error message

TNDWHLP, Action Bar Help Processor

Creates the appropriate Help information for presentation to a CUA based user.

The Module's structure is as follows:

WHLP1000 Build the Help information panels

- **WHLP1100** Build the Action Bar Help selection window
- **WHLP1200** Build the detail Help information window

WHLP2000 Write, then read the device

WHLP3000 Check for primary commands

- **WHLP3100** End
- **WHLP3200** Cancel
- **WHLP3300** Down
- **WHLP3400** Up
- **WHLP3500** Help

WHLP4000 Edit the individual fields

- **WHLP4100** Edit the Action Bar Help selection request

TNDWLOG, Action Bar Logon/Logoff Prompt

Provides the CUA logon and logoff activity.

The Module's structure is as follows:

WLOG1000 Present the Logon window

WLOG2000 Write, then read the device

WLOG3000 Check for primary commands

- **WLOG3100** Down
- **WLOG3200** End
- **WLOG3300** Help
- **WLOG3400** Profile
- **WLOG3500** Quit or Cancel

WLOG4000 Edit the input fields

WLOG9100 Issue an error message

TNDWMSG, Window Message Processor

Provides the ability to present Network Director messages in a CUA pop up window.

The Module's structure is as follows:

WMSG1000 Display the message in a pop up window

WMSG1100 Relocate the message to a screen area

WMSG2000 Write, then read the device

WMSG9000 Relocate message to screen area

TNDWOPT, Options Action Bar Command

Allows a CUA user to manipulate the contents of the Options settings.

The Module's structure is as follows:

WOPT1000 Present the Options pull down

WOPT2000 Write, the read the device

WOPT3000 Check for primary commands

WOPT3100 Down
WOPT3200 End
WOPT3300 Help
WOPT3400 Profile
WOPT3500 Quit or Cancel

WOPT4000 Edit the input field values

WOPT5000 Generate a new PDE

WOPT9100 Issue an error message

TNDXFER, Selection Transfer Processor

Transfers ownership of the device to the selected APPLICATION.

The Module's structure is as follows:

XFER1000 Assign actual ADB, if ROTATE type picked

XFER2000 Selection has been made

- **XFER2100** Set up the SSI data
- **XFER2200** Internal APPLICATION has been selected
- **XFER2300** CLSDST PASS the device somewhere
- **XFER2400** Check for sending to a site

XFER9000 Try to obtain the device lock

XFER9100 Allocate a one time broadcast message

XFER9200 Issue a message with ANETERM

Control Blocks

This portion of The Network Director's *Internals* manual discusses the **Control Blocks** used within The Network Director to control logic flow and to describe network elements.

Definition

A Control Block is a portion of dynamic storage that is mapped internally within The Network Director and is used to represent a portion of the logical network or a process occurring within the network.

All Network Director originated control blocks will reside within the partition or address space executing The Network Director. Any control blocks generated by VTAM or VSAM via normal processing of those access methods are fully under their control and may or may not reside in the partition or address space.

Naming Conventions

Each control block resides on The Network Director's source library as an independent member or book. It is contained in a 370 ASSEMBLER Macro with the form *TNDxxx*, where the xxx is a character string that is the control block's "short name" or acronym.

All Network Director control block fields begin with the three byte xxx from the control block's Macro name.

Individual Control Block Descriptions

Each Control Block is briefly identified via its "long name", short three byte name, purpose, and general connections with other Control Blocks. Additional comments may also be located under the individual descriptions. The reader should also remain aware that additional information related to the usage of and relationships between the control blocks is available in other portions of this manual.

The Control Blocks in use within The Network Director are:

Name	*Description*
AAE	Authorized Application Element
AAI	Application Action Item
ACB	Access Method Control Block
ADB	Application Definition Block
AIB	Authentication Information Block
AIE	Authorized Interval Element
ALB	Application Load Balance
ANE	Active Network Element
ATE	Authorized Terminal Element
AUE	Authorized User Element
BWA	Batch Work Area
DCE	Device Command Element
DFB	Dispatchable Function Block
DIR	System Directory
DMT	Director Message Text
EDL	Edit Data Line
EFR	External File Record
EWA	Edit Work Area
GDB	Group Definition Block
GSA	General Storage Area
HDE	Help Data Element
HIX	Help IndeX
IDE	Initial Data Element
IWA	Info Work Area
KDB	Keys Definition Block
LBE	LOG Buffer Entry
MDE	Message Data Element
MIE	Message Index Element
MIX	Message IndeX
MME	MSG Message Elements
MWA	MSG Work Area
NCB	Network Control Block
NDL	Network Distribution List
NEL	Network Element List
NIB	Node Initialization Block
PDA	Primary Director Area
PDE	Profile Data Element
PPE	Pending Parameter Element
RPL	Request Parameter List
SAR	System Accounting Record
SDB	Site Definition Block
SMR	System Measurement Record
SSE	Screen Selection Element
SSX	Single System eXtended
SWA	Selection Work Area
SWK	SHOW Work Area
TDB	Terminal Definition Block
UDB	User Definition Block
VWA	VTAM Work Area
WAE	Work Area Element

Each of the control blocks is discussed in further detail in the following pages of this publication.

A simple pictorial representation of how several of the basic control blocks are interrelated is:

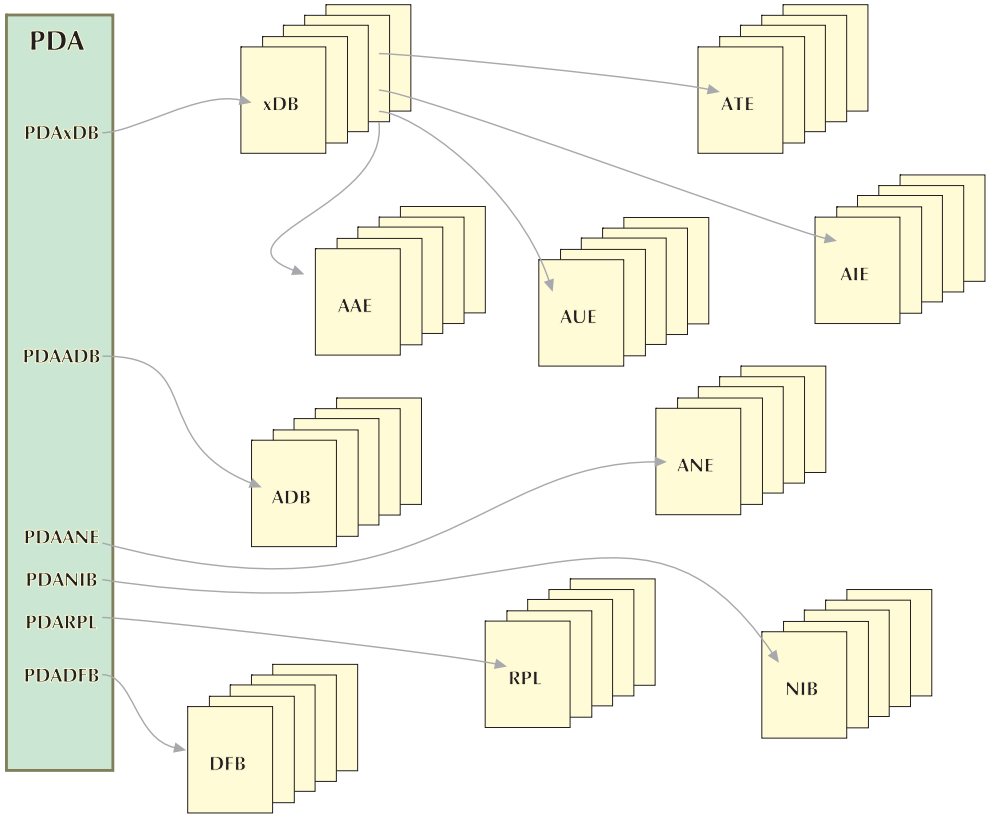

Figure 4. xDB Chains

This figure is intended to provide the reader with a general concept of how several of the control blocks are related. The precise details should be extracted from the control block DSECTs themselves.

You can produce an Assembly listing of all the DSECTs used within The Network Director by assembling the Module TNDDSECT (present in the DATA library).

As an authorized Network Administrator, you will be able to use the TNDDSECT listing and the DUMP command to view the contents of the control block chains within an executing copy of The Network Director. As with most software products, the flow of information through The Network Director is represented by the changing contents of the various control block chains.

You may also find this information useful when attempting to diagnose specific problem situations within The Network Director or attempting to diagnose situations associated with an LU that The Network Director is in session with (E.G. locating and viewing the current contents of a LU BIND image).

AAE - Authorized Application Element

Purpose	describe a single application that is available for use for the purposes of Selection.
Connections	Chained from any of the xDBs (TDB, UDB, or GDB) or the defaults area of the PDA. This chain is constructed when processing the APPLICATIONS= operand of the individual Parameter Statements.
Comments	The AAE simply describes the applications that may be used by the network element owning the chain. The application itself is still controlled via it's AIE, etc chains. During construction of the Application Selection Panel, SEL begins processing with the base control block (TDB, UDB, or GDB). Each AAE control block will create an entry on the Application Selection Panel (subject to the specification of the SELECTIONS operand). The ADB associated with the AAE will dictate whether the application is truly available to the network user or not (it may be HELD, etc).

AAI - Application Action Item

Purpose	contain the action characters and their associated commands
Connections	The AAI is chained from the ADB it is associated with.

ACB - Access Method Control Block

Purpose	provide the interface block between The Network Director and the VTAM, VSAM access methods.
Connections	The VSAM and VTAM ACBs are pointed to from the PDA.
Comments	The Network Director utilizes these as pseudo ACBs. They contain only those fields used within The Network Director. The functional modules VSAM and VTAM will translate the pseudo ACB to the real ACB address at function execution. See the section on VTAM Characteristics for a further discussion.

ADB - Application Definition Block

Purpose	identify a logical application and its attributes within The Network Director.
Connections	The ADB chain is maintained from the PDA in the order that they are presented to The Network Director.
Comments	Each ADB represents a single APPLICATION statement and is the base for all attribute chains related to the application (AIE, etc). The logical application's status is maintained in the ADB along with the physical target's name and status. Current network element connection counts are also maintained in the ADB.

AIB - Authentication Information Block

Purpose	identifies the information for a network element and the AUTHENTICATION processes and/or the internal password checking associated with the user
Connections	The AIB chain is maintained from the PDA in a random order and are utilized by LOGON to establish authorization to signon
Comments	the AIB is automatically maintained on the External File and will be updated each time a network element related to it logs onto the system. Passwords and associated history are stored in the AIB for SECURITY=DIRECTOR installations.

AIE - Authorized Interval Element

Purpose	describe a time and/or day interval that the associated network element is available for use.
Connections	Chained from any of the xDB control blocks including the ADB.
Comments	Each AIE contains a single interval that the base network element is available for activity within the network. It is possible to have many AIEs for a single network element. The AIE chain is built from the appropriate base control block under the control of the DAY= and TIME= parameters on the appropriate Parameter Statement.

ALB - Application Load Balance

Purpose	contains the BALANCE and ROTATE settings associated with a specific subsystem associated with the current pool of application subsystems.
Connections	Chained from the ADB that represents the initial subsystem choice. The ALB contains the currently connected counts and address of the other ADBs associated with the BALANCE grouping.
Comments	The ALB chain contains all information necessary for The Network Director to determine where the next user that selects the parent subsystem should be sent to.

ANE - Active Network Element

Purpose	identify a specific network element (think of it as an LU) and the status, etc currently associated with it.
Connections	Chained from the PDA in a continual chain. It is also pointed to by the DFB and the virtual NIB. The ANE contains the addresses of the UDB, PDE, DFB, NIB,ADB, and GDB currently associated with the network element. It is also the anchor for the SSE, DCE, and MME chains.
Comments	The ANE is the primary control block used to generally interface with activities occurring at the LU. The current ID area information is maintained in the ANE as well as merged attributes collected from the *pattern* definition control blocks (TDB, UDB, GDB, and DEFAULTs). The ANE also contains the address of the current and/or last input and output (logical and physical) terminal buffers associated with the device.

ATE - Authorized Terminal Element

Purpose	establish the network terminals that are valid for a particular USER or GROUP to be utilized at.
Connections	Chained from the GDB or UDB that defines the USER or GROUP that will be susceptible to the restriction.
Comments	Each ATE is generated from an increment of five individual entries in the TERMINALS= operand of the GROUP or USER statement. If the GDB or UDB has no ATE chain, the USER or GROUP is authorized to use any network terminal device. Like the AUE chain, the ATEs are checked by LOGON during logon processing to validate the authorization for the user.

AUE - Authorized User Element

Purpose	identify the list of specific Ids that may be used from a network terminal.
Connections	Chained from the TDB associated with the network terminal that has the restriction.
Comments	A TDB with no AUE chain has no restrictions related to who may use it (USER=NO is an exception). A single AUE is generated for each five operands of the USERS= operand on the TERMINAL statement.

BWA - Batch Work Area

Purpose	describe the work area utilized between the CSECTs that make up TNDUTIL (the batch processing program).
Connections	Addressed via common base register by all batch modules and passed between them as a parameter.
Comments	The BWA is utilized for communications between the batch controller (BATCH) and the detail processors (BAT1 or BAT2). The BWA is obtained by BATCH one for each execution of TNDUTIL and retained until the batch requests are all processed.

DCE - Device Command Element

Purpose	describe the queue of commands that are scheduled to be executed against a particular network element.
Connections	Chained from the ANE that describes the network element that has a queued command.
Comments	This command queue will be processed by SEL one at a time and before any interaction with the physical terminal. This is typically the manner with which The Network Director can control what will occur to the terminal when it returns from a target application. The NSI Stack Command request will place an entry on the top of the command queue.

DFB - Dispatchable Function Block

Purpose	identify a unique, independent piece of work within The Network Director. The DFB chain is used to control the order and speed with which particular network elements are processed.
Connections	Chained in a priority manner from the PDA. DISP will scan the DFBs from the top down and will dispatch any DFB that is ready to execute. Pointed to by the ANE when associated with a network element.
Comments	The DFB represents the basic dispatchable entity within The Network Director. Once initialization is complete, almost all activity will be accomplished as a "DFB task". This describes the characteristics of the Modules that run dispatchable via DISP. DFBs can be dynamically created within The Network Director or may be of a longer term, never terminating type (TIMER, OPER, etc).

The DFB chain is generally structured as:

1. Operator Services --------- OPER
2. Receive Any RPLS --------- INPUT
3. Monitor Task --------- MNTR
4. Network Elements --------- ASK then SEL
5. Maintenance Services ------ MAINT
6. Network Status --------- NTWK
7. Timing Services --------- TIMER

DFBs that are dynamically created are attached in general DFB class 4. That is, after Operator and Input Services, but before any long duration activities (File I/O, etc).

DIR - System Directory

Purpose	constitutes the System Directory for a single user of The Network Director. Information is collected from the DIRECTORY definition, the active security system, and the External File.
Connections	Chained from the ANE after a user logs on. The DIRs are also chained to each other based in the PDA.
Comments	The DIR chain can be initially loaded from the External File via the specification of the DIRECTORY ATTRIBUTE on the DEFAULT configuration parameter.

DMT - Director Message Text

Purpose	contains the characteristics associated with an internal Network Director error message.
Connections	Chained in random order from the PDA.
Comments	Installation modified DMTs will be stored into the External File by EDMT. Most DMTs are created by LOCAT from the MSGS CSECT and will reside on the DMT chain until they have not been referenced for a LOG determined number of scans down the DMT chain.

EDL - Edit Data Line

Purpose	represents the contents of a single line of text that will be manipulated by the Message Editor (EDIT) or the message Print (PRINT) action code.
Connections	Chained both forward and backward and anchored at a location determined by the calling Module. EDIT anchors it in the EWA and PRINT in the associated WAE.
Comments	MDEs and HDEs are translated by FORM into EDL chains. EDIT2's SAVE command translates the EDL chains back into the appropriate MDE or HDE for replacing in the External File.

EFR - External File Record

Purpose	describe the contents of a single External File record.
Connections	The individual types of EFRs are chained from the PDA in a randomly ordered sequential stack.
Comments	The External File itself serves multiple purposes within The Network Director. EFR is the term generally used to describe a record on the External File. The EFRs will be maintained in storage until MAINT or STOR decide to purge them to conserve main storage. Without regard to where it is stored, the EFR represents an element of the External File. Each EFR will describe specific information associated with its record type. LOCAT will obtain and translate the appropriate EFR into the internal control block, which will actually be manipulated within the other Director Modules. There is an individual EFRTYPE for HDEs, HIXs, MDEs, MIXs, DMTs, and PDEs.

EWA - Edit Work Area

Purpose	provides a common work area for sharing amongst the various Edit modules (EDIT, EDIT1, and EDIT2).
Connections	Allocated and released by EDIT from the MX storage pool, EDIT1 and EDIT2 obtain the EWA address as one of the initial parameters passed during the editing process.
Comments	The EWA contains information appropriate to the edit session, which will include partially completed block operations as well as prior Locate and Change values. It is also the anchor for the Edit sessions EDL chain.

GDB - Group Definition Block

Purpose	identify a GROUP that may be used within the logical network.
Connections	Chained from the PDA in the order that GROUP statements were encountered during execution. Pointed to by the ANE when the network element is a member of the GROUP.
Comments	A single GDB is generated for each GROUP statement. When a network element identifies itself as a member of the GROUP (by utilizing a TERMINAL with a GROUP= operand or a USER statement with the GROUP= operand), the GROUPs non default characteristics will be merged with the associated UDB/TDB and DEFAULT operands to produce the ANE values that will actually be utilized to interact with the network element. A GDB chain with no entries represents Configuration Parameters with no GROUP statements.

GSA - General Storage Area

Purpose	organize the allocation of a major portion of The Network Director's dynamically acquired main storage.
Connections	Chained from the PDA in individual chains (one chain per storage pool type).
Comments	The GSAs and their individual storage pools are managed by STOR. The GSA is used by DISPLAY to report on the storage consumption and usage characteristics within The Network Director's environment. The GSA will take on slightly different formats dependent upon the storage pool it resides in. Without exception, the GSA is the controlling prefix of each contiguous (virtual) portion of storage acquired. The GSA described main storage areas will not include the LOG buffer or Trace buffer (activated via GLOBALS LOGSIZE or TRACE operands).

HDE - Help Data Element

Purpose	contains the text associated with a single Info panel in compressed format.
Connections	Chained from the PDA in a standard sequential fashion.
Comments	The HDE is located via FIND calls to LOCAT, which will attempt to acquire the HDE from the External File if it is not already on the HDE chain. The HDE is typically converted to a chain of EDLs by the HDE FINDer for display or editing purposes.

HIX - Help IndeX

Purpose	contains the text words associated with a single logical point in the Help Hierarchy.
Connections	Chained from the PDA in a standard sequential fashion.
Comments	The HIX is edited via EHIX and is utilized by HELP to create and interpret Help Text keyword phrases.

IDE - Initial Data Element

Purpose	identify the arguments that will be passed to the target application subsystem when it is selected.
Connections	Chained from the ADB in the exact order that it is specified in the INITIAL-DATA= operand of the APPLICATION statement.
Comments	The IDE is utilized by SEL after a selection has been made, but before the CLSDST PASS=. The data forwarded to the target subsystem will begin with the INITIAL-FUNCTION operand followed by a blank. Each IDE will then control the subsequent arguments. Each IDE entry in the list is separated by a comma. If the ADB has no IDE chain, The Network Director will not pass any data when the application is selected.

IWA - Info Work Area

Purpose	describes the common work areas passed amongst the INFO processing modules (INFO and INFO2).
Connections	Passed between INFO processors as a parameter, the IWA is not *anchored* on any other control block.
Comments	The IWA is allocated when the INFO processor becomes active and will be freed when the terminal operator QUITs or terminates INFO.

KDB - KEYS Definition Block

Purpose	identifies a set of LU1 key definitions or synonyms
Connections	associated with a network element from the ANE after activation.
Comments	the KDB address is set from the name associated with the PROFILE definition for the TERMINAL.

LBE - LOG Buffer Element

Purpose	describes the fixed portion of the variable entries maintained in the LOG buffer queue in storage.
Connections	Contained totally within the LOG queue. The LBEs are allocated as variable length elements immediately following one another until they wrap back around on older LBEs.
Comments	The LBEs are managed via a Buffer Header which is made up of four fullwords as described below.

```
LBEID   DC  C'LBES'          eye-catcher
LBEEND  DS  A                Address of buffer end
LBEOLD  DS  A                Address of oldest entry
LBENEW  DS  A                Address of newest entry
```

MDE - Message Data Element

Purpose	contains the compressed text associated with a single Message Facility message.
Connections	Chained from the PDA. Access to the MDEs is not under any type of a Locking control. Thus, a given MDE can be addressed and pointed to by a variety of View, and Edit sessions.
Comments	The MDEs may be of a main storage only type or may be alternatively staged out to the External File (by MAINT). LOCAT has the responsibility for locating the MDE for all Modules.

MIE - Message Index Element

Purpose	represents a single message entry that a given network element may manipulate via the Message Facility.
Connections	This fixed length control block is the individual entry that makes up the contents of the MME. It is also pointed at by the SSE when the Primary Messages Menu is active.
Comments	The MIEs are collected by MSG1 into MMEs so that MSG can properly display the Message Menu and allow logical paging through the index of messages. The MIE is also used to map the entries in the MIXs.

MIX - Message IndeX

Purpose	is the control block that contains the entries associated with a particular network elements messages within the Message Facility.
Connections	The MIXs are chained from the PDA.
Comments	The MIXs are maintained by SEND, DELETE, and EDIT2's SAVE command. Each entry in the MIX (a MIE) represents a single message the user either owns or has received. The MIX is initially constructed for each user that has a message stored on the External File. It is maintained (in storage only) during execution to reduce potential VSAM i/o operations.

MME - MSG Message Elements

Purpose	represents the entries that may be displayed and manipulated via the Message Facility's Primary Messages Menu.
Connections	The MMEs are always chained from the ANE when the MSG processor is active.
Comments	The MMEs are created by MSG1 and manipulated by EDIT2 for COPY operations and MSG itself for paging operations and other message action codes.

MWA - MSG Work Area

Purpose	contains information associated with the processing of the Message Facility and facilitates communications between the various MSG CSECTS.
Connections	Address as a general work area, the MWA has an assigned base register in the individual MSG routines.
Comments	It is freed when the Message facility terminates.

NCB - Network Control Block

Purpose	establish a manner for application programs to communicate with The Network Director's Network System Interface.
Connections	It is contained in the user's partition or address space. Thus, it is not pointed to by any other Network Director control blocks. It is however passed across the LU6 link with NSI and is available to NRP for processing requirements.
Comments	The NCB is the primary communications block between the application program and NSI. It must be the first argument provided by the calling application program on each call to NSI.

NDL - Network Distribution List

Purpose	represents the individual userids that are collected by a single message facility user into a group identified by a unique name.
Connections	Chained from the PDA and loaded from the External File, as required.
Comments	The individual NDLs are "private" lists.

NEL - Network Element List

Purpose	describes a control block representing the operand values from a NETWORK=ELEMENTS= list specification.
Connections	Anchored from the appropriate GDB representing the operand values associated with the GROUP.

NIB - Node Initialization Block

Purpose	identify a logical unit within the VTAM terminal network.
Connections	Chained from the PDA as a sequentially ordered pool of NIBs. Addressed by the ANE and normally by an RPL when an request is active against the NIB.
Comments	A NIB is allocated during ANE creation and will exist for the entire Network Director's execution. The NIB is the staging area (via SHOWCB and MODCB) for the real VTAM NIB contents. This is a Network Director NIB and is equivalent to VTAM's NIB, but it is not the same.

PDA - Primary Director Area

Purpose	provide a central location for key chains and information critical to The Network Director.
Connections	The PDA is always pointed to by register 12 (once INIT is complete). It is the base for all the primary control block chains within The Network Director (See Figure 4 on page 116).
Comments	The PDA is split into multiple logical areas. Each area has a specific function and usage within The Network Director. Its initial values (prior to PARM processing) are set via the PATCH CSECT. The PDA is initialized during INIT's processing and will remain in fixed CB storage for the entire execution of The Network Director.

PDE - Profile Data Element

Purpose	contains the data values associated with a network element's Profile.
Connections	The PDE is pointed to from the ANE and is used by most DFB controlling tasks to evaluate the meaning of PFKEYs (when they are utilized by the terminal operator).
Comments	The PDE may be stored on the External File in a Profile EFR. The contents of the PDE are modified by the network element invoking the Profile process within The Network Director.

PPE - Parameter Pending Element

Purpose	provide an interface between a calling Module and the parameter processing routines (PARMS, PARM1, PARM2, PRMS)
Connections	Chained from the PDA, it is the responsibility of the DFB task that allocated it to free it when complete.
Comments	The PPE provides the area that can be utilized to process asynchronously multiple parameter type queries. OPER and ADMIN may both require PARMS services concurrently. To allow this concurrent usage, the PPE allows PARMS to keep track of the parsing, etc operations it has underway for each caller. The PPE is generated by the caller and will be optionally freed when the Parameter Processing is complete.

RPL - Request Parameter List

Purpose	represents an activity to be performed against a terminal or the External File (VTAM or VSAM).
Connections	Chained from the PDA in an unordered list.
Comments	RPLs are allocated, utilized, and freed as necessary by various functional Modules within The Network Director. The fields within the RPL are generally analogous to the real VTAM and VSAM RPL, but the control blocks are different in format.

SAR - System Accounting Record

Purpose	describe the accounting record that is produced by The Network Director.
Connections	Chained from the PDA in a standard unordered sequence on the SMR chain. The SMR and SAR is logically the same activity. Differentiation is made by STAT when the ACCOUNTING operand is evaluated.
Comments	When activated, The Network Director will evaluate the SMF, ACCOUNTING, and EVENTS keywords to determine when to create a SAR/SMR and on what medium it should be record. This chain is required to allow multiple DFBs to be concurrently creating and writing accounting records.

SDB - Site Definition Block

Purpose	describe the status and operands associated with a single SITE statement.
Connections	Chained from the PDA in the order they are presented to The Network Director.
Comments	Each SDB is periodically tested for availability by MNTR (on the same cycle as ADB availability testing).

SMR - System Measurement Record

Purpose	describe the accounting record that is produced by The Network Director.
Connections	Chained from the PDA in a standard unordered sequence. The SMR will normally exist only long enough for the DFB task to receive acknowledgement that it has been written to the statistics recording medium.
Comments	When activated, The Network Director will evaluate the SMF, ACCOUNTING, and EVENTS keywords to determine when to create a SAR/SMR and on what medium it should be record. This chain is required to allow multiple DFBs to be concurrently creating and writing accounting records.

SSE - Screen Selection Element

Purpose	identifies an element on a Selection or Primary Messages Menu that may be selected by the operator for activity.
Connections	SEL chains the Selection SSEs from the ANE. MSG chains the SSEs from its WAE. Each SSE will in turn contain a pointer to an ADB (for SEL) or a MIE (for MSG).
Comments	The SSE is the basis for matching input SBA sequences with screen related choices during Selection. It is also the control block chain scanned by NTFY to determine whether a particular network element has a particular choice on it and what the status of the selection is (UP, DOWN, etc). SEL maintains its SSEs in the CB pool and MSG in the MX pool. This is intended to recognize that continual allocation and deallocation of Selection SSEs would create unnecessary overhead.

SSX - Single System eXtended

Purpose	map the incoming control block that provides the necessary information to implement the Single System Image concept.
Connections	None. The SSX (if present) will reside in the CINIT RU User Data area.
Comments	The SSX is equivalent to the SSI buffer, but contains more formatted information and a more rigorous definition for future purposes.

SWA - Selection Work Area

Purpose	provides an anchor location for temporary (transient) information associated with the creation and processing of a Application Selection Panel.
Connections	None. The SWA is utilized by SEL, SEL1, and BUILD to pass information between them about the Selection process.
Comments	The SWA exists while a SEL DFB does. It will be freed and reallocate across terminal I/O if there is no TIMEOUT in effect. Otherwise, it will exist until LOGOFF or TIMEOUT occurs.

SWK - SHOW Work Area

Purpose	allows communications about the current SHOW process being performed to be passed amongst the various CSECTs that constitute the SHOW processor.
Connections	Addressed via a unique base register in each SHOW routine.
Comments	The SWK is freed when SHOW terminates.

TDB - Terminal Definition Block

Purpose	define a pattern of LU names (one or more) that will be utilizing the facilities of The Network Director.
Connections	Chained from the PDA in the order they are defined. The TDB also points at other control block chains required to complete the description of the TERMINAL statement (AAE, AIE, AUE, etc).
Comments	There is a single TDB for each TERMINAL statement. Note, however, that a single TDB may not define a single LU. The TERMINAL statement (like the USER statement) defines only a pattern that is to be utilized when creating a network element's characteristics. The TDB also contains a count of the number of network elements currently within The Network Director that are using this TERMINAL pattern.

UDB - User Definition Block

Purpose	identify one or more user(s) that is/are authorized to utilize the network.
Connections	Chained from the PDA in the order they are encountered during execution. The UDB is also pointed at by the ANE.
Comments	A single UDB is generated for each USER statement encountered. The Network Director will scan the UDB chain attempting to validate the Id: field (when entered by a terminal operator in the Identification Area). Just like the TERMINAL statement, the USER statement defines only a pattern for the Id field that allows The Network Director to associate the USER definition with a Network Element.

VWA - VTAM Work Area

Purpose	describes the common work area utilized while active in the VTAM interface routines
Connections	allocated by VTAM and passed amongst the appropriate modules. The VWA is not anchored from any other control block.
Comments	the VWA exists for each call to the VTAM interface and will be freed when the specific function is completed.

WAE - Work Area Element

Purpose	provide a location for individual module register saving and a general work area.
Connections	Chained from the DFB in the order they are utilized in a recoverable chain.
Comments	Each task that runs under a DFB will PUSH the caller's registers onto the bottom of the WAE stack and then acquire the next WAE for its own use. The WAE also contains a fixed number of bytes that can be used by the Module in any manner it so chooses. However, any data stored in the WAE will not be retained after the Module has returned to its caller (ENDed). The WAE may a fixed WAE (CB pool) or a dynamic WAE (MX pool). When a DFB terminates, DISP will scan the WAE chain and FREE any dynamic WAE that was acquired. The DFB will also itself be FREEd if it was allocated in recoverable storage (MX pool).

Macros

The Network Director makes use of multiple ASSEMBLER macros within its source Modules. This section of the *Internals* manual describes the **Macros** that are unique to The Network Director. The standard Operating System macros are not listed. The reader should refer to the appropriate IBM manual for descriptions of those macros.

Definition

An ASSEMBLER Macro is a sequence of preset ASSEMBLER instructions that can assist the ASSEMBLER program in generating ASSEMBLER instructions and definitions. The Network Director uses ASSEMBLER Macros to contain generalized code sequences, standard Module prologue and termination sequences, and to describe all the Control Blocks referenced in "Control Blocks" on page 113.

The Network Director's source code is assembled utilizing the IBM Assembler under CMS, which implies that some of the macro operations may not be functional when utilized with a DOS assembler. However, every effort has been made by NRS to minimize these difficulties for the DOS installation. DOS installations should have to modify the TNDSTART macro only (removing/replacing &SYSDATE and &SYSTIME). However, NRS has **not** verified that all Network Director source code will assemble and operate properly when using a DOS assembler.

Naming Conventions

All the Macros within The Network Director are present on the Source library as individual members or books. Each is prefixed with the standard Network Director prefix **TND**. There are a few macros prefixed with **NRS**, which are macros that are utilized throughout the North Ridge Software product line as well as within The Network Director.

Internally, each Macro generally follows the naming conventions that have been established for Modules. Logical labels within the Macros are in the general form .xxxxyyy where:

.	required by the ASSEMBLER
xxxx	is an acronym for the Macro (usually Macro name)
yyy	is a sequential numeric value

Figure 5. Macro Label Format

Most of the Macros have one or more operands. The operands are generally keyword and not positional. The keyword operands have been assigned logical names associated with the function each is to perform. However, there is no general convention used with the Macro operand names.

The Macros themselves have no standalone meaning or interconnections similar to the Module or Control Block descriptions. Each Macro takes on meaning during the assembly process only and its use during assembly controls its affect on The Network Director.

Macro Operand Syntax

Most of the operands present within The Network Director's Macros that provide an address or a value of an element can be specified as identified under the discussion of NRS$ARG. Quickly, arguments may be specified as a label, in standard register notation, as a external VCON, or a field within a DSECT.

Individual Macro Descriptions

Each of the Macros are discussed individually on the following pages. Their individual name, general purpose, and operands are briefly identified and discussed. As with the Modules and Control Blocks, the specific status of a particular Macro can be verified by investigating its content on The Network Director's Source Library.

The Macros currently within The Network Director are:

Name	***Description***
ALLOC	allocate a control block or storage
ACF	generate an ACF2 DSECT
CALL	invoke another Module
CICS	CICS SSI Generation
CMPR	Compare arguments including Wild Characters
END	Module termination
FIELD	Define a SBA sequence for a field
FILE	External File Services
FIND	scan control block chains
FREE	release main storage
GEN	Stage One Generation
ISSUE	log (issue) a message
OPSYS	invoke operating system services
PARAM	define a parameter
START	Module prologue
TAM	Invoke VTAM services
TERM	Logical Screen Activity
TIME	Acquire current Time of Day
TITLE	Module title, copyright, and version number
WAIT	WAIT for an ECB post and enter dispatcher
xxx	Control Block DSECT
NRS$ARG	inner macro to obtain a argument value
NRS$LOAD	inner macro to load a register
NRSDROP	register drop
NRSOPSYS	invoke miscellaneous operating system services
NRSREGS	register equates
NRSUSE	register using

ALLOC - Allocating Control Blocks or Storage

Purpose	provides an interface to the facilities of STOR. The calling Module typically uses ALLOC to reserve storage.
Comments	ALLOC is used for most of The Network Director's storage allocation. STOR acquires a set of virtual storage pages from the operating system and will manage these pages while responding to requests via ALLOC from other Modules within The Network Director. The primary control blocks are ADB, ANE, DFB, EFR, GDB, NIB, PPE, RPL, SMR, TDB, MDE, PDE, MIX, HIX, HDE, DMT, and UDB. The secondary control blocks are AAE, AIE, ATE, AUE, DCE, and IDE
TNDALLOC	[AMOUNT=resulting number of bytes] [ANCHOR=label] [AT=WAEPARMS] [BYTES=number of bytes] [CB=control block name] [NAME=identifier text] [POOL={CB|MX|TP|CB24|MX24|TP24}] [R15ZERO=label]

where:

AMOUNT requests that the specified halfword should receive the value that is equivalent to the BYTES operand (this is most useful when BYTES=TERMIO has been specified).

ANCHOR is the location of the chain anchor for all secondary chain allocations.

AT identifies the location at which ALLOC should build the required parameter list.

BYTES is the number of bytes to acquire. The storage will be allocated out of the requested POOL and full word aligned. A specification of TERMIO indicates that ALLOC should generate the code to compute the number of bytes required for a terminal input/output buffer and then allocate that much.

CB is the primary control block name (see "Control Blocks" on page 113) to allocate. STOR will return with the control block allocated, properly chained, and generally initialized.

NAME is the text that can be used to identify the primary Control Block that is to be allocated.

POOL is the storage pool that the BYTES should be allocated from. Allocation will be from a GSA element from 31 bit storage (on properly equipped systems) unless MX24, CB24, or TP24 is specified.

R15ZERO identifies the tag to branch to if the allocation is successful.

ACF - Generate an ACF2 DSECT

Purpose	generates the ACF2 control block maps (DSECTs) that are utilized to map ACF2 items.
Comments	ACF is used to generate the control blocks necessary to implement The Network Director's ACF2 facilities.
TNDACF	control block [BASE=register value]

where:

control block requests the generation of a specific control block (ACINHRT, ACGRSRC, ACVALD, ACVUAXB, ACTRM, ACMCB, ACUABUFR, or LIDREC).

BASE is the machine register to be associated (NRSUSEd) as the base register for this control block.

CALL - Calling another Module

Purpose	provides a standard manner with which to invoke another Module within The Network Director.
TNDCALL	module name [AT=WAEPARMS] [NOTZERO=label] [PARM=(parameter 1, parameter 2, ...)] [R0=value] [R15ZERO=label]

where:

module name is the Module to invoke. The operand may be a name, VCON, ADCON, Fullword, or literal specification that will result in one of the above. Register notation is supported as well. The name is also checked for one of the predefined names present in the module section of the PDA. If none of the preceding will result in a valid transfer, TNDCALL will assumed that "module name" is an external VCON and will generate ASSEMBLER instructions to acquire its address.

AT identifies where the parameter list should be placed.

NOTZERO is the location to transfer to if R15 is not equal to zero when the called module returns.

PARM specifies the arguments that should be passed to the module in the standard OS fashion. R1 will point to an address list, which contains a single address for each PARM supplied.

R0 identifies the value that will be placed into register zero prior to the called module being invoked.

R15ZERO is the location to transfer to if R15 is equal to zero when the called module returns.

CICS - CICS Single System Image Generation

Purpose	generate the executable code required to support CICS Single System Image.
Comments	Both the GMM and TCT options are designed to allow either to execute without the other with no adverse effect to CICS. See the appropriate Installation Guide (CICS SSI Installation) for more information.
TNDCICS	TYPE={GMM|TCT} [CSSF={YES|NO}] [CSSN={YES|NO}] [ENTRIES={10|value}] [GMMNAME=DFHGMM] [SNPNAME=DFHSNP]

where:

TYPE GMM requests the generation of TNDGMM (the Good Morning Message program) set symbols. TCT requests generation of the VTAM Logon intercept routine.

CSSF YES indicates that GMM should automatically CSSF LOGOFF the LU if the signon attempt fails. NO indicates that GMM should simply terminate and allow normal CICS procedures to handle the LU.

CSSN YES indicates that GMM should attempt to automatically enter the signon program (DFHSNP operand value) in an attempt to log the LU onto CICS (presuming an entry is found in TNDTABLE. NO indicates that there is no requirement to enter the signon program.

ENTRIES specifies the number of entries to be generated in the TCT's TNDTABLE. This is the number of concurrent requests that will be capable of being queued between the VTAM Logon exit and GMM.

GMMNAME identifies the CICS PPT entry name for the program to be given control if the LU that enters does not have a TNDTABLE entry. This will occur when the device did not come from The Network Director or the SSI=NO option was in effect.

SNPNAME identifies the CICS PPT entry name for the program to be given control when GMM attempts to automatically logon the LU.

CMPR - Compare Two Arguments

Purpose	generates the code required to compare two 8 byte arguments for equal values. CMPR takes into account Wild Characters in one or both arguments.
TNDCMPR	argument-1 argument-2 [NOTZERO=label] [R15ZERO=label] [WILD={1|2}]

where:

arguments are two positional operands that represent the two items that are to be compared.

NOTZERO identifies the label to branch to if the arguments do not compare.

R15ZERO identifies the label to branch to if the arguments do compare.

WILD establishes which of the arguments may contain wild characters. 2 indicates that both may contain wild characters. 1 indicates that only the first may.

END - Module Termination

Purpose	provides the code necessary to terminate the module's execution.
TNDEND	[R1=return code] [R15=return code] [DFB={CONTINUE|DONE}]

where:

R1 will set a value into R1. This is typically used to return a particular control block address, etc.

R15 will set a value into R15. This is typically used to reflect a particular return code to a calling module. If left unspecified, the value present in R15 prior to TNDEND will be returned to the calling routine.

DFB controls the status of the DFB upon exiting the Module. CONTINUE indicates that the DFB will remain available for DISP to process as normal. DONE indicates that the DFB task is complete and that the DFB should be removed from the DFB chain. This operand does not apply unless the Module is being dispatched as a DFB.

FIELD - Create SBA Sequence for a 3270 Field

Purpose	generates the proper Assembler DC statements to define a virtual 3270 field.
TNDFIELD	[ATTR=({SKIP|UNPROT|PROT|ALPHA|NUMERIC| BRIGHT|DARK|MDT|IC|PEN, ...)] [COL=1] [DATA=] ROW=

where:

TYPE describes the type of 3270 field to be generated. SBA generates a Set Buffer Address order. RA generates a Repeat to Address order.

ATTR identifies the contents of the 3270 field's attribute byte.

COL specifies the column number that the field will begin in.

DATA describes the contents of the 3270 field.

ROW specifies the logical row number that the field will begin in.

FILE - External File Services

Purpose	establish the mechanism to call for services from the External File manager.
TNDFILE	{OPEN|CLOSE|GET|UPDATE|ERASE|ADD} EFR=address EFRTYPE={MDE|PDE|MIX|HDE|HIX} [NOTZERO=label] [R15ZERO=label]

where:

request identifies the activity that is to be performed against the External File.

EFR points at the External File Record being operated on by the function.

EFRTYPE identifies the exact type of EFR being operated on.

NOTZERO identifies the label to branch to if the requested operation does not complete normally.

R15ZERO identifies the label to branch to if the requested operation completes normally.

FIND - Locating a Control Block

Purpose	provide a standard manner with which to locate a specific Control Block on a specific chain.
Comments	Each time TNDFIND is invoked a single control block will be located. FIND is the mechanism used for individual processing Modules to cause LOCAT to load an image of a control block from the External File.
TNDFIND	[AT=WAEPARMS] CB={DFB|EFR|GDB|ANE|TDB|UDB|HIX|HDE|MIX| MDE|PDE|DMT} [EFRTYPE={MDE|PDE|MIX|HIX|HDE|DMT}] [EXT=extension] [ITEM=cb type] NAME=identifier text [NOTZERO=label] [R15ZERO=label]

where:

AT is the location at which FIND may construct the parameter list required to call LOCAT.

CB identifies the primary control block chain that is to be searched.

EFRTYPE identifies the exact type of EFR being scanned for. This operand is required for CB=EFR requests.

EXT provides the Extension value for all xDB (except GDB) FINDs that is required to locate the proper control block.

ITEM allows the TNDFIND issuer to set the control block being searched for to an equate value (as defined by the PDA). This is an alternative to hard coding the control block type on the CB= operand.

NAME identifies the information that can be used to locate the proper primary control block.

NOTZERO identifies the label that should receive control if the control block is not found.

R15ZERO identifies the label to transfer to if the control block is successfully located.

FREE - Release Main Storage

Purpose	provides a uniform manner with which to invoke STOR to release previously ALLOCated storage.
Comments	For ADDRESS specifications, STOR will validate that the proper boundary has been specified and will free the bytes associated with the location specified in ADDRESS. When freeing a Control Block, STOR will not actually release the storage associated with the control block, but merely mark it as FREE.
TNDFREE	[ADDRESS=address to free] [AT=WAEPARMS] [CB=control block name] [POOL={MX|TP}]

where:

ADDRESS specifies the location of the storage that is to be freed.

AT is the location that FREE may build the parameter list.

CB is the name of the control block that should be FREEd.

POOL specifies the Director's storage subpool that the ADDRESS is to be freed from.

GEN - Stage One Generation

Purpose	produces the necessary job control statements to complete The Network Director's installation.
Comments	TNDGEN is a installation aid only and is not used within The Network Director for any functional purpose. Additional information and discussion of TNDGEN can be located in the Network Administrator's Guide.

ISSUE - Issue a Message

Purpose	establishes a uniform manner to invoke LOG and **may** present a message back to the user or to the Network Administrator. Messages issued via ISSUE may also be placed into the LOG queue for later viewing via ADMIN.
Comments	ISSUE will invoke LOG to acquire the message text and to accomplish parameter text replacement. LOG will also decide whether to place the message into the LOG queue or write it to the operator (based on message attributes). LOG will place the resulting message text into the location specified in LOCATE only if the operand is used.
TNDISSUE	[LOCATE=message target location] MESSAGE=message number [PARM=(parameter 1, parameter 2, ...)] [CBS=(block 1, block 2, ...)]

where:

CBS specifies one or more Network Director control blocks that should be available for variable replacement to the message processor (TNDLOG). The section on "Variables" in the *Network Administrator's Guide* identifies the variables that may be used in the messages and identifies which control block the variable is derived from.

Inclusion of the variable preceded by a variable substitution character in the message text indicates your request to place the variable value into the message at execution time.

LOCATE specifies a storage location where the resulting message text should be placed. This is typically used to allow the calling Module to place the message text onto the network element's panel.

MESSAGE is The Network Director's message number that is to be issued.

PARM identifies the individual arguments that will be used to replace variable text within the message.

Each variable passed via the PARM operand can be thought of as a positional argument, which in turn will be utilized by The Network Director's message processor to replace the equivalent positional specification in the message text (either dollar signs for alphanumeric data or pound signs for numeric values).

OPSYS - Operating System Services

Purpose	establish a standard manner to request The Network Director's specialized operating system services.
Comments	OPSYS provides the operating system facilities that are necessary and are not already covered by NRSOPSYS.
TNDOPSYS	{OPEN|CLOSE|GET|PUT} [AT=WAEPARMS] DCB= [NOTZERO=label] [PARM=(parameter 1, parameter 2, ...)] [R15ZERO=label]

where:

AT identifies the location at which OPSYS can build the parameter list.

DCB is the address of the DCB or DTF to be operated on.

NOTZERO specifies the label to which to branch if the operation does not successfully complete.

PARM specifies arguments required by specific "functions".

R15ZERO identifies the label to pass control to if the operation is successful.

PARAM - Create a Parameter

Purpose	creates the necessary bit and byte strings to establish a Network Director Parameter and its associated operands.
Comments	PARMS processes all requests within The Network Director that are in the generalized parameter format. PARAM defines the table (PRMS) that is used during PARMS processing to validate the input data stream.
TNDPARAM	[{STATEMENT|KEYWORD|POSITIONAL|NAME}] [ID=numeric value] [LENGTH={8|maximum length}] [LIST=maximum entries in list] NAME=parameter-name [TYPE={NUMERIC|ALPHA|DAY|TIME}] [USE=(PARM,OPER,DECK)]

where:

positional identifies the general characteristics for the parameter. STATEMENT specifies a Parameter Statement. KEYWORD identifies a keyword operand on the last Parameter Statement and POSITIONAL declares a positional parameter (specified in the order they will appear). NAME indicates that the parameter being specified is the identifying "name" for the control block.

ID is the numeric value that will be placed in the PPE identifying the statement associated with the PPE being processed. The ID is automatically assigned for all PARAM entries except STATEMENTs.

LENGTH is the maximum size the parameter can be.

LIST identifies that the parameter is a KEYWORD with a list format and the number of entries that can be in the list.

NAME is the 1 to 16 byte name associated with the Parameter Statement or Operand being defined.

TYPE identifies the general characteristics of the Operand. NUMERIC specifies that the parameter must be convertible to a numeric value (including the special Network Director suffixes). ALPHA accepts any string of characters. DAY and TIME specify that the parameter has those specialized data formats.

USE controls when the parameter or operand is valid. PARM enables the parameter during initial Configuration Parameter processing, OPER enables the parameter for usage from the Operator's console, and DECK allows use from the Network Administrator's terminal.

START - Module Prologue

Purpose	produces the standard Network Director initialization instruction sequences for individual modules and DFB tasks. DOS installations will have to modify this macro to eliminate or replace the &SYSDATE and &SYSTIME specifications.
TNDSTART	[CHAIN={YES|NO}] [BASE2=register] [BASE3=register] [BYTES=72] [PARM=register] [WAE={YES|NO}]

where:

name specifies the CSECT name for this module.

CHAIN specifies whether the prefix generated code should generate the standard register saving instruction or not.

BASE2 specifies that this Module will require a second base register and which register to use.

BASE3 specifies that this Module will require a third base register and which register to use.

BYTES identifies for WAE=NO generations how many bytes of storage are to be allocated from the operating system associated with the save area that will be obtained.

PARM for WAE=NO generations, this identifies a register that can be utilized to save and restore the contents of R1 during GETMAIN/GETVIS activities.

WAE identifies whether the Module being STARTed will operate as a DFB task (under control of The Network Director's dispatcher).

TNDSTART assumes that register 10 (R10) will be the routine's base register and that R15 will contain the routine entry point address when it first receives control.

TAM - Invoke the Terminal Access Method

Purpose	provides the mechanism to request work of the terminal access method providing physical connection to the terminal network.
TNDTAM	{OPEN|CLOSE|SETLOGON|INQUIRE|OPNDST| SIMLOGON|CLSDST|SENDCMD|RCVCMD|SEND| RECEIVE} [ACB=PDATACB] [NOTZERO=label] [PARM=] [RPL=RPL] [R15ZERO=label]

where:

ACB identifies the ACB associated with the access method that is used by The Network Director to identify itself.

NOTZERO is the label to transfer control to in the event the operation requested is not completed normally.

PARM specifies additional characteristics associated with certain of the primary functions.

RPL establishes the RPL that will be used to accomplish the request.

R15ZERO identifies the label to receive control next if the requested operation is completed successfully.

TERM - Logical Screen Facilities

Purpose	allows a DFB controlling Module to make a request of the logical terminal service routine (SCRN) for terminal input or output.
Comments	Once READ operations are complete, the ANE will contain the address of the input buffer in ANESCRN. ANELSCR will contain the length of the input. ANECURS will contain the logical cursor address and ANEAID the logical AID key used. Note that all SBA sequences present in input buffer or presented via TERM to SCRN are expected to be logical screen addresses. SCRN will accomplish all translation of logical SBAs to real SBAs and back during interactions with the actual device.
TNDTERM	{READ|WRITE} [DATA=ANESCRN] [ERASE={YES|NO}] [LENGTH=ANELSCR] [MESSAGE=0] [MSGSIZE=0] [NOTZERO=label] [R15ZERO=label]

where:

option READ indicates that terminal input is desired. WRITE indicates that the specified output buffer is to be written to the device.

DATA is the fullword address of the output data buffer.

ERASE indicates whether the entire screen is to be cleared on the output operation (YES) or not (NO).

MESSAGE is the address of a message that should be placed into the Broadcast area of the output panel. This will override any outstanding Broadcast Message queued via MDEs for one output screen only.

MSGSIZE is the size of the Broadcast Area Message to be inserted.

NOTZERO is the label to transfer to in the event the requested operation does not complete properly.

R15ZERO identifies the statement label to pass control to if the output operation completes normally.

TIME - Acquire Current Time of Day

Purpose	establishes an interface to the STCK routine to acquire the current time of day in multiple formats.
Comments	The current time of day or the converted time (for CONVERT invocations) is always returned in register 1 in the packed format 0HHMMSSF. R0 contains the date in 0YYMMDDF format. Normally, TNDTIME merely causes STCK to refresh the values for time and day, etc in the fixed fields within the PDA.
TNDTIME	[ANSWER=] [CONVERT=0] [DATE=]

where:

ANSWER identifies the 8 byte field that should receive the formatted EBCDIC time of day in HHMMSS format.

CONVERT specifies a fullword containing the stck value that should be converted to EBCDIC time.

DATE specifies the 8 byte location that should receive the formatted EBCDIC date in DATE-FORMAT format.

TITLE - Module Title

Purpose	provides a standard manner with which to begin each Module. TITLE produces the cosmetic Network Director title at the top of the ASSEMBLER listings as well as the Version Number.
Comments	TITLE also will set those ASSEMBLER GLOBAL SET SYMBOLs that impact the general characteristics of the Module being ASSEMBLED. As an example, The Network Director's formal Version number is set within the TITLE macro.
TNDTITLE	title

where:

name is the internal 1 to 5 byte name used to identify the Module.

title is the text that briefly identifies the function of the module.

VART - Variable Table

Purpose	generates an appropriate entry into the variable table that defines how the variable processor should extract and format a variable.
Comments	TNDVAR uses the resulting table to interpret, parse, and replace values into appropriate locations at execution time.
TNDVART	variable name control-block [LEN=] [LENOUT=] [OFFSET=0] [PARM=0] [SPECIAL=] [TYPE=MVC]

where:

variable name establishes the literal string that represents the variable.

control-block is the control block in which the originating data resides.

LEN sets the length of the originating field.

LENOUT establishes the length of the destination field (after conversion).

OFFSET is the offset within the identified control block of the originating data.

PARM a halfword value that passes parameters to a special processing routine

SPECIAL identifies a special processing routine that will receive control for the variable being defined

TYPE establishes the type of operation performed to move the data from the originating field to the destination field (MVC or CVD are valid).

WAIT - WAIT for Event

Purpose	establish a common manner with which to enter The Network Director's DISPatcher. Additionally, WAIT allows the specification of one or move events, any of which (when posted) will cause the DFB to be dispatched by DISP.
Comments	WAIT will cause The Network Director's dispatcher (DISP) to be entered. The DFB issuing WAIT will be redispatched immediately after the WAIT macro when one or more of the ECBs specified on the WAIT have been posted. This includes the expiration of the SECONDS interval (if specified) or terminal input (if specified).
TNDWAIT	[ECB=address of ECB] [ECBLIST=address of ECB addresses] [SECONDS=seconds to wait] [TERMIO={NO|YES}]

where:

ECB identifies the ECB that, when posted, will cause this DFB to become eligible for dispatching.

ECBLIST identifies the starting location of a list of addresses. Each of the addresses points at an ECB that should be included in the consideration for DFB dispatching.

SECONDS establishes a "timed wait" interval in addition to any ECB specified on the ECB or ECBLIST operands.

TERMIO indicates whether terminal input should be considered as an event that will logically POST this DFB (YES) or if input is to be ignored (NO).

xxx - DSECT Generation

Purpose	generalize the method utilized within The Network Director to generate ASSEMBLER DSECTs for the various control blocks.
Comments	The PDA and DFB DSECTs are generated with a BASE operand default of R12 and R11 respectively. Many of the macros within The Network Director will expect proper addressability to these control blocks and establishing their default BASEs in the Macro guarantees that this is accomplished. As discussed in the earlier section on Control Blocks, the xxx value can be replaced with any valid Network Director acronym to generate the appropriate Control Block. Remember, you can generate a complete listing of all Network Director control blocks by assembling TNDDSECT on the distribution libraries. The TNDACF macro is a special case of DSECT generation. It also has an additional positional operand which implies which ACF2 control block should be generated. Valid positional parameters are: ACVALD, ACTRM, ACVUAXB, ACMCB, or LIDREC.
TNDxxx	[BASE={NO|register}] [DSECT={YES|NO}]

where:

BASE establishes the register to be utilized as the base register in a USING statement. NO indicates that the DSECT should not have any BASE generated for it.

DSECT establishes whether the Control Block should have a DSECT statement generated or not.

NRS$ARG - Generalized Argument Macro

Purpose	simplify the coding of Macro operand handling via the use of an inner macro to resolve register notation, etc. This macro provides the general ability to specify argument values in a variety of manners.
Comments	The value may be an Assembler self defining term or a symbol equated to a register in standard register notation. If the value begins with a open parenthesis "(" and has two operands, the following rules apply: **S** a field present in a DSECT whose address is to be placed into the register. This actually can be any Assembler label that will allow a SCON type reference. ***** an indirect address. That is, the second field contains an reference to a location in storage whose contents should be loaded into the register. **H** a halfword value that should be placed into the register. **X** a single byte that should be inserted in the low order byte of the register after clearing the entire register.
NRS$ARG	register,value

where:

register specifies the register to be initialized

value is the argument whose value or address is to be placed in the register.

The following samples should help clear this up:

```
NRS$ARG R1,13                LA   R1,13

NRS$ARG R3,WAEWORK           LA   R3,WAEWORK

NRS$ARG R5,(R4)              LR   R5,R4

NRS$ARG R1,(S,ANENAME)       LA   R1,S(ANENAME)

NRS$ARG R2,(*,ANEDFB)        L    R2,S(ANEDFB)

NRS$ARG R1,(H,ANELINES)      LH   R1,S(ANELINES)

NRS$ARG R3,(X,PDASMF)        XR   R3,R3
                             IC   R3,S(PDASMF)
```

NRS$ARG provides the ability for the macro operand to reside just about in any type of storage definition. It is primarily an aid when accessing the contents of DSECT defined storage.

NRS$ARG also has some overlap with NRS$LOAD, which will eventually be replaced within The Network Director.

NRS$LOAD - Argument LOAD Macro

Purpose	simplify the coding of Macro operand handling via the use of an inner macro to resolve register notation, etc.
Comments	The NRS$LOAD macro is intended for use as an internal macro. NRS$LOAD interrogates the type of "value" that has been specified and based upon the value's "type", NRS$LOAD will initialize "register" in the proper manner. It will operate as follows: 1. If the value begins with an open parenthesis "(", NRS$LOAD will use the 370 Load Register mnemonic to place the value into the register. 2. If the value is an A-type, V-type, WXTRN, EXTRN, or fullword constant, NRS$LOAD will use the 370 Load mnemonic. 3. If the value is a Y-type and halfword constants will be loaded via the 370 Load Halfword mnemonic. 4. If the value is none of the above, NRS$LOAD will generate a 370 Load Address mnemonic to acquire the value and place it into the register.
NRS$LOAD	register,value

where:

register specifies the register to be initialized

value identifies the element whose address or value should be placed into the register.

NRSDROP - Register Drop

Purpose	remove previous ASSEMBLER USING assignments from a register and from the internal USING table.
Comments	Either the register or the name specification will cause both the register and the name to be DROPped from the current USING table (which is maintained by NRSUSE and NRSDROP).
NRSDROP	{register|name}

where:

register specifies the register to be DROPped

name is the value that has previously been associated with the register via a NRSUSE Macro.

NRSOPSYS - Operating System Services

Purpose	establish a standard manner to request operating system services from NRSMVS or NRSVSE.
Comments	NRSOPSYS provides the majority of basic operating system dependent services used within The Network Director. Other more specialized operating system services are localized in TNDMVS or TNDVSE.
NRSOPSYS	function [PARM=(parameter 1, parameter 2, ...)]

where:

function identifies the operating system service being requested.

Valid functions are: GETMAIN, FREEMAIN, ABEND, WTO, INITOPER, READOPER, STIMER, STAE, POST, WAIT, ID, SNAP, and TIME.

PARM specifies arguments required by specific "functions".

NRSREGS - Register Equates

Purpose	isolate the primary register equates in a single location.
Comments	The LIST option produces onto the ASSEMBLER listing via MNOTEs the current contents of the internal table maintained by NRSUSE and NRSDROP.
NRSREGS	[{DEFINITION|LIST}]

where:

DEFINITION requests that NRSREGS simply generate the standard register equates.

LIST requests a list of the current ASSEMBLER GLOBALs that contain the current USING status of the various registers and their equate symbols.

An example of this list is:

```
**     CURRENT REGISTER EQUATES
**
**          R1  - WAEPARMS
**          R2  - WORK REGISTER
**          R3  - WORK REGISTER
**          R5  - PPE
**          R10 - TNDDISP
**          R11 - DFB
**          R12 - PDA
**          R13 - WAE
```

NRSUSE - Register Using

Purpose	establish a common manner to establish register equates (as they apply to the various control blocks, etc.
Comments	USE is the vehicle for placing USING characteristics into the internal register table utilized by DROP and REGS to manipulate the status of register usage.
NRSUSE	base reference,register

where:

base reference is the control block or other storage reference that will be addressed via the register in operand two.

register identifies the register value that will be used to address the item reference in operand one.

Internal Facilities

The Network Director utilizes many common methods to manipulate the work flow within its execution environment. This section of the manual discusses several of these **Internal Facilities**.

Register Usage

All registers within The Network Director are referenced via the register equates produced by NRSREGS. Each major segment of the individual modules contains a NRSREGS LIST to document the register usage in effect when the segment is entered.

The only register usage that will not follow this convention is contained within non Network Director Macros. The standard IBM Supervisor Services Macros are an example. The macro expansions contain absolute register references and will not be cross referenced in the assembler's cross reference listing or via the NRSREGS LIST mechanism.

Individual registers have been assigned general purposes within The Network Director. Some of them are relatively fixed in purpose and others are widely varied in their use.

The internal assignments for the general purpose registers are:

Register	***General Purpose***
R0	is reserved for usage within Macro expansions.
R1	is generally used to pass parameters and parameter list addresses.
R2 - R9	are used as the individual Module desires. Generally, Control Blocks are addressed with the higher registers and progressing to the lower valued registers. Conversely, general work registers typically begin with R2 and work to the higher numeric values.
R10	is the base register of the Module currently in control.
R11	is always the base for the DFB (for DFB based tasks).
R12	will always point at the PDA (after it has been created by INIT).
R13	will always point at a valid OS save area. Once the basic Network Director dispatching environment is set up, this will be the base for the WAE.
R14	is used to create intermodule transfers via the 370 BALR instruction. Thus, this register typically contains the address where the current Module received control from or the location where this Module last invoked another Module.
R15	is used as the general return code register. It is set by many of the routines within The Network Director to indicate conditions associated with the Module's processing.

Figure 6. Register Conventions

At no time is the same register used for two or more different purposes. If registers are at a premium, their values are stored and NRSDROPed prior to loading new values and NRSUSEing them in a new fashion.

Chain Construction

Most The Network Director Control Block chains are connected in an identical manner. Each Control Block's second fullword is named xxxNEXT and contains the address of the next control block on the chain. The last entry on the chain will contain binary zeroes in the NEXT field. The WAE and EDL chains are the exceptions to this basic rule. They also have xxxPRIOR fullwords which point at the control block on the chain immediately before the current one.

This control block structure applies to secondary chains as well as primary chains. This structure allows each chain to be mapped and scanned with a minimum of instructions. As an example:

```
          LA    R9,PDAADB-4             START AT TOP OF CHAIN
          NRSUSE ADB,R9                 FOR THE ASSEMBLER
LOOP      DS    0H
          ICM   R9,B'1111',ADBNEXT      GET THE NEXT ADB
          BZ    ENDCHAIN                EXIT IF END OF CHAIN..
          CLC   .....                   IS IT THE ONE I WANT?
          BNE   LOOP                      .. NO, TRY THE NEXT
```

The chains are manipulated in different manners depending on the specific type. Some chains have an implied priority (like the DFB chain) while others are simply sequential in nature.

Vector Transfer

The Network Director is a structured system of multiple functional modules. These modules will transfer control to each other in one of two standard manners.

Any Module that is not dispatched under the control of a Network Director DFB will follow standard 370 register saving conventions and will be entered with R15 containing its entry point address and R14 containing the address of the return location. Any parameters that are required will be pointed to by R1.

DFB tasks will also be entered with R15 containing the entry point address and R14 the return location. Register saving will be accomplished by storing R0 thru R15 at the current WAE (addressed by R13) and incrementing R13 to point at the next WAE. The WAEs are chained off the DFB and can be viewed as a LIFO stack of control blocks that identify the path through the various Modules that the particular DFB has taken.

Prior to PDA creation, all Modules are invoked via specific VCON references in the calling module. Once the PDA is established, INIT will initialize most commonly used Module addresses in the PDA Vector Area. The Vectors are basically the current addresses of other Modules within The Network Director's load module (or phase).

CALL will then respond to the target Module name being described as a fullword in the PDA and will Load the address from the PDA for transfer instead of generating additional VCONs. This technique reduces the size of The Network Director's Load module (or phase) and provides a standard manner with which every call to a specific Module can be tracked and managed.

Operating System Services

Any request for operating system services is issued via OPSYS and results in a CALL to an entry point in the various Network Director operating system dependent Modules. This interface between the operating system dependent Modules (MVS, VSE, NRSMVS, and NRSVSE) and other Modules is exactly the same for all supported operating system environments.

Storage Management

The Network Director is a fully reentrant implementation. This mandates that The Network Director make use of dynamically acquired storage for all work areas and control block chains.

To further improve the virtual storage reference pattern within The Network Director, all storage that is acquired from the operating system is grouped into separate pools of storage. Each pool has differing allocation and reference characteristics. The classifying of storage into various pools is intended to reduce The Network Director's paging requirements as well as to improve the utilization of acquired storage.

Storage Pools

The Network Director currently uses three pools or classes of storage. They are named the Mixed (MX), Tele-Processing (TP), and Control Block (CB) pools.

During initialization, INIT acquires the first GSA for the CB pool to establish the PDA. Thereafter, each storage pool expands as necessary to satisfy the internal requests. STOR will automatically acquire additional operating system storage when an individual pool fills. If the storage acquisition fails, the DFB making the request will enter a "storage stall" condition until some of the existing storage is freed by another DFB. The LOG will also receive several messages associated with the storage shortage.

Tele-Processing

The TP Pool is utilized for all terminal input and output buffers. It contains the actual data transmission streams that represent The Network Director's panels. It is a reusable pool of storage that is allocated by STOR in 256 byte segments.

The TP pool is the area that VTAM will be placing information associated with RECEIVE commands. It is also the area that logical to physical SBA translation will be occurring in (under SCRN's control).

Control Block

The CB Pool contains all of The Network Director static control blocks. They are all grouped into this pool in anticipation of continual reference during chain scans. This allows The Network Director to "pack" control blocks on a small number of virtual pages and will improve chain scan time.

The CB pool is further broken into "sub-pools" that pack control blocks of the same type onto the same virtual pages.

The CB pool's storage is allocated in 4 byte increments and is not reusable. Once a control block is allocated, its storage will never be freed for other usage. The storage is allocated on a FIFO basis with the sequence of chains based upon the order The Network Director encountered the requirement to generate a Control Block.

Mixed

The MX Pool contains all dynamic storage that is not included in the TP or CB pools. It is typically used for temporary extra work space for a particular DFB. Virtual work space for the Editor is an example of MX storage usage.

The MX pool is allocated in 128 byte segments and is reusable. It is typically the storage pool with the highest reference characteristic as its usage is as a scratch pad area for all internal functions within The Network Director.

Managing the Storage Pools

Each storage pool is made up of one or more virtual storage pages. Each virtual storage acquisition is mapped via a Network Director GSA (General Storage Area) Control Block. The GSAs for each pool are chained from the PDA in the order that the storage was acquired.

The GSA for each storage pool segment describes the capacity of the segment, and its current allocation status. The allocation status of each segment is maintained via a *byte allocation map* technique where each byte within the byte map represents a pool storage element. The pool storage element's size differs depending upon the pool, but the mechanism is the same for all the pools.

For 31 bit capable systems, each GSA element identifies the pool element as residing above or below the 16MB addressing point. All TNDALLOC calls for storage will be allocated from 31 bit addressing GSA elements unless the caller specifically requests MX24, CB24, or TP24.

Storage is allocated in all the pools on a first fit basis starting with the first GSA on the appropriate chain. Allocation always occurs on a fullword boundary and in contiguous virtual storage (a single allocation may not span GSA segments of any given pool).

WAE Storage Space

Each DFB based task can also utilize a general purpose storage area that is tacked on the end of the WAE and after the general purpose register save area. This space differs from ALLOCated storage in that its contents will be preserved only while the Module is in control. Once the Module returns control to another "level", the contents of the WAE become unpredictable.

The WAE space is typically used for short term work areas that are required by the Module only during a single execution of the Module. It is also of a fixed size and may not provide sufficient storage for all Module needs.

Dynamically acquired WAE space is freed during DFB termination by the dispatcher. Installation exits using TNDSTART and TNDEND macros do **not** have to allocate, free, or chain their own save areas. This is automatically done by The Network Director's macros.

Storage Balancing

The Network Director, by default, does not release any storage obtained from the operating system until termination. However, during startup it is typical that the amount of storage allocated will exceed what is actually necessary to process for the remainder of execution. Thus, a mechanism termed "storage balancing" is available within The Network Director.

This allows each installation the ability to specify the percentage of the high water mark of storage required within a storage pool should be retained for later execution. The percentage for retention is specified for both the MX and TP pools as different operands on the STORAGE-BALANCE operand of the GLOBALS statement.

During execution, if the amount of storage actively utilized by The Network Director drops below the percentage specified, The Network Director will attempt to locate a pool element that is completely empty that it can release back to the operating system (FREEMAIN/FREEVIS). If located, the pages are released, which will increase the amount of storage actively utilized. This process will continue slowly until the specified percentages are generally achieved.

Dispatching

The Network Director is a single operating system task and manages its internal functions via an internal dispatcher. DISP contains the dispatcher and its philosophy and characteristics are discussed in the following paragraphs.

Philosophy

The Network Director runs within the computing facility as a key component of the logical network. It is basically a service function to the network users and will generally require good throughput characteristics in order to provide adequate service time. This throughput requirement led to the philosophy that developed the internal dispatcher.

Many of The Network Director's functions and facilities will require other operating system services. Many of these services generally have implied operating system WAITs associated with them. If The Network Director were to WAIT for each of these services with implied WAITs in a single thread fashion, the service time for a large network would become excessive.

The Dispatcher's goal is to provide an environment where individual Network Director components can request their service of the system without impacting any other component. The dispatcher also provides a manner with which to prioritize various activities within the network. These facilities are provided in addition to the basic dispatcher's requirement. That requirement is to provide an environment where network operations can be overlapped to optimize throughput.

The DFB

To manage each function or facility within The Network Director, the dispatcher utilizes the Control Block named the DFB (Dispatchable Function Block). It represents a "piece of work" that requires CPU time to complete. The DFB chain is continually scanned by the dispatcher. Each scan evaluates whether any of the DFBs are ready for dispatch.

If a particular DFB is ready, the dispatcher will set up the proper registers and give the DFB task control of the partition, virtual machine, or address space. Once the DFB has completed its work or when it is at a "wait point", it will return to the dispatcher. When control is returned, the dispatcher simply begins its scan of the DFBs again.

WAITLIST Mechanism

Whenever a DFB based task is at a "wait point", the Module issues the TNDWAIT macro. This macro causes the ECB or ECBs associated with the wait point to become associated with the DFB task. The dispatcher is then entered and will begin its scan of the DFBs again. When one or more of the ECBs associated with the DFB's wait point are posted, the DFB will be redispatched from the point it WAITed at.

During the dispatcher's scan of the DFBs, it is collecting all real DFB related ECBs into a real WAITLIST maintained in DISP's storage. If DISP encounters end of chain and there are no DFBs to dispatch, DISP will issue a real operating system WAIT on the multiple ECB list constructed during the DFB scan.

This WAIT will also include TIMER's timed wait ECB. Whenever one or more of these real ECBs are POSTed, DISP will receive control again from the operating system and the DFB scan will begin again.

In general, DISP attempts to maintain control and overlap long WAIT operations with other productive work. This balancing of the various DFB tasks and their overlapping characteristics allows The Network Director to maintain a high throughput characteristic.

Dynamic/Static DFBs

Clearly, the central control block for controlling Network Director activities is the DFB. The DFBs can be generally grouped into the dynamic and static types.

Static DFBs are those DFBs that will never terminate for the entire execution of The Network Director. These DFBs represent internal Network Director service facilities like OPER, TIMER, INPUT, MNTR, and MAINT.

Dynamic DFBs are created during The Network Director's execution in response to a request for work from a network element. An interrupt from a network terminal that is not currently connected to a DFB will create a DFB to handle the interrupt. DFBs can also be created by operator request for action. Requesting a Print Action Code within the Message Facility will create a DFB to manage the output operations to the designated printer.

Dynamically created DFBs and their associated WAE chains are allocated from the MX storage pool and are FREEd during DFB termination processing.

External File

The Network Director makes use of an External File for multiple purposes. This External File is a standard VSAM KSDS and is used via conventional VSAM requests that are coordinated through TNDVSAM.

Overview

The Network Director generally makes use of only main storage resources to accomplish the majority of its tasks. This is done to minimize total system requirements to keep the logical network functioning in an efficient manner.

However, there are certain facilities and characteristics that can not be satisfied through the exclusive use of main storage. Generally, these facilities are those that require large volumes of information or the retention of information across Network Director executions.

The External File provides the mechanism to satisfy these requirements. It is divided into six distinct segments. Each segment of the External File is dedicated to satisfying a specific requirement. The file segments are:

Messages Message Facility Messages

Profile Modified Profiles

Codes Network Director Messages

Info INFO Mechanism

Definitions Stored definition blocks (SAVE and RELOAD targets)

Access Access Information Blocks (used for AUTHENTICATION and internal password manipulation

Figure 7. External File Segments

Message Facility Messages

Any Messages that have been specified as eligible for disk queueing (via the GLOBALS statement) will be stored in the External File as compressed variable length character strings. The Messages themselves are stored in an unFORMATted manner. The FORMATting will occur within FORM after a request to manipulate the message has been received.

Modified Profiles

The second segment of the External File is for the storing of any network element Profiles that were changed during Network Director execution. This Profile section is

always searched for the presence of a Profile record during network element LOGON processing. The Profile record will be read and kept in main storage on the PDE chain.

Network Director Messages

The Network Director can issue a large variety of messages. The message text itself is maintained in the MSGS CSECT. However, any message modified via the EDIT Command Line command will be maintained in the third segment of the External File. MAINT will access this portion of the file during initialization to locate message text that has been modified for use during execution.

Info Mechanism

The Network Director is also delivered with a large amount of information intended to assist the network users in the use of The Network Director. The final portion of the External File contains the INFO or HELP text records that contain the actual information that will be displayed by the INFO facility.

In addition, the Help segment of the file will contain HELP Index records. These records provide the information necessary to translate INFO Text and Display Direct commands.

The fourth portion of the External File is also the location that any added INFO information will be added if the installation has chosen to extend the INFO information.

Definitions

Each control block or definition that is the target of a SAVE command will have a unique record stored in the External File that contains the individual settings for each operand of the definition block that was saved.

Each record is further categorized by a set "name" and "version", which can be utilized to maintain several sets of definitions. Each set can subsequently be referenced by the RELOAD statement to cause the definition to become active within The Network Director.

Access

Each network element subject to AUTHENTICATION validation or internal password validation will have an AIB (Access Information Block) stored into the External File. The AIB maintains information that must be kept between executions of The Network Director (password history, date device last used, etc.).

The AIB can be displayed and manipulated via the SHOW ACCESS command within Network Administration. Overview information about all the AIBs can be obtained via execution of TNDUTIL's DISPLAY command.

Characteristics

All the records within the External File are maintained as variable length records. Each record is stored in a compressed format intended to reduce The Network Director's disk storage consumption.

The compression technique is optimized to the type of data record being stored. The Profile records contains a simple data representation of Profile information. The Codes records contain the message text as a single variable length string with each word (or token) separated by a single blank. Help and Message records are stored as singular long strings of data separated as necessary by X'15's (New Line characters).

Each data record is read from the External File and then expanded as necessary as its use dictates. Profile records typically are translated into a Profile panel. Codes records are placed onto the DMT chain and available for LOG to locate and utilize instead of the contents of the MSGS CSECT. Messages and Help records are normally FORMATted by the appropriate DFB facility and then displayed upon the terminal.

CINIT RU Processing

Whenever The Network Director enters a session with a device, it does so as a result of a session request that is accompanied by a RU (Request Unit) identified as a CINIT RU. This RU has a "user data area" that can be utilized by the initiating LU to provide information to The Network Director. The user data area can also be formatted by another software subsystem that is passing the device to The Network Director (such as another Network Director, or a 3274 emulation package in another computer, etc.).

When establishing a session (SNA BIND or VTAM OPNDST), The Network Director will look at the CINIT RU user data area to evaluate if there is any data present of significance. Since this area is undefined by SNA, The Network Director has defined a control block identified as a SSX that has implied meaning, when present.

If the first three bytes of the user data area have the literal string **SSI** or **SSX** in them, The Network Director will assume that the data buffer also contains a userid, a password, and may contain an "initial command". This SSX format can be utilized to automate the signon between Network Directors or if The Network Director is operating as a security interface for non-IBM systems that are capable of forwarding a userid and password acceptable to the IBM based system.

The CINIT RU also contains a BIND image, which is compared with the prior BIND image for the device known by The Network Director. If they are the same (comparison is up to, but not including the PSERVIC portion), no additional action is taken. If they are different, The Network Director will attempt to utilize the most recent BIND image presented to it.

Inactive Counter Logic

Philosophically, The Network Director is always attempting to maintain a "panel" on all devices it is in session with. As a result, The Network Director will attempt to write to devices that are off, have their keylocks off, or are simply unplugged. When this occurs, The Network Director will receive a non-zero return code from ACF/VTAM (the exact return code is a function of the device type).

To accommodate temporary errors, The Network Director will retry the output operation multiple times in an effort to create the panel on the device. If the failures occur consecutively GLOBALS NETWORK-RETRIES times, The Network Director will place the device on the "inactive list" and will terminate attempts to write to the device. Activation of the device can normally be accomplished by the device operator pressing

any AID key or the SIGNAL key. A Network Administrator can also reactive the device by issuing a RELEASE NET= command for the device.

Iteration Counter Logic

The **Inactive Counter** just discussed identifies NETWORK-RETRIES consecutive unsuccessful attempts to write to a device during a single session. There are several recovery situations where The Network Director will restart the session between the device and The Network Director by CLSDSTing the device and issuing a SIMLOGON for the device. When this occurs the inactive counter is reset to zero.

However, it is possible that the device will not properly restart due to network failures and The Network Director can enter a never ending process of *restarting the session*. The **Iteration Counter** is designed to detect this situation and place the device on the inactive list.

Each time a device successfully enters session with The Network Director, the iteration counter is reset. If the iteration counter reaches five (meaning no successful sessions), then the device is placed on the inactive list. Iteration counter suspended devices are not eligible for automatic release as inactive counter devices are (issue DISPLAY INACTIVE to establish why devices are going inactive).

Timing Related Items

The Network Director contains its own dispatcher that provides timing related services. Several of the internal facilities rely upon the dispatcher for redispatch at appropriate time intervals. Due to the design of the dispatcher and the various internal DFBs, it is possible that an activity (TIMEOUT, DIM, STATUS-INTERVAL, TIME, etc.) may not take effect at exactly the precise instant originally intended.

The timing delay will typically be only a second or two, but depending upon how your system is operating it could be considerably longer. The Network Director will **never** dispatch an event too early. The design of the dispatcher presumes that timing related events that are delayed a few seconds are an acceptable alternative to larger CPU consumption to guarantee a precise timing operation.

Dump Analysis

The information contained in this Section of the manual is intended to provide a general introduction to the reader about how to interpret a system dump produced by The Network Director.

The Network Director does not require that you have the ability to interpret or read dumps of a partition, virtual machine, or address space. The following information is provided as a brief introduction to dump interpretation for those installations with personnel interested in pursuing the contents of a Network Director abnormal termination. Normally, North Ridge Software, Inc. will be involved in resolution of any abnormal termination.

General Information

The Network Director always runs as a conventional application program within the operating system. As such, it should never be producing any type of operating system WAIT states and all dumps should be of the traditional application type partition, virtual machine, or address space abnormal termination dumps.

WAIT states are typical in DOS when operating in any type of a privileged mode or in OS when running under the control of a RB in a authorized mode. If any type of a stand alone dump or DOS WAIT state is encountered, you should follow problem diagnosis associated with the appropriate operating system. The Network Director itself executes in other than problem program state only when required by system components (RACF, SMF, etc). Thus, abends creating system type abends or wait states are minimized.

This section presumes that any dump that is produced will be a conventional application partition or address space abnormal termination dump (DOS PDUMP or OS ABEND). The dump that is produced will be present on the DOS SYSLST file or the OS SYSUDUMP or SYSABEND DD statements. If your installation has any abnormal termination aid packages, they will manage The Network Director's abnormal termination exactly as any other application subsystem.

Initial Problem Identification

You should first establish the point of interruption within the partition or address space. If the PSW points outside of The Network Director's Modules or Control Blocks, the problem may not be associated with The Network Director. You should investigate closely whether the abend is within The Network Director or, perhaps, outside of it and still within the address space, virtual machine, or partition. This will be the case if an abend has occurred within a VTAM executable Module that is dispatched within The Network Director's addressable area.

Assume that the PSW does point within The Network Director. You should be able to subtract R10 from the PSW address (rightmost word of the PSW) and compute an offset into a Network Director Module. The Module will be identified by simply locating the address within the dump pointed to by register 10 (the standard base register).

Locating Control Blocks

As discussed in the previous section, several of the general purpose registers will be addressing various key Network Director Control Blocks. R12 should point at the PDA. If a DFB task is in control, R11 will point at it. Remember that the first three bytes of each Network Director Control Block contains the three byte identifier of the Control Block. The address of the last dispatched DFB is also stored at PDADDFB, which may assist if R11 has been destroyed.

Once you have located the current DFB, you can identify the path through The Network Director's Modules by following the WAE chain. This chain is anchored in the DFB at DFBWAE and will contain the exact path that was followed to arrive at the currently executing Module. Each WAE's R10 value will point at the entry point of the Module associated with the WAE. Additionally, the WAE R14 value will be the address within the Module that will receive control when the lower level Module completes and returns.

Each DFB is also named and identified via the DFBTYPE field. This will allow you to quickly generalize the DFB's function (Message Facility, Selection, Network Director, etc). If the problem that produced the dump is *load introduced*, you should map the entire DFB chain to identify the types and volume of DFB activity that is active within The Network Director.

Save Area Chaining

Since The Network Director uses R13 to address the current WAE, R13 will point at the current save area when The Network Director is in control. If a non Network Director module is in control, its R13 will normally point at its save area. At R13 plus 4 bytes should be the address of the previous save area. If you follow (back chain) the save areas far enough, you will eventually arrive at a Network Director WAE.

Once the WAE that was last in effect has been located, you can follow the WAEs back to the DFB and identify the function that was in control when the external routine was entered.

External DCBs/DTFs and ACBs

The Network Director dynamically builds all external DCBs, ACBs, and DTFs in the CB storage pool. These operating system control blocks will be pointed to from the PDA. Their condition at abend can be determined by locating them and interpreting their appropriate fields according to the operating system rules.

Control Block Debugging Map

The following pictorial representation provides a quick view of the key control blocks and their general connections. These are not the only control block linkages that you may have to investigate, but they provide the basis for establishing the activity occurring within The Network Director.

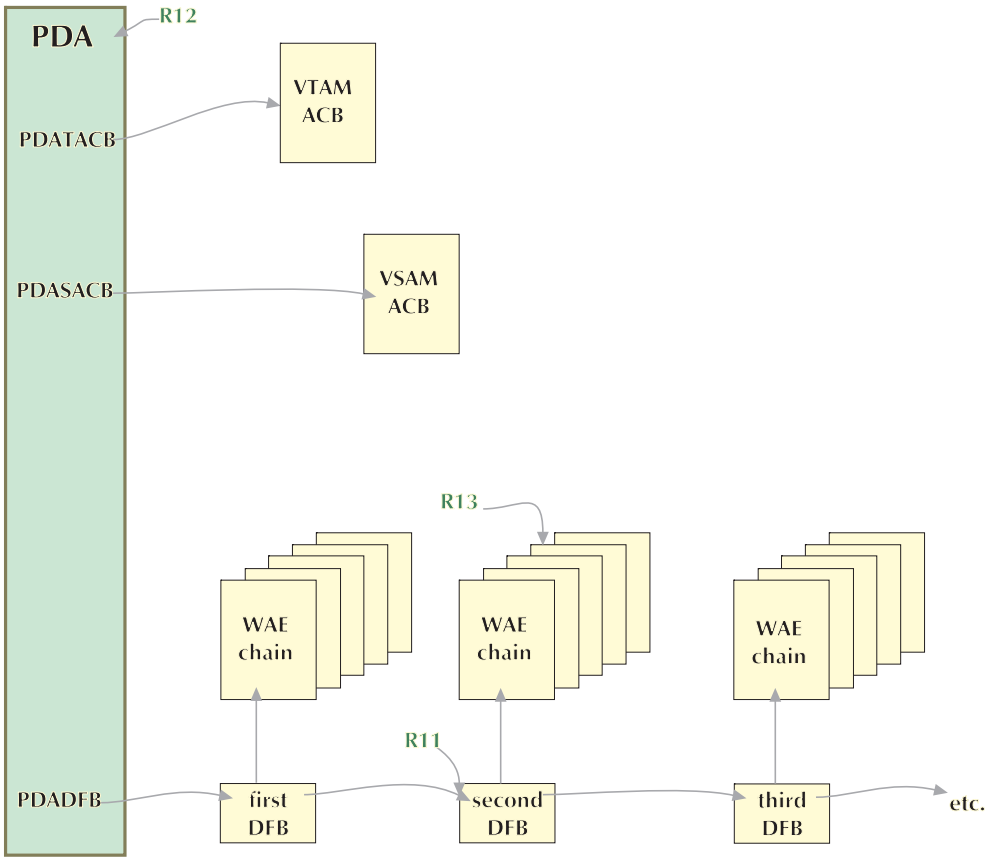

Figure 8. Control Block Debugging Map

The Section on Control Blocks in this manual will assist the reader in identifying additional items of use during dump analysis. Additional information associated with problem reporting in available in Problem Reporting within The Network Administrator's Guide.

VTAM Characteristics

The Network Director runs as a standard VTAM Application Program and makes heavy use of the documented VTAM Application Program Interface. This section of the manual discusses the characteristics and implementation of The Network Director as it applies to the general VTAM interface.

General Characteristics

Each VTAM application subsystem has several decisions to make about how to interact with VTAM and its Application Program Interface. The major points that any VTAM subsystem must address are listed individually in the following paragraphs. The following major points also contain a quick statement of how The Network Director has chosen to address the specific issue.

Single Thread or Multithread

It is possible to construct a VTAM application subsystem that processes only a single logical unit (terminal) at a time. All operations within the subsystem are suspended until the operation outstanding against a single device are fully complete.

The Network Director has chosen to implement a fully Multithread facility. The DFBs represent independent pieces of work (or actions) that are internally managed by DISP. This mechanism allows The Network Director to completely overlap unrelated activity within the VTAM network as well as within The Network Director.

Synchronous or Asynchronous

This basically describes the usage of specific options on VTAM SEND and RECEIVE macros (OPTCD=SYN). It allows the VTAM subsystem to imply its single or multithread characteristic to VTAM. Synchronous operations are typical of single thread subsystems.

The Network Director is an asynchronous implementation (OPTCD=ASY) in all areas that asynchronous activity can be utilized. This allows The Network Director to schedule VTAM operations independent of completion of other network activity. The VTAM OPEN and CLOSE requests are synchronous activities that are not under the control of The Network Director. The SETLOGON START issued to initiate terminal sessions is also synchronous for the same reason.

ECB Posting or RPL Exits

VTAM notifies the subsystem of asynchronous completion of network activity in two manners. First, an RPL Exit can be used to schedule a portion of code that will interrupt other mainline code. As an alternative, VTAM can directly POST a subsystem specified ECB.

RPL Exits are generally easier to implement and code, but can interfere with throughput in an active network. ECB posting allows greater control of the order in which work will be accomplished within the subsystem.

The Network Director utilizes ECB posting in all locations except in the NSI (which are generally single terminal related and not throughput oriented) and the SIMLOGON and INQUIRE requests (which require RPLEXIT specification during certain ERP situations). All dispatching decisions are made by DISP based upon the completion status of DFB related ECBs.

Receive Any or Continue Specific

To provide a convenient manner to manage logical units, VTAM provides multiple manners with which the subsystem can receive information of the network. The subsystem can receive input from any device (RECEIVE OPTCD=ANY) or from just a specific device (RECEIVE OPTCD=CS).

The Continue Specific mode is generally simpler to construct, but can require a large number of active RPLs to manage a significant sized network. Receive Any requires less active RPLs, but can have throughput problems unless input is disposed of quickly (meaning the RECEIVE OPTCD=ANY must be reissued to receive the next input from the next terminal).

The Network Director makes extensive use of the RECEIVE OPTCD=ANY mechanism to keep the number of active VTAM RPLs under control and will only change to Continue Specific in the case of multiple inbound RUs for a single input from a single device (inbound RU chaining). The INPUT Module dispenses incoming messages quickly by generating other DFBs to handle the work associated with the input transmission. Each RECEIVE ANY RPL is set to receive any type of input from the LU sessions in progress (DFSYN, DFASY, and RESP type inputs).

VTAM Control Blocks

The Network Director makes heavy use of the fields within the VTAM Control Blocks. The primary control blocks are the ACB, NIB, and RPL.

These control blocks are mapped within The Network Director via operating system independent macros identified as TNDACB, TNDNIB, and TNDRPL. These are not the actual VTAM Control Blocks. They are pseudo control blocks used within The Network Director and will typically map one for one to a real VTAM Control Block. All modules within The Network Director reference directly the pseudo Control Blocks. When TNDVTAM receives control to actually issue the command represented by a particular RPL, it moves the appropriate pseudo Control Blocks fields into the "real" VTAM control blocks (that were previously GENCBed during initialization).

This pseudo control block mechanism allows the fields that are modified to be placed into the real control block within a single routine (eliminating the need to use completely different object decks in the different operating systems). The fields within the pseudo control blocks are named identically to those in the actual control blocks. This allows the Source code to remain readable in a conventional VTAM application subsystem manner.

Processing of Abnormal Situations

The Network Director deals with all terminals through their defined SNA BIND image (collected by The Network Director during initial LOGON). Depending upon the type of device and communication vehicle, some devices will normally transmit other than normal (CONTROL=DATA) type of input. This is as defined by SNA protocols and The Network Director attempts to respond to these situations as dictated by SNA.

The manner in which The Network Director responds is a function of what information is presented to it and how.

LUSTAT and Sense Input

The Network Director will normally receive Sense information from a LU along with a LUSTAT RU. When this occurs, The Network Director will always produce appropriate messages into the LOG which can be viewed and interpreted. Additionally, The Network Director will respond to two unique sets of Sense code combinations.

The first combination will be handled as if the LU had struck the CLEAR key instead of providing Sense. The Sense codes currently included in this category are:

Sense	*Meaning*
08 27	Retry Requested
08 2B	Presentation Space Integrity Lost

Figure 9. Simulated CLEAR keys (Sense codes)

The second set of Sense codes will cause The Network Director to immediately terminate the session (VTAM CLSDST). The LU should be returned to The Network Director when it becomes available again (through the standard LOGAPPL mechanism or through a queued SIMLOGON issued by The Network Director).

Sense	*Meaning*
08 00	General Path Error
08 02	Intervention Required
08 07	Resource Not Available
08 1C	Function Not Supported
08 2D	LU Busy
08 2E	Intervention Required
08 31	Power Off - Test Request

Figure 10. Sense codes causing CLSDST

CANCEL

When The Network Director receives a SNA CANCEL, it will immediately mark the associated DFB (if any) for abnormal termination and will issue a positive response to the LU for the CANCEL. The Network Director will then wait for the SLU to initiate the next activity.

Quiesce at End of Chain (QEC)

When a QEC is received, The Network Director will place the associated ANE into a HELD status and will immediately send a positive response.

Release Quiesce (RELQ)

A RELQ command causes The Network Director to remove the LU's associated ANE from the HELD status. This enables the ANE for additional activity.

RPL Slowdown Logic

During initialization, The Network Director normally receives a significant number of devices that must be initialized with a proper panel. This initialization process can be a significant consumer of resources within the CPU in both Network Director and ACF/VTAM terms. Since The Network Director will allocate RPLs as necessary, it is entirely possible that The Network Director will cause ACF/VTAM capacity difficulties and NCPs slowdown difficulties.

To address this, the RPL-MAXIMUM operand has been provided as a *governor* for the amount of concurrent work that The Network Director will initiate at any given point in time within the Access Methods. Once this number of active RPLs has been established, The Network Director will begin queueing additional work within The Network Director. The dispatcher will continue dispatching only DFBs that have already begun operation. Once the number of active RPLs is below the RPL-MAXIMUM value, the slowdown condition will be eliminated and The Network Director will begin normal dispatching again (beginning with any work that may have been queued).

Care should be exercised that the RPL-MAXIMUM number not be set so low that The Network Director is incapable of processing peaks that may occur from time to time (a major subsystem going down, etc.). Careful review of a LOG listing during initialization will provide an indication of the frequency of RPL slowdown processing.

RPL Held Logic

During execution, The Network Director will from time to time receive a request to terminate a session associated with a device that has pending work (a DFB attached). LOSTERM and RELREQ exits are examples of VTAM exits that may cause this to occur. Invalid or unexpected return codes to normal VTAM activities may also create this situation.

When this occurs, The Network Director will internally abend the DFB. This amounts to little more than control block cleanup, but has a side effect if a pending VTAM operation had been scheduled. If there is an allocated RPL that is not active, the RPL is simply freed by The Network Director back to the pool of RPLs available for usage. However, if the RPL is active to ACF/VTAM, The Network Director will mark the RPL HELD and will not free the RPL back to the pool (it cannot, since the RPL is still being "used").

Subsequently, The Network Director will recover the held RPLs by issuing CHECK against them. This collection logic is executed about once an hour within The Network Director and is intended to keep the RPL allocation routines at an appropriate value. This is particularly important as the number of active RPLs can impact the performance of the system as a whole (active VTAM RPLs, Network Director RPL-MAXIMUM, etc.).

BIND Processing

As a ACF/VTAM subsystem, The Network Director makes extensive utilization of the bind image associated with the devices within the VTAM tables. This bind image is retrieved the first time a device enters a session with The Network Director and is retained until termination or a DELETE command is issued.

The BIND image is mapped via usage of the IBM provided DSECT ISTDBIND and will dictate to The Network Director the type of terminal. For this reason, it is important that the BIND image properly represents the device. Some subsystems do not always make use of the VTAM BIND image (IMS, CICS, etc.) and you may not always be aware of an improper VTAM defined bind image.

If you suspect the BIND image is incorrect or would simply like to see what it contains, you can locate it by entering two Network Administration commands.

1. Enter DUMP NIB=xxxxxxxx (where xxxxxxxx is the LU name for the device)
2. Enter DUMP 123456 (where 123456 is the address located at offset X'28' in the NIB)

This will produce a hexadecimal dump in storage of the bind image that The Network Director retrieved from ACF/VTAM via INQUIRE SESSPARM. Evaluate it by comparing it to the ISTDBIND DSECT or using the appendices in the VTAM Resource Definition manuals.

READ PARTITION QUERY

If the BIND image for the device indicates that it is capable of extended data stream (EDATS), The Network Director will automatically issue a 3270 Write Structured Field command with a READ PARTITION QUERY command in it to retrieve the USABLE AREA reply (if WSF=YES is in effect for the device). This reply will indicate to The Network Director what the device believes is the current screen size. If the USABLE AREA reply indicates a screen size that differs from the BIND image The Network Director retrieved from ACF/VTAM, then The Network Director will adjust the BIND image utilizing the USABLE AREA reply values and restart the session with the device (utilizing the modified BIND image).

This approach means that devices capable of changing their screen size at the device (3180, 3192, etc.) without informing ACF/VTAM may do so and The Network Director will respond to the setting the terminal operator has chosen.

The Network Director will utilize the Write Structured Field command each time the device enters a session with The Network Director (returns from a subsystem, powers on, etc.). Thus, to get The Network Director to accept a new screen size, the device will normally have to power off and back on, which causes The Network Director to restart the session and discover that the screen size has been changed.

The Network Director also scans the response from the Read Partition Query for the COLOR structured field. If present, this indicates to The Network Director that the device is capable of the extended field orders, which allow The Network Director to use the SFE orders to set the extended colors and attributes). If not present, The Network Director will assume the device is not capable of the extended field support.

BIND Presentation Space Specification

The presentation space size (PSERVIC byte 11) has historically contained X'00', X'01', X'02', X'7E', or X'7F'. These specifications have controlled what the primary and alternate screen sizes are for the device. The Network Director automatically accepts these specifications and will (for Extended Data Stream devices) automatically query appropriate devices to determine if the PSERVIC is proper or not (WSF=YES).

If your installation uses a PSERVIC value of X'03', it indicates that the VTAM host application (in this case, The Network Director) should issue a Write Structured Field Query (WSF Q) to ask the device what screen sizes it is operating with. The Network Director supports this type of PSERVIC specification on devices that will also honor the PSERVIC.

When the Presentation Space specification in PSERVIC is not X'03', The Network Director will restart the session (UNBIND and BIND) after it has modified the BIND image to the values returned from the WSF Q. This requirement to restart the session is not present for devices that are type X'03's. For this reason, NRS recommends that devices that will be changing or configuring the screen size at the device should be generated with a type X'03' in PSERVIC byte 11. This will produce the proper result within The Network Director (screen size support) and will also reduce the amount of session initiation activity required to support the device.

An example of the MODEENT definition for a SNA 3290 is:

```
D329003  MODEENT LOGMODE=D329003,FMPROF=X'03',TSPROF=X'03',
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',
               RUSIZES=X'8787',
               PSERVIC=X'028000000000000000000300'
```

Figure 11. Sample MODEENT Definition for WSF Q BINDs

Storage Estimates

The following information is a discussion of The Network Director's storage requirements and provides insight for the reader into how storage is allocated and utilized by The Network Director. Naturally, the actual storage consumption will vary from installation to installation dependent upon usage characteristics, but the following items are intended to provide a basis for accurate computations.

Basic Storage Concepts

The Network Director dynamically allocates all storage required during execution from dynamic storage. This is OS GETMAINed or DOS GETVISed storage (GCS is OS GETMAIN). The Network Director's load module or phase is essentially reentrant (not self modifying). Thus, wherever possible, all storage is managed outside of the nucleus itself.

The Network Director arranges all obtained storage into 3 **subpools**, called the MX (MiXed), CB (Control Block), and TP (TeleProcessing) pools. Each element on the chain is mapped via a GSA (General Storage Area) prefix, which contains information about the individual storage segment (the size for each pool element is set via GLOBALS STORAGE-POOLS).

Once storage is allocated from the operating system, it is slowly released (FREEMAIN or FREEVIS) under control of the STORAGE-BALANCE operand. Thus, this discussion will deal with **working set** and **maximum requirements**. Working Set discusses the amount of storage generally referenced by The Network Director when simply operating normally with an average amount of work requirements (operator's hitting ENTER, etc.). Maximum Requirements are storage requirements necessary for The Network Director to process peak period demands (typically at system initialization or after IPL).

The WS (Working Set) requirements are generally dictated by the size of the network (number of LUs), the number of timed events being managed, and the number of concurrent events within the network. The MR (Maximum Requirements) are dictated by the number of devices and the types of screens that will be delivered to The Network Director at the same time. This typically occurs only during the initial start up or initialization process, but can also occur if a major subsystem terminates in the middle of a day.

Syntax

For the purposes of this discussion, the following syntax rules are being followed:

1. All Network Director storage items are identified by 3 byte ids. Generally, these correspond to a specific control block within The Network Director, but can also be other combined items (see Storage Definitions). E.G. ADB indicates the number of bytes required for an Application Definition Block.

2. Any 3 byte id surrounded with parends indicates that the length of the storage item should be multiplied by the number of elements associated with the storage item. E.G. (IDE) indicates that storage consumption is the length of the IDE times the number of INITIAL-DATA= operands specified.

Storage Definitions

The following 3 byte storage ids are used within The Network Director.

Id	Size	Pool	Name (Related Operand)
AAE	16	CB	Authorized Application Element (APPLICATIONS= values)
AAI	76	CB	Application Action Items (ACTIONS= values)
ACB	96	CB	Access Control Block
ADB	156	CB	Application Definition Block (APPLICATION)
AIE	24	CB	Application Interval Element (TIME= & DAY= values)
ANE	268	CB	Active Network Element
ATE	56	CB	Authorized Terminal Element (TERMINALS= values)
AUE	56	CB	Authorized User Element (USERS= values)
DCB	100	CB	Dataset Control Block
DCE	80	CB	Device Command Element (stacked commands)
DFB	156	MX	Dispatchable Function Block (internal work units)
DMT	96	CB	Director Message Text
DSA	??		ANE+NIB+security+(SSE)+(DCE)
EBA	4096	MX	EFR buffer area (VSAM I/O buffer)
EDL	96	MX	Edit Data Line
EDT	1152	MX	each dispatchable task
EFR	64	CB	External File Record
EWA	512	MX	Edit Work Area
GDB	120	CB	Group Definition Block (GROUP)
HDE	36	CB	Help Data Element

Id	Size	Pool	Name (Related Operand)
HIX	100	CB	Help IndeX
IDE	120	CB	Initial Data Element (INITIAL-DATA= values)
LSZ	??	CB	80*number of record images (LOGO= value)
MDE	72	CB	Message Data Element
MIX	32	CB	Message IndeX
MME	1470	MX	MSG Message Elements
NCB	255	--	Network Control Block
NEL	??	--	Network Element (an active device)
NIB	168	CB	Node Information Block
PDA	4800	CB	Primary Director Area
PDE	196	CB	Profile Data Element
PPE	458	CB	Pending Parameter Element
RPL	168	CB	Request Parameter List
SDB	36	CB	Site Definition Block (SITE)
SMR	250	CB	System Measurement Record
SSE	16	CB	Screen Selection Element
SWA	352	MX	Selection Work Area
TBA	2380	TP	terminal buffer area (estimate for 3270 Model 2)
TDB	152	CB	Terminal Definition Block (TERMINALS)
UDB	136	CB	User Definition Block (USERS)
WAE	256	MX	Work Area Element

Figure 12. Storage Estimation Ids

Definition Statement Storage Requirements

The Network Director's definition statements take up the following storage space:

Statement	Formula
APPLICATION	ADB+(AIE)+(IDE)+(AAE)
GROUP	GDB+(ATE)+(AAE)+(AIE)+(AUE)+LSZ
PROFILE	PDE
TERMINALS	TDB+LSZ+(AAE)+(AIE)+(AUE)
USERS	UDB+LSZ+(ATE)+(AAE)+(AIE)
SITE	SDB+(AUE)

Figure 13. Definition Statement Storage Estimates

Fixed Overhead

The following formula can be utilized to compute the basic fixed overhead size for The Network Director:

```
Executable Code                              500,000

GLOBALS LOGSIZE=                              16,384
PDA                                            4,800
(GLOBALS RPLS= * (EDT+RPL+4089))              10,818
EDT (if GLOBALS CONSOLE=YES)                   1,152
EDT + RPL (if VTAMOPER=YES)                    1,320
2*(EDT+RPL)                                    2,640
EDT                                            1,152
DEFAULT (AAE) + LSZ
GLOBALS TRACE=                                     0
2*ACB                                            414
2*DCB (if OS or GCS)                             200
14*WAE                                         3,584
approximate (DMT) + (EFR)                      5,256
                                              -------
                                              547,720
```

Figure 14. Fixed Overhead Storage

Thus, The Network Director will consume about 450K of virtual storage prior to allocations for local definitions. This overhead is what is generally required to simply operate the nucleus of the system. To compute the remainder of the overhead, you will have to add in the appropriate values for the definitions (the parameter statements) and estimates for the number of concurrent internal tasks and the total number of network elements (devices) being managed by The Network Director.

Sample Computation

Assume the following Configuration Parameters:

```
GLOBALS RPLS=2,LOGSIZE=16K,RPL-MAXIMUM=100
*
DEFAULT LOGO=

      10 line logo

LOGO-END
*
APPLICATION    ..............  10 applications defined
*
GROUP  GROUP1,LOGO=

       5 line logo

LOGO-END,APPLICATIONS=( 5 applications )
*
GROUP GROUP2,LOGO=

        10 line logo

LOGO-END,APPLICATIONS=( 10 applications )
*
USERS   ....................... 10 (all referencing GROUPs)
```

Figure 15. Sample Storage Computation Deck

The resulting storage computation would be:

Storage Item	100 devices	1000 devices
Fixed Overhead		
GLOBALS LOGSIZE	16,384	16,384
DEFAULT LOGO=	800	800
APPLICATION definitions	1,560	1,560
GROUP definitions	1,440	1,440
USER definitions	1,360	1,360
Load Module/Phase	400,000	440,000
Sub-total:	421,544	421,544
Control Blocks (CB Pool)		
(DMT)*50	5,200	5,200
(RPL)*RPL-MAX	16,800	16,800
(ANE)*NEL	26,800	268,000
(NIB)*NEL	16,800	168,000
(SSE)*#selections	11,800	118,000
Sub-total:	77,400	576,000
MR Processing		
(EDT)*RPL-MAX	115,000	115,200
(EBA)*(RPL-MAX/2)	268,800	268,800
(TBA)*RPL-MAX	238,000	238,000
Sub-total:	621,800	621,800
Totals:	1,120,744	1,619,344

Figure 16. Sample Storage Computation

NOTE: the EBA computation above is for OS systems only. DOS and GCS environments utilize DOS VSAM, which provides no overlapping of operations. Thus, for DOS or GCS there will only be a single EBA allocated.

Miscellaneous Issues

RPL-MAXIMUM is the key item to control the MR or largest storage requirements characteristic of The Network Director. A reduction in this value will bring on corresponding storage reductions in all area of MR processing, but the RPL-MAX should never be set below GLOBALS RPLS= + 4 (MNTR, PO, and 2 RPLs for general output and ADMIN type requests).

RPLS+8 is a better *lowest value* for normal operations. Thus, RPL-MAXIMUM=10 should introduce a major savings in MR type processing. However, NRS recommends that you start with RPL-MAXIMUM set at its default of 100 and experiment at your installation to establish the best position for this governor.

DOS and VM installations should also remain aware that VSAM related operations by The Network Director are essentially single threaded. This is a function of how VSAM operates within these operating system environments and will impact the storage estimates discussed in this section.

Simple elimination of the External file will also net a large reduction in MR type processing (all EBA requirements will be eliminated).

These estimates do not take into account any requirements that VTAM itself may have in the partition or address space. Other control blocks allocated by the operating system

will continue to be allocated and should be provided for. Additionally, these estimates do not make reference to ACF2, RACF, or TOP-SECRET requirements, which are also in addition to The Network Director's requirements.

The information presented here is current as of the date of this publication. During enhancement activities, The Network Director may increase its load module component, control block sizes, or allocation techniques, please contact NRS to explain any difference you may notice.

Installation Exits

The Network Director offers a large variety of parameters within the Configuration Parameters and other sources to tailor the operations of The Network Director. The Network Administrator can also dynamically modify the attributes of many network elements.

However, it is still possible that an individual installation may wish to modify the characteristics of a particular Network Director facility. Modification can usually be achieved through the use of one or more of The Network Director Installation Exit routines. These routines receive control during The Network Director's normal processing and will function as an integral portion of The Network Director.

Definition

A Installation Exit is a 370 ASSEMBLER routine that is dynamically loaded or linkedited with The Network Director and receives control at one or more specific points within the logic associated with processing a network element (terminal, user, group, request, etc). The Installation Exit routine will have the opportunity to modify the action that The Network Director will take, modify Network Director control blocks, or to simply track the type of activity occurring within the logical network.

Each Installation Exit must follow all established Network Director conventions as discussed in this manual. Each exit will receive control at different points within The Network Director and must also follow any specific or implied rules associated with the process underway at the time of entry.

All exits must be written in 370 ASSEMBLER code and should be reentrant. The Network Director's source library and macros may be used to assist in establishing proper linkage (DFB type, etc). North Ridge Software, Inc. will provide assistance relative to the exit's interface to the rest of The Network Director, but it is the installation's responsibility to insure that the exit does not damage the integrity or the performance of the network.

List of Exits

The Installation Exits currently available within The Network Director are all prefixed with the standard **TND** and are further described as follows.

Exit	*General Usage*
EXT01	Generalized Security
EXT02	Message Facility Actions
EXT03	INPUT Preview
EXT04	Event Logging
EXT05	Network Request Processor Preview
EXT06	Operator Input Preview
EXT07	Parameter Validation
EXT08	Network Administration Preview
EXT09	Profile Audit
EXT10	Storage Management
EXT11	Pre Selection
EXT12	Post Selection - CLSDST PASS
EXT13	Command Line Preview
EXT14	Pre dispatch
EXT15	Individual Selection Review
EXT16	Selection Evaluation
EXT17	SMR Preview
EXT18	Selection Screen Review
EXT19	GROUP Assignment Review
EXT20	Input/Output Review
EXT21	LOGON Exit Review
EXT22	Notify/Broadcast Review
EXT23	Alternate Parameter Deck
EXT24	LOGOFF
EXT25	Authentication
EXT26	Post Initialization
EXT27	Pre Termination
EXT28	New Password
EXT29	ACF2 Security System Call
EXT30	Password Check
EXT31	Selection Return
EXT32	Interval Operation Review
EXT33	User Defined DFB
EXT34	Post Security System Call (ACF2)
EXT36	Post Logon Exit
EXT37	Application Selection Panel construction

The term *preview* is used to imply that the Installation Exit will receive control before the logic normally associated with the process underway.

Exit Processing

Network Director exits can be linked with The Network Director's nucleus or dynamically loaded by simply placing them into one of the execution time libraries associated with The Network Director's executable routines (e.g. OS Job Pack Area/STEPLIB, VM GLOBAL LOADLIB).

A few exits will not operate correctly unless they are linked with The Network Director's nucleus. This is because the exit is designed to receive control before The Network Director has established an ability to dynamically LOAD another routine (EXT07, EXT23, and EXT26) or some actions will occur before the Dispatcher is operational (EXT04, etc.).

During initialization, The Network Director will look to see if an exit has been linkedited with The Network Director's nucleus. If it has, the linkedited routine will be accepted and remain in effect. If no exit has been linked, The Network Director will issue an appropriate operating system LOAD request to bring the exit into dynamic storage. If successful, the address pointer for the exit will be placed in the PDA for subsequent usage by The Network Director's internal routines.

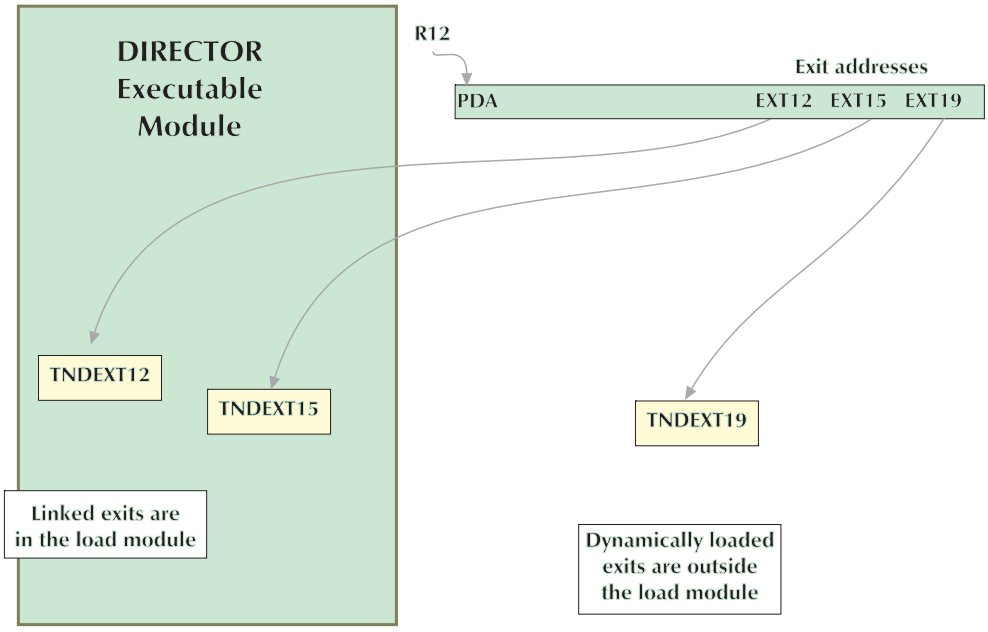

Figure 17. Exit Location

Dynamically manipulating the exit routines (loading and deleting) can be accomplished through the usage of SHOW EXITS (see the *Network Operator's Guide*).

Conventions

In general, any Installation Exit that executes in The Network Director's address space, virtual machine, or partition will receive control with R12 pointing at the PDA, R13 pointing at a standard 18 fullword save area, R14 containing the return address, and R15 containing the Installation Exit's entry point address. Return codes are always passed in R15 and parameter lists are addressed via R1.

It is the exit's responsibility to save and restore the registers when entering and exiting the Installation Exit. The Network Director presumes that the Installation Exit is authorized to become a portion of the operating Network Director nucleus. An abend in the Installation Exit will impact the entire Network Director environment.

DFB task exits must also follow rules associated with The Network Director's dispatcher based environment. R11 will point at the DFB and R13 will address the caller's WAE.

The exits must follow the established dispatcher conventions exactly as any other DFB related CSECT would (TNDSTART, TNDEND, etc). DFB exits must avoid issuing requests that will cause The Network Director's partition, virtual machine, or address space to enter a full operating system WAIT condition. TNDWAIT may be used in most exits to simulate the process of WAIT.

Considerations

GCS/VSAM installations should be aware that VSE/VSAM does **not** restore the registers per OS conventions (R13 will typically be destroyed upon return). Therefore, if you are intending to invoke VSAM from an exit in the GCS environment, you will have to make special provision for the VSE/VSAM characteristics.

NRS also recommends that your installation exits utilize the TNDSTART macro for CSECT prologue so that the DISPLAY EXITS command will produce output that is meaningful.

To get an idea of how the exits interrelate, consult the sample output LOG listing in "Appendix A. Sample Exit LOG Listing" on page 223. This listing should provide some insight into how the various exits receive control during execution.

Suggested Exit Code Structure

A skeleton sample exit is:

```
TNDEXTxx TNDTITLE 'SAMPLE EXIT'
*********************************************************************
**                                                                 **
**       THIS IS A SAMPLE OF HOW TO STRUCTURE AN EXIT.             **
**                                                                 **
*********************************************************************
         TNDPDA WXTRN=NO               PRIMARY DIRECTOR AREA
         TNDWAE ,                      WORK AREA ELEMENT
         TNDDFB ,                      DISPATCHABLE FUNCTION BLOCK
         PRINT GEN                     PRINT MACRO EXPANSIONS
         EJECT ,
EXTxx    TNDSTART ,                    MODULE BEGINNING
*********************************************************************
**                                                                 **
**  ONCE TNDSTART IS COMPLETE, OUR REGISTERS ARE SAVED AND THE     **
**  BASE REGISTER HAS BEEN SET UP.  THERE IS A GENERAL REENTRANT   **
**  WORK AREA AVAILABLE AT WAEWORK OR MORE STORAGE CAN BE          **
**  OBTAINED VIA TNDALLOC AND FREED VIA TNDFREE.                   **
**                                                                 **
         NRSREGS LIST                  WHAT IS ALLOCATED???       **
*********************************************************************



         other instructions as required



         TNDEND R15=R15                END OF ROUTINE AND RETURN CODE
         LTORG ,                       PUT LITERALS HERE
         END ,                         END OF ASSEMBLY
```

Figure 18. Installation Exit Structure

The distribution tape contains several machine readable samples of exits. Members, books, or files named TNDEXTxx can be utilized as a start point. TNDISSUE can be utilized to place information into The Network Director's LOG queue for viewing by valid Network Administrators.

If you have additional questions about how to write exit code or general structural questions, contact North Ridge Software, Inc.

All exits should code WXTRN=NO on the TNDPDA macro to avoid *duplicate definition* errors from the assembler when the TNDSTART or CSECT statement is processed (the PDA normally defines the same tag). Coding WXTRN causes the PDA macro expansion to bypass this tag generation.

Individual Exit Descriptions

The following pages of this section describe each Installation Exit, its environment, and the options available to it (when returning to The Network Director).

EXT01 - Generalized Security

Purpose	provides a manner with which the installation may define its own mechanism to check the password (or other identification information) within The Network Director.
Characteristics	runs as a DFB callable Module. It will be invoked whenever a network operator has attempted to identify himself. EXT01 is intended to provide a manner with which an interface to an installation security package not supported by The Network Director can be constructed.
Input	**R1 + 0** address of ANE attempting logon **R1 + 4** address of message area
Output	**R15 = 0** allow logon and proceed as normal to selection **R15 = 4** disallow logon and increment TRIES counter **R15 = 8** disallow logon and place the ANE on the Inactive list **R15 = 12** proceed with other normal security checks and allow their results to control this logon's completion
Comments	EXT01 can be used to search a table, interrogate a file, or invoke another security system. EXT01 is entered after the password check and group assignment has been done, but before the TIME, DAY, MAXIMUM, USER, or TERMINAL checks. EXT01 is intended primarily for those installations that are attempting to replace all of The Network Director's security package interface. To do simple password checking and extended validation, utilize EXT30 instead. Any message provided in the message area will be scheduled to be displayed in the terminal operator's message area. Messages should be placed in this area utilizing TNDISSUE, which will cause the message to be placed into the LOG queue for displaying by the Network Administrator.

EXT02 - Message Facility Actions

Purpose	provides a manner with which specific Message Actions can be checked for authorization. EXT02 can be used to provide a mechanism to insure the network operator is utilizing the Message Facility in the installation's desired manner.
Characteristics	runs as a portion of the MSG DFB. EXT02 receives control immediately after the operator has made a request, but prior to MSG invoking the appropriate module.
Input	**R1 + 0** address of the single byte containing the Message Action code **R1 + 4** address of the message name being operated on **R1 + 8** the ANE address **R1 + 12** address of the possible message area
Output	**R15 = 0** continue with the operation **R15 = 4** reject the operation with an appropriate Network Director error message **R15 = 8** reject the operation with the error message placed in the area pointed to by R1+16. The message may not exceed 60 bytes.
Comments	When forming non standard Network Director error messages, you should create your own prefix error message numbers that can be cross referenced with other Network Director error messages.

EXT03 - INPUT Preview

Purpose	review the input data transmission prior to any Network Director operations have begun.
Characteristics	runs as a portion of INPUT's DFB. It receives control immediately after INPUT has RECEIVED the data from VTAM and prior to INPUT inspecting the contents of the transmission.
Input	**R1 + 0** address of the virtual RPL
Output	**R15 = 0** continue processing the input as normal **R15 > 0** simulate a CLEAR key from the device
Comments	EXT03 can be used to modify the input transmissions that The Network Director is to operate on. Extra care should be exercised in EXT03, as it will be executing within INPUT after a RECEIVE ANY RPL has been satisfied and prior to another being reinstated by INPUT. Thus, any large delays in EXT03 can create throughput problems within the logical network (particularly if RPLS= is set at a low value).

EXT04 - Event Logging

Purpose	monitor information that is entering the LOG file or queue.
Characteristics	runs as a portion of LOG. It can investigate information and messages that are entering the LOG file. Information can be changed or rejected from entry on the LOG queue.
Input	**R1 + 0** the DMT address **R1 + 4** address of the 133 byte area (carriage control in byte 1) of the print line that will be sent to TNDLOG and back to the caller of the LOG processor.
Output	**R15 = 0** continue to LOG as normal **R15 > 0** do not LOG the message (prohibits the placement of the message into the LOG queue, but not necessarily the placement of a message on a terminal screen).
Comments	R11 will point at the DFB in control at the time of the message being issued. It is also the DFB that EXT04 is running under. Any modified message text will be reflected in the output listing (TNDLOG), the Network Administration LOG queue, **and** to the terminal operator.

EXT05 - Network Request Processor Preview

Purpose	screen calls originating from the Network System Interface.
Characteristics	runs as a portion of NRP's DFB. EXT05 may modify the NSI request, reject it, or allow it to proceed. EXT05 receives control immediately after the NRP has been dispatched and prior to any evaluation of the NCB or the NDA.
Input	**R1 + 0** address of the NCB **R1 + 4** address of the ANE
Output	**R15 = 0** continue to process the NSI request **R15 > 0** reject the NSI request (EXT05 should also set a response code in NCBRSP).
Comments	EXT05 will receive control only when NRP is dispatched. If NSI has been disabled (NSI=NO), NRP will never process a request and will not enter EXT05.

EXT06 - Operator Input Preview

Purpose	screen any commands entering The Network Director from the operator's console.
Characteristics	runs as a portion of OPER's or SMSG's DFB. EXT06 has the opportunity to evaluate any input that comes from the operator and to translate it prior to evaluation by The Network Director.
Input	**R1 + 0** address of the input data
Output	**R15 = 0** continue to process the input **R15 > 0** ignore the input operator command (EXT06 should have placed a message on the operator's console explaining why the command is being rejected).
Comments	EXT06 can be also utilized to create additional operator commands by utilizing The Network Director's normal OPER mechanism and intercepting normal OPER processing. Operator messages should be issued via the TNDISSUE macros.

EXT07 - Parameter Validation

Purpose	evaluate each parameter operand and statement for validity within The Network Director.
Characteristics	runs as a portion of the PARMS service processor, which may run as a DFB or as a normal module. EXT07 is entered after each parameter operand has been parsed and prior to it being processed. EXT07 may modify the parameter's value or cause it to be rejected.
Input	**R1 + 0** the PPE address **R1 + 4** address of PARMS WAEWORK
Output	**R15 = 0** continue processing as normal **R15 > 0** reject current operand and use operand's default value (if applicable).
Comments	EXT07 has the opportunity to evaluate each parameter operand prior to PARMS interpreting it. The PPE may be used to evaluate the current operand against previously processed operands. EXT07 will be entered from PARM, DECK, and OPER as necessary. Typically, EXT07 checks the PPE to evaluate if the statement in process is applicable to its logic flow. This *statement screening* is then followed by the logic associated with processing the specific operand.

EXT08 - Network Administration Preview

Purpose	provide additional checking associated with input from a Network Administrator's terminal.
Characteristics	runs as a portion of ADMIN's DFB. EXT08 is entered before ADMIN, DUMP, PO, REPT, or DECK has been dispatched with input and before any evaluation of the message's content.
Input	**R1 + 0** the ANE associated with the Network Administrator **R1 + 4** the address of the input data stream **R1 + 8** the PPE
Output	**R15 = 0** continue processing as normal **R15 = 4** ignore input and reposition the LOG display **R15 = 12** simulate TIMEOUT condition (terminate Network Administration and logoff the user)
Comments	EXT08 can be utilized to implement additional security checking on operations that are being accomplished at Network Administrator terminals. EXT08 is entered twice for each command (once before The Network Director's parser has evaluated the command, and once after). EXT08 can determine the type of entry by looking for PPEPARSD in the PPE (If PPEPARSD is not on, the entry is before the parser. If PPEPARSD is on, the parser has already validated the command).

EXT09 - Profile Audit

Purpose	provide a location to edit any values placed into a Profile by an network user.
Characteristics	runs as a portion of PROF, which runs as a DFB task that interrupts other DFB related tasks. EXT09 receives control after the network user has modified the Profile Control Block.
Input	**R1 + 0** the PDE address **R1 + 4** address of the ANE associated with the network element **R1 + 8** address of a message area
Output	**R15 = 0** continue with the Profile process **R15 = 4** issue the message in the message area to the user
Comments	EXT09 may modify the ANE as it wishes. Typically, EXT09 is used to validate any profile elements that are in logical error. As an example, a specific set of users may be restricted to using only certain printers for message printing. EXT09 may be used to enforce this local installation rule.

EXT10 - Storage Management

Purpose	monitor the activities of STOR during management of the dynamic storage associated with The Network Director.
Characteristics	runs as a portion of STOR, which may run under control of a DFB or as a normal operating system level task. EXT10 receives control prior to STOR actually accomplishing allocation or freeing of any storage.
Input	**R1 + 0** request type. 12 = allocate storage. 16 = freeing storage. **R1 + 4** address to free or number of bytes to acquire **R1 + 8** storage pool. 4 = CB. 8 = MX. 12 = TP.
Output	**R15 = 0** continue as normal with allocation/freeing operation **R15 > 0** reject request and abend either the DFB making the request or The Network Director
Comments	Setting R15 greater than 0 will create abnormal termination of The Network Director or one of its DFB tasks. This should not be done unless you are specifically diagnosing a problem associated with the internals of The Network Director.

EXT11 - Pre Selection

Purpose	screen selection requests originating at network terminals.
Characteristics	runs as a portion of SEL's or ASK's DFB. EXT11 can modify the request prior to normal selection logic processes. EXT11 receives control before SEL or ASK has evaluated the meaning of the input request. Thus, EXT11 may modify the selection made at the terminal.
Input	**R1 + 0** ANE address **R1 + 4** address of message area
Output	**R15 = 0** continue with normal processing **R15 = 4** rebuild the screen **R15 = 8** indicate failed logon **R15 = 12** indicate failed logon - set ANE Inactive **R15 = 16** indicate failed logon - ID in use **R15 = 20** logon incomplete
Comments	To issue a message, set R15 equal to zero and place the message text at the address specified by the second parameter. EXT11 is preferred over EXT03 for editing most input Selection situations because EXT11 runs as a portion of SEL's DFB. There will be a single DFB for each selection process in progress. This allows greater throughput characteristics than constant evaluation under INPUT's DFB.

EXT12 - Post Selection - CLSDST Pass

Purpose	review the decisions made by The Network Director associated with the transfer of a terminal to a target subsystem.
Characteristics	runs as a portion of SEL's DFB. EXT12 receives control after The Network Director has evaluated the network user's request. SEL will have primed the RPL and NIB with all associated information for the transfer and for SSI's operation (when in effect).
Input	**R1 + 0** the ADB address being requested **R1 + 4** the ANE address **R1 + 8** the message area **R1 + 12** the SWA address
Output	**R15 = 0** continue with the CLSDST PASS operation **R15 = 4** reject the transfer and update the message area only with the provided message. **R15 = 8** discontinue the transfer and rewrite the entire Application Selection Panel including the provided message, as appropriate, in the message area)
Comments	EXT12 can be used to control the characteristics of data that is passed to SSI as well as a final audit for any Network Director terminal transfers.

EXT13 - Command line Preview

Purpose	Evaluate the input entered from a command line prior to The Network Director's parsing and evaluation process.
Characteristics	Executes as a portion of SEL's DFB. EXT13 receives control after SEL has extracted the command, but prior to operating on it.
Input	**R1 + 0** the ANE address **R1 + 4** the entered command (67 bytes long) **R1 + 8** the SWA address
Output	The exit can modify the SWA or ANE to modify the incoming request.

EXT14 - Pre dispatch

Purpose	remain involved during each DFB dispatch
Characteristics	runs as a portion of DISP, which is the primary operating system level Module during The Network Director's execution. EXT14 receives control immediately before DISP dispatches a DFB.
Input	**R1 + 0** posted DFB that is about to dispatch
Output	**R15 = 0** continue with dispatch **R15 > 0** do not dispatch this
Comments	DISP is the primary controlling Module for distribution of the CPU resource within The Network Director. EXT14 can be used to monitor the DFBs being dispatched by DISP. This exit will be entered as a traditional operating system task and not as a DFB related task. Since EXT14 is entered for each DFB dispatch, you should minimize the activity accomplished in EXT14 as it will directly impact the dispatch time for the DFBs.

EXT15 - Individual Selection Review

Purpose	review whether a particular selection should be presented to a specific network element
Characteristics	runs as a portion of BUILD, which is the portion of the Selection process responsible for creating the Application Selection Panel itself. EXT15 receives control immediately after BUILD has decided to include an APPLICATION in the user's Application Selection Panel. EXT15 may evaluate the decision and allow BUILD to proceed or to bypass the selection and locate the next application to include.
Input	**R1 + 0** the address of the APPLICATION name about to be included. This is the name specified on the associated APPLICATIONS= operand of a USER, GROUP, TERMINAL, or DEFAULT statement. **R1 + 4** the address of the VTAM LU Name of the terminal that will receive the Application Selection Panel **R1 + 8** the address of the Id: of the USER logged onto the terminal. This will be blank if no user has logged onto the device. **R1 + 12** the address of the address of the ACMCB (for ACF2) or the ACEE (for RACF or TOP-SECRET). If this address is zero, the involved terminal or user is not currently being managed by the security subsystem.
Output	**R15 = 0** include this as a selection **R15 > 0** bypass this selection and go on to the next

EXT16 - Selection Evaluation

Purpose	controls various elements of an individual selection on the Application Selection Panel.
Characteristics	Operates as a portion of BUILD, during the Application Selection Panel formatting phase. This exit is similar to EXT15, but receives control later in BUILD's buffer creation process. EXT16 can modify the cosmetics associated with the individual selection or control which APPLICATIONS are "hidden".
Input	**R1 + 0** the address of the storage location where the selection has been built **R1 + 4** the ANE address **R1 + 8** the ADB address **R1 + 12** the SSE address **R1 + 16** the SWA address
Output	**R15 = 0** continue with processing (title may have been changed) **R15 > 0** bypass this selection and go on to the next
Comments	The first parameter points at a storage location that contains the first character of the APPLICATION TITLE in it. The virtual SBA orders will immediately precede it in storage and the Status Area will follow the title text.

EXT17 - SMR Preview

Purpose	review the data record about to be produced into the external accounting medium (OS SMF, DOS or OS sequential media, or VM DIAGNOSE).
Characteristics	runs as a portion of the DFB issuing the accounting write operation. EXT17 receives control immediately prior to STAT initiating the write to the external recording medium. EXT17 may look at the data record and may modify it or elect to have STAT simply bypass writing the accounting record.
Input	**R1 + 0** the address of accounting record **R1 + 4** type of accounting record 4 .... SMR 8 .... TNDSAR 12.... SAR
Output	**R15 = 0** continue with the write to the external recording medium **R15 > 0** do not issue this write and completely bypass recording this specific EVENT record
Comments	EXT17 can be used to tailor the specific SMR types that are produced onto the external medium.

EXT18 - Selection Screen Review

Purpose	review the activities that were accomplished by The Network Director to create the Application Selection Panel.
Characteristics	runs as a portion of the DFB that has created the output Application Selection Panel that is about to be sent. EXT18 can review the SSE chain, scan the actual terminal buffer, etc.
Input	**R1 + 0** the address of ANE **R1 + 4** the address of SWA
Output	**R15 = 0** continue with the write normally **R15 = 4** continue with the normal output write operation, but include the message placed at SWAMSG **R15 = 8** do not send the output panel. Instead, invoke the input selection processor to process input (like a pfkey value, or possibly queued as a DCE). The exit may provide input or modify what was received by placing the information into SWACMD. **R15 = 12** do not send the output panel and terminate the DFB with no further action.
Comments	Use the flag ANELOGD in ANESTAT to determine if the output is for a user that has successfully logged on or not

EXT19 - GROUP Assignment Review

Purpose	review the GROUP that The Network Director is about to assign to a network element that is signing on.
Characteristics	receives control as a portion of the DFB processing a logon request after the basic userid and password combination has been validated, but before The Network Director has located the GROUP= operand.
Input	**R1 + 0** the address of the GROUP name **R1 + 4** the address of the ANE **R1 + 8** the address of the ACMCB (ACF2) or ACEE (RACF, TOPSECRET), if the security system has already been called (see the Comments below).
Output	**R15 = 0** continue with the GROUP lookup. EXT19 can modify the 8 character value accessed via the first parameter to change the GROUP name being assigned. **R15 > 4** bypass all GROUP assignment for this user (this produces a result exactly as if no GROUP= operand was provided for this USER).
Comments	EXT19 is typically utilized to modify the GROUP being assigned to the USER. It can also be utilized to extract a GROUP name from an alternate location or derive one from RACF, ACF2, or TOPSECRET related information. To make use of the ACEE or ACMCB, you must specify GROUP=RACF or GROUP=ACF2. This is required to get The Network Director to call the security system for password validation **prior** to invoking EXT19. EXT01 receives control **after** EXT19 has been entered.

EXT20 - Input/Output Review

Purpose	evaluate the input or output data stream being logically processed by The Network Director.
Characteristics	receives control as a portion of the DFB processing a TNDTERM request (SCRN).
Input	**R1 + 0** identifies the type of data stream (4 = output, 8 = input) **R1 + 4** the address of the data stream **R1 + 8** the length of the data read or to be written
Output	None.
Comments	EXT20 can be utilized to monitor the types of data stream being processed by The Network Director. It could be used to do optional translation of the contents, etc. The data stream is the "physical I/O buffer" addressed by ANEIO, which implies that the buffer have the actual 3270 orders in it (not The Network Director translated ones). EXT20 may modify the data stream and the length. Updating the parameter list itself will cause The Network Director to propagate the data address and length back into the proper fields within The Network Director.

EXT21 - LOGON Exit Review

Purpose	inspect the incoming CINIT RU being processed by the VTAM LOGON exit.
Characteristics	receives control as a portion of the VTAM asynchronous LOGON exit. For OS systems, this is a portion of the IRB scheduled by VTAM and care must be exercised not to introduce WAITs, etc. EXT21 cannot generally utilized TNDISSUE or any operations that may imply terminal input or output. EXT21 can be utilized only to control whether The Network Director will schedule an internal element (DFB) to process the CINIT that has been received.
Input	**R1 + 0** address of the LU name **R1 + 4** the address of the CID for the session **R1 + 8** the CINIT RU received
Output	**R15 = 0** continue with normal processing (attempt to BIND) **R15 > 0** ignore the device (essentially, act as if no CINIT RU was received).
Comments	Since EXT21 is given control from the LOGON exit, you should be fully aware of the implications of a VTAM LOGON exit and The Network Director's processing associated with it prior to implementation of EXT21.

EXT22 - Notify/Broadcast Review

Purpose	review the asynchronous output being delivered and decide whether to allow it to proceed, alter it, or cancel it exit.
Characteristics	receives control as a portion of the NTFY DFB task or whenever the output processor (SCRN) is including a BROADCAST message with the delivery of another panel. The NTFY task is scheduled whenever an asynchronous event has occurred (I.E. a BROADCAST command from a Network Administrator or an APPLICATION status change)
Input	**R1 + 0** address of the MDE (for BROADCASTS) or the address of the ADB (for APPLICATION status changes) **R1 + 4** the address of ANE **R1 + 8** address of the data stream being sent
Output	**R15 = 0** continue with normal processing (output the data stream) **R15 > 0** bypass this output
Comments	The data stream can be modified in place or, if invoked under a NTFY DFB task, replaced with one of the exit's choice. To obtain a larger buffer, use TNDALLOC and place the new data stream's address in ANESCRN and its length in ANELSCR.

EXT23 - Alternate Configuration Parameters

Purpose	provides an alternative for the storage and presentation of the initial Configuration Parameters.
Characteristics	receives control prior to any DFB task activity and is therefore a portion of the job step operating in a standard 370 program environment. The PDA is available, but other control blocks may or may not be allocated (depending upon the contents of the standard source for the Configuration Parameters (SYSLST, or TNDPARMS). EXT23 can be utilized to store the Configuration Parameters in a location other than normally supported by The Network Director.
Input	**R1 + 0** address of the PPE
Output	**R1** contains the address of the 80 byte record image **R15 = 0** continue with normal processing (parse the statement addressed by R1) **R15 = 4** logical end of file
Comments	EXT23 does not receive control from the dispatcher's environment and therefore should not make use of TNDWAIT. Any waits implied by operations in EXT23 will hold up the initialization of The Network Director.

EXT24 - LOGOFF

Purpose	provides a location for reviewing the LOGOFF event that is about to occur.
Characteristics	receives control after the LOGOFF activity has been initiated (by the operator, as a result of a TIMEOUT, etc.), but prior to the logoff activity having been accomplished.
Input	**R1 + 0** address of the ANE **R1 + 4** fullword representing the reason for the LOGOFF (see SARFFLAG for values and meanings)
Output	**R15 = 0** continue with normal logoff processing **R15 > 0** bypass this logoff activity

EXT25 - Authentication

Purpose	provides a location for additional user verification (in addition to security package validation).
Characteristics	receives control from the LOGON processor as a function of the AUTHENTICATION keyword.
Input	**R1 + 0** address of the ANE
Output	**R15 = 0** network element has passed authentication **R15 > 0** network element has failed authentication
Comments	EXT25 is designed to operate with specific versions of The Network Director that are supporting the AUTHENTICATION keyword. If your version of The Network Director does not recognize the keyword, EXT25 will not be available. Contact North Ridge Software, Inc. for more details.

EXT26 - Initialization

Purpose	provides a location for system wide operations that should be accomplished prior to the normal dispatching cycle.
Characteristics	receives control from MAIN prior to the dispatcher (immediately prior to the issuance of message 199).
Input	**R12** address of the PDA **R13** address of a WAE
Output	None.
Comments	EXT26 receives control after all parsing of parameters and the control block structure is in place to complete initialization. The VTAM and VSAM ACBs are not open and the dispatcher (DISP) has not received control yet (TNDWAIT should not be used).

EXT27 - Termination

Purpose	provides a location for system wide operations that should be accomplished after termination has begun but prior to final cleanup.
Characteristics	receives control from MAIN after the dispatcher has terminated, but prior to the termination processor (DONE).
Input	**R12** address of the PDA **R13** address of a WAE
Output	None.
Comments	EXT27 receives control after a STOP command has been processed by the dispatcher. The VTAM and VSAM ACBs will still be OPEN (if they were open at the time of the STOP command) and all terminal sessions will still be in progress.

EXT28 - New Password

Purpose	provides a location for notification that a network terminal operator has set a new password via a security package.
Characteristics	receives control from ACF2 or RACF interface code **after** the security package has accepted the new password and **before** The Network Director has migrated the new password to the current password location.
Input	**R1 + 0** address of the ANE **R1 + 4** address of the new password
Output	**R15 = 0** continue (move the new password to the current password field) **R15 > 0** do not move the new password to the current password field, but do allow the logon to complete
Comments	The old password will reside in ANEPASS. Both the old and the new passwords will be in clear text form (they will be encrypted by The Network Director at exit from EXT28). EXT28 can be utilized to audit how network users are setting new passwords or other activities that may be required by your installation when a new password is set.

EXT29 - ACF2 Security System Call

Purpose	provides a location for review of each call made by The Network Director to ACF2.
Characteristics	receives control from The Network Director's ACF2 interface code immediately **prior** to giving control to ACF2.
Input	**R1 + 0** address of the ACF2 control block
Output	**R15 = 0** continue (the control block may have been modified, but cannot be moved). **R15 > 0** do not call ACF2, process the control block present as if it had been processed by ACF2
Comments	R0 and R1 are saved and restored by the EXT29 caller. These represent the ACF2 parameter list and function code being performed.

EXT30 - Password Check

Purpose	provides a location to preview the password that is about to checked.
Characteristics	receives control from LOGON during the logon validation process **prior** to comparing the password entered with the control block and/or the security system.
Input	**R1 + 0** address of the ANE (encrypted password stored at ANEPASS) **R1 + 4** address of message area (message to be returned to user)
Output	**R15 = 0** allow the logon and bypass normal Network Director checks **R15 = 4** reject the logon, message area may be filled in **R15 = 8** reject the logon and place the device on the Inactive List **R15 = 12** continue with normal Network Director password validation

EXT31 - Return from Application

Purpose	receives control when a network element has returned from an APPLICATION (right after message 457).
Characteristics	receives control from BUILD (for external applications) or from SEL (for internal applications) when the device is about to have an Application Selection Panel constructed for it.
Input	**R1 + 0** address of the ANE **R1 + 4** address of the ADB the user was connected to
Output	None.

EXT32 - Interval Operation Review

Purpose	receives control from control tasks within The Network Director to review the decision about operation of a specific *interval* operation or asynchronous activity within the network
Characteristics	receives control from NTWK or MNTR and can cause the associated system activity to continue or be deferred.
Input	**R1 + 0** activity being performed (0 = STATUS-INTERVAL) **R1 + 4** address of the ANE
Output	**R15 = 0** continue with the operation **R15 > 0** bypass the operation for the device

EXT33 - User DFB

Purpose	provides a manner in which an installation specific task can dispatch under its own unique DFB within The Network Director's dispatching environment.
Characteristics	receives control under a "USER" DFB, which must have been previously created via TNDALLOC
Input	R11 will address the USER DFB and R12 the PDA. If other arguments are desired, the code issuing TNDALLOC may use DFBPARM in the USER DFB to pass information.
Output	To be determined by the exit implementor.

EXT34 - Post Security System Call

Purpose	provides a manner in which an installation specific task can evaluate and inspect the results of a call to the security package.
Characteristics	receives control under a DFB, immediately after the ACF2, RACF, or TOPSECRET SVC has been invoked.
Input	**R1 + 0** ANE address **R1 + 4** ACEE or ACF2 control block address,
Output	**R15 = 0** Continue with logon **R15 = 4** Invoke the SVC again (presumably, the exit has taken an action and modified in some manner the control blocks associated with the call).

EXT36 - Post Logon Exit

Purpose	provides a point during logon processing where "logon complete" logic can be inserted.
Characteristics	receives control as a DFB task from TNDLOGON right after the logon action has completed successfully
Input	**R1 + 0** address of the ANE completing logon
Output	No processing options. Actions may be taken in ANE related control blocks and definitions.

EXT37 - Application Selection Panel Construction

Purpose	provides a location where the local installation can control which elements will be included on the Application Selection Panel
Characteristics	receives control as a DFB task from TNDBLD2 after all SSE screening has been achieved and prior to actual Application Selection Panel construction
Input	**R1 + 0** address of the ANE associated with the Application Selection Panel
Output	The exit can accomplish multiple actions. Most typically, EXT37 can scan the SSE chain (anchored in ANESSE) and determine which SSE (Screen Selection Elements) it will allow to be placed on the Application Selection Panel

Appendix A. Sample Exit LOG Listing

The following output from a Network Director execution displays an example of how various exits interact with the decisions that are made by The Network Director.

```
 22:10:03 TND0007C The Network Director (4.0.0) - Initialization is in progress
 22:10:03 TND0753G Operational in 24 bit addressing mode
 22:10:03 TND0204G -------------- Parameter Deck scan beginning --------------
 14:10:03 TND0008G GLOBALS APPLID=SEATTLE,WTO=10,SITE=SEATTLE,OPSYS=VM,
 14:10:03 TND0008G          SECURITY=VM,LOGON-MESSAGE=DETAIL,
 14:10:03 TND0008G          NAME='Sample Exit System'
 14:10:03 TND0008G *
 14:10:03 TND0008G APPLICATION ADMIN,TARGET=TNDADMIN,UPDATES=YES,
 14:10:04 TND0008G          TITLE='Network Administration'
 14:10:04 TND0008G APPLICATION MESSAGES,TARGET=TNDMSG,
 14:10:04 TND0008G          TITLE='Messages'
 14:10:04 TND0008G APPLICATION INFO,TARGET=TNDINFO,UPDATES=YES,
 14:10:04 TND0008G          TITLE='Network Information'
 14:10:04 TND0008G APPLICATION CMS,TARGET=VM,
 14:10:04 TND0008G          TITLE='CMS - Timesharing',INITIAL-DATA=(&NAME)
 14:10:04 TND0008G *
 14:10:04 TND0008G PROFILE SAMPLE,PRINTER=PRT01,PA3=DROP
 14:10:04 TND0008G *
 14:10:05 TND0008G DEFAULT PASSWORD=,SELECTIONS=*,DIM=1M,
 14:10:05 TND0008G          IDENTIFICATION=YES,
 14:10:05 TND0008G          PROFILE=(SAMPLE,CHANGE),TIMEOUT=15M,
 14:10:05 TND0008G          COMMANDS=YES,CUA=YES,
 14:10:05 TND0008G          LOGO=
 14:10:05 TND0008G          Director:  &VERS       SSCP:   &SSCP...
 14:10:05 TND0008G          Jobname:   &JOBNAME    Hostpu: &HOSTPU..
 14:10:05 TND0008G          Site:      &SITE...    CPU:    &CPUID
 14:10:05 TND0008G LOGO-END
 14:10:05 TND0008G GROUP SYSTEMS,APPLICATIONS=(INFO,MESSAGES,CMS,ADMIN),
 14:10:05 TND0008G          MAXIMUM=4
 14:10:05 TND0008G *
 14:10:05 TND0008G USERS ++++++++,GROUP=SYSTEMS,PASSWORD=
 14:10:05 TND0008G *
 14:10:05 TND0008G TERMINAL OPERATOR,APPLICATIONS=(ADMIN)
 14:10:05 TND0230G OPENing the VSAM ACB
 14:10:10 TND0201G -------------- Parameter Deck scan complete ---------------
 14:10:10 TND0060C Initialization is complete
 14:10:12 TND0675G Executing on a 4381, CPUID is 123456
 14:10:12 TND0676G Operating system release: 6.0.0618
 14:10:12 TND0677G The following PTFs have been applied:
 14:10:12 TND0647G This system has no zaps applied
 14:10:12  TND0663G  EXT01  assembled  at  11.37  on  08/27/93  for  4.0.1,  is  at
00024E98[1]
 14:10:12 TND0663G EXT02 assembled at 11.38 on 08/27/93 for 4.0.1, is at 00024D30
 14:10:12 TND0663G EXT03 assembled at 11.38 on 08/27/93 for 4.0.1, is at 00024BC8
```

---

[1] This is a list of exits that are active for this execution of The Network Director. It is printed in the LOG during initialization to provide a reminder of which exits are currently being utilized.

```
14:10:12 TND0663G EXT06 assembled at 11.39 on 08/27/93 for 4.0.1, is at 00024A60
14:10:12 TND0663G EXT08 assembled at 11.39 on 08/27/93 for 4.0.1, is at 000248F8
14:10:12 TND0663G EXT09 assembled at 11.40 on 08/27/93 for 4.0.1, is at 00024790
14:10:12 TND0663G EXT11 assembled at 11.41 on 08/27/93 for 4.0.1, is at 00024628
14:10:13 TND0663G EXT12 assembled at 11.41 on 08/27/93 for 4.0.1, is at 000244C0
14:10:13 TND0663G EXT13 assembled at 11.42 on 08/27/93 for 4.0.1, is at 00024358
14:10:14 TND0663G EXT15 assembled at 11.43 on 08/27/93 for 4.0.1, is at 000241F0
14:10:14 TND0663G EXT16 assembled at 11.36 on 08/27/93 for 4.0.1, is at 00024088
14:10:14 TND0663G EXT18 assembled at 11.34 on 08/27/93 for 4.0.1, is at 00025E98
14:10:14 TND0663G EXT19 assembled at 11.36 on 08/27/93 for 4.0.1, is at 00025D30
14:10:14 TND0663G EXT20 assembled at 11.35 on 08/27/93 for 4.0.1, is at 00025BC8
14:10:14 TND0663G EXT24 assembled at 11.35 on 08/27/93 for 4.0.1, is at 00025A60
14:10:14+TND0199C Control is being given to Sample Exit System (Version 4.0.0)
14:10:14 TND0209G OPENing the VTAM ACB
14:10:16 TND0554G VTAM Version 3.4.0 is in use[2]
14:10:16 TND0555G   Component id 5664-28001-340 is present
14:10:16 TND0556G   APPL: A01SEATTLE ACBNAME: SEATTLE NETID: NRS
14:10:16 TND0559G   SSCPNAME: SSCP1 HOSTPU: VTAM1
14:10:16 TND0557G   Subarea Address: 1 Element: 0 Maximum: 63
14:10:16+TND0623G Sample Exit System waiting for console input
14:10:17 TND0196C Application CMS is now active
14:10:17 TND0217G Establishing session for T01005 - Subarea: 1 Netid: NRS
14:10:18 TND0206G --- TNDEXT18 entered ----
14:10:18 TND0206G --- TNDEXT20 entered ----
14:10:30 TND0206G --- TNDEXT03 entered ----
14:10:30 TND0206G --- TNDEXT20 entered ----
14:10:30 TND0206G --- TNDEXT11 entered ----
14:10:30 TND0206G --- TNDEXT18 entered ----
14:10:30 TND0206G --- TNDEXT20 entered ----
14:10:35 TND0206G --- TNDEXT03 entered ----[3]
14:10:35 TND0206G --- TNDEXT20 entered ----
14:10:35 TND0206G --- TNDEXT19 entered ----[4]
14:10:35 TND0206G --- TNDEXT01 entered ----
14:10:35 TND0165S Id TEST1 - is now active at T01005
14:10:35 TND0206G --- TNDEXT20 entered ----
14:10:36 TND0206G --- TNDEXT11 entered ----
14:10:36 TND0206G --- TNDEXT15 entered ----
14:10:36 TND0206G --- TNDEXT15 entered ----[5]
14:10:36 TND0206G --- TNDEXT15 entered ----
14:10:36 TND0206G --- TNDEXT15 entered ----
14:10:36 TND0206G --- TNDEXT16 entered ----[6]
14:10:36 TND0206G --- TNDEXT16 entered ----
14:10:36 TND0206G --- TNDEXT16 entered ----
14:10:36 TND0206G --- TNDEXT16 entered ----
```

---

[2] This VTAM related information is extracted from the VTAM Vector List provided to The Network Director as a result of the OPEN ACB. It may not be present in your system (it's dependent upon operating system and VTAM release levels).

[3] This entry into EXT03 is where the terminal operator has entered his userid and password (requesting logon).

[4] EXT19 is entered at this point to set the GROUP for the user **attempting** to logon. The actual success or failure of the logon attempt will be established by EXT01 (the next exit).

[5] EXT15 is being entered repetitively here by The Network Director's selection screen creation process to establish the number of selections that will be present on the Application Selection Panel

[6] EXT16 is now being entered to review the individual formatted entry for the Application Selection Panel. Contrast EXT16 with when EXT15 receives control.

```
14:10:36 TND0206G --- TNDEXT18 entered ----[7]
14:10:36 TND0206G --- TNDEXT20 entered ----
14:10:39 TND0206G --- TNDEXT03 entered ----[8]
14:10:39 TND0206G --- TNDEXT20 entered ----
14:10:39 TND0206G --- TNDEXT11 entered ----
14:10:39 TND0206G --- TNDEXT12 entered ----
14:10:39 TND0227G Lu T01005 - TEST1 - has selected ADMIN
14:10:39 TND0206G --- TNDEXT20 entered ----
14:10:46 TND0206G --- TNDEXT03 entered ----
14:10:46 TND0206G --- TNDEXT20 entered ----
14:10:46 TND0206G --- TNDEXT08 entered ----[9]
14:10:46 TND0249R Input: d n=t01005
14:10:46 TND0206G --- TNDEXT08 entered ----
14:10:46 TND0254G Network Element T01005
14:10:46 TND0458G   is logged on as User TEST1
14:10:46 TND0310G   connected to ADMIN at 14:10:39 on 08/27/93
14:10:46 TND0632G   is a member of the GROUP SYSTEMS
14:10:46 TND0762G   Initial Logmode: NSX32702 Netid: NRS Subarea: 1
14:10:46 TND0456G   is active at offset 1D8A within REPT
14:10:46 TND0311G   has 24 logical lines of 80 characters each
14:10:46 TND0206G --- TNDEXT20 entered ----
14:10:49 TND0206G --- TNDEXT03 entered ----
14:10:49 TND0206G --- TNDEXT20 entered ----
14:10:49 TND0206G --- TNDEXT08 entered ----
14:10:49 TND0206G --- TNDEXT20 entered ----
14:10:53 TND0206G --- TNDEXT03 entered ----
14:10:53 TND0206G --- TNDEXT20 entered ----
14:10:53 TND0206G --- TNDEXT09 entered ----
14:10:53 TND0206G --- TNDEXT20 entered ----
14:10:56 TND0206G --- TNDEXT03 entered ----
14:10:56 TND0206G --- TNDEXT20 entered ----
14:10:56 TND0354G Profile TEST1 modified by User TEST1 at Terminal T01005
14:10:56 TND0206G --- TNDEXT20 entered ----
14:11:01 TND0206G --- TNDEXT03 entered ----
14:11:01 TND0206G --- TNDEXT20 entered ----
14:11:01 TND0206G --- TNDEXT08 entered ----
14:11:01 TND0457G Lu T01005 - TEST1 - has returned from ADMIN
14:11:01 TND0206G --- TNDEXT15 entered ----
14:11:01 TND0206G --- TNDEXT15 entered ----
14:11:01 TND0206G --- TNDEXT15 entered ----
14:11:01 TND0206G --- TNDEXT15 entered ----
14:11:01 TND0206G --- TNDEXT16 entered ----
14:11:01 TND0206G --- TNDEXT16 entered ----
14:11:01 TND0206G --- TNDEXT16 entered ----
14:11:01 TND0206G --- TNDEXT16 entered ----
14:11:01 TND0206G --- TNDEXT18 entered ----
14:11:01 TND0206G --- TNDEXT20 entered ----
14:11:09 TND0206G --- TNDEXT03 entered ----
14:11:09 TND0206G --- TNDEXT20 entered ----
14:11:09 TND0206G --- TNDEXT11 entered ----
14:11:09 TND0206G --- TNDEXT13 entered ----[10]
14:11:09 TND0184G KKKK is an unidentified command or application
14:11:09 TND0206G --- TNDEXT20 entered ----
14:11:16 TND0206G --- TNDEXT03 entered ----
```

---

[7] EXT18 is entered after the entire Application Selection Panel has been built, but prior to the output actually being sent.

[8] EXT03 has been entered as a result of the terminal operator having struck an aid key to make a request from the Application Selection Panel (looking at the time of day in the messages should help in interpreting when these items occur).

[9] EXT08 is entered to review the Network Administrator command.

[10] The terminal operator entered the literal string **KKKK**.

```
14:11:16 TND0206G --- TNDEXT20 entered ----
14:11:16 TND0206G --- TNDEXT11 entered ----
14:11:16 TND0206G --- TNDEXT13 entered ----
14:11:16 TND0184G KKKK is an unidentified command or application
14:11:16 TND0206G --- TNDEXT20 entered ----
14:11:20 TND0206G --- TNDEXT03 entered ----
14:11:20 TND0206G --- TNDEXT20 entered ----
14:11:20 TND0206G --- TNDEXT11 entered ----
14:11:20 TND0206G --- TNDEXT12 entered ----[11]
14:11:21 TND0227G Lu T01005 - TEST1 - has selected CMS
14:11:31 TND0457G Lu T01005 - TEST1 - has returned from CMS
14:11:32 TND0206G --- TNDEXT15 entered ----
14:11:32 TND0206G --- TNDEXT15 entered ----
14:11:32 TND0206G --- TNDEXT15 entered ----
14:11:32 TND0206G --- TNDEXT15 entered ----
14:11:32 TND0206G --- TNDEXT16 entered ----
14:11:32 TND0206G --- TNDEXT16 entered ----
14:11:32 TND0206G --- TNDEXT16 entered ----
14:11:32 TND0206G --- TNDEXT16 entered ----
14:11:32 TND0206G --- TNDEXT18 entered ----
14:11:32 TND0206G --- TNDEXT20 entered ----
14:11:37 TND0206G --- TNDEXT03 entered ----
14:11:37 TND0206G --- TNDEXT20 entered ----
14:11:37 TND0206G --- TNDEXT11 entered ----
14:11:37 TND0206G --- TNDEXT12 entered ----
14:11:37 TND0227G Lu T01005 - TEST1 - has selected ADMIN
14:11:37 TND0206G --- TNDEXT20 entered ----
14:11:44 TND0206G --- TNDEXT03 entered ----
14:11:44 TND0206G --- TNDEXT20 entered ----
14:11:44 TND0206G --- TNDEXT08 entered ----
14:11:44 TND0249R Input: st i
14:11:44 TND0206G --- TNDEXT08 entered ----
14:11:44 TND0206G --- TNDEXT20 entered ----
14:11:44 TND0054C Termination Stage Two beginning
14:11:50 TND0288G        1 terminals, 1 users active within Sample Exit System
14:11:50 TND0291G    1 users have selected ADMIN - Network Administration
14:11:50 TND0306G OPERATOR - Subarea: 0 Netid: NRS at Sample Exit System
14:11:50 TND0306G T01005 - TEST1 Subarea: 1 Netid: NRS at Network Administration
14:11:50 TND0289G        0 network elements are on the Inactive List
14:11:50 TND0318G CB pool is 88K with 69408 free, 0K in 31 bit storage
14:11:50 TND0318G MX pool is 72K with 43392 free, 0K in 31 bit storage
14:11:50 TND0319G   largest contiguous is 8064 bytes
14:11:50 TND0318G TP pool is 24K with 8704 free, 0K in 31 bit storage
14:11:50 TND0319G   largest contiguous is 4352 bytes
14:11:50 TND0320G Total pool allocations - 24K, free - 118K, 31 bit - 0K
14:11:50 TND0658G Maximum storage used: MX 72K, TP 48K
14:11:50 TND0383G The ADB chain has 4 elements with 0 free
14:11:50 TND0383G The ANE chain has 2 elements with 0 free
14:11:50 TND0383G The DFB chain has 10 elements with 1 free
14:11:50 TND0383G The DMT chain has 25 elements with 7 free
14:11:51 TND0383G The EFR chain has 2 elements with 1 free
14:11:51 TND0383G The GDB chain has 1 elements with 0 free
14:11:51 TND0383G The NIB chain has 3 elements with 0 free
14:11:51 TND0383G The PDE chain has 2 elements with 0 free
14:11:51 TND0383G The PPE chain has 2 elements with 0 free
14:11:51 TND0383G The RPL chain has 7 elements with 1 free
14:11:51 TND0383G The TDB chain has 1 elements with 0 free
14:11:51 TND0383G The UDB chain has 1 elements with 0 free
14:11:51 TND0383G The DIR chain has 1 elements with 0 free
14:11:51 TND0732G Logical File GETs: 7 PUTs: 1 ADDs: 0 ERASEs: 0
14:11:51 TND0053C Termination is complete
```

[11] EXT12 has the opportunity to modify the data being passed to the subsystem.

When developing installation exits, it is generally very useful to utilize the debugging facilities available from the operating system (TSO TEST or CP PER).[12] The Network Director displays the addresses of the exits during initialization to assist in this process. In addition, if you find it useful to create a listing similar to the one in this appendix (placing appropriate information in the LOG queue), simply code

```
TNDISSUE MESSAGE=206,PARM=(=CL72'--- TNDEXT?? ENTERED ---')
```

at the appropriate location in your exit.[13] This mechanism can also be utilized to display key control block fields, etc. or other messages and items that will assist in the proper operation of your exit.

---

[12] If you are using TSO TEST, The Network Director may not execute any privileged operations (SWAP=NO or calling RACF/TOP-SECRET), but most other functions should operate normally.

[13] TNDISSUE may be utilized only if you utilize TNDSTART to begin your CSECT and the proper WAE and DFB chaining has been set up.

Glossary

API: the Application Program Interface is the defined set of Assembler macros that provide a mechanism for programs to communicate with ACF/VTAM.

chain: identifies a collection of Network Director Control Blocks and their manner of connection.

CLSDST PASS: is the VTAM Macro and activity that allows The Network Director to transfer ownership of a network terminal to another VTAM subsystem.

control block: is a portion of storage that is utilized to describe a particular event or element within the logical network.

DFB task: identifies an activity that will compete for CPU resources within the rules associated with The Network Director's dispatching mechanism.

dispatching: is used to describe the logic flow followed by DISP when selecting the next piece of work (DFB) to accomplish.

DOS: IBM's Disk Operating System

External File: describes the disk based file used to store information that is to be preserved across Network Director restarts.

Installation Exit: A portion of code written by the computing facility to extend the function, features, and/or the security of The Network Director.

LOG queue: is the area within main storage that is committed to the wrap around message queue that is available for viewing via a Network Administrator's terminal.

logical network: is the combination of the Configuration Parameters, Network Administrator commands, and actual processing events that describe the network elements and how they are to be processed.

Macro: is the term used to describe prepared ASSEMBLER statements that can be used to shorten ASSEMBLER level source tasks.

Module: describes an executable CSECT within The Network Director.

MVS: IBM's Operating System (also described as OS).

network element: is equivalent to any item that is used to make up the logical network. It is usually described via a Network Director control block (xDB, ADB, ANE, etc).

network user: identifies a User or Terminal that is utilizing the facilities of the terminal network to accomplish his or her assigned tasks.

parameter statement: is used to describe a network element within the logical network.

segment: describes a portion of a Module. Each Module is divided into multiple segments, each of which has a unique function.

selection: used to describe the process a network user goes through to choose the next activity.

SMR: a System Measurement Record. This is the data record produced by The Network Director for accounting purposes. OS SMF or VSAM are the recording mediums available for the collections of SMRs.

source element: identifies a OS PDS Member or DOS Source Book. Each source element is analogous to a Network Director Module.

SSI: Single System Image is the term utilized within The Network Director to describe the facilities, exits, and concepts associated with eliminating the requirement for the terminal operator to have to continually logon to the various subsystems active within the system.

storage pools: are individual main storage areas that have differing allocation and manipulation techniques and are managed by STOR. CB, MX, and TP are the storage pool names.

vector: describes the general mechanism utilized to transfer between Modules via the PDA.

VTAM: the Virtual Telecommunications Access Method is the IBM provided software that implements SNA concepts within a diverse network of computers, terminals, and applications.

Index

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W