3 (2)

NO. 161

CYBER 180 SYSTEM DESIGN NOTE

Title: CYBER 180 Transaction Processing

Author: G. M. Herman

Date: 7/21/78

Status:
□ early warning
☒ preliminary
□ approved

Reviewed by: 7/26/78 AD&C

Approved by: ________________ NPP

This note will be superseded by: a subsequent update. This system design note replaces design note 150.

MEMO

CONTROL DATA CORPORATION

TO:	LOCATION:	FROM	EXT.	LOCATION:
Distribution		G. M. Herman	2707	ARH293

SUBJECT: CYBER 180 Transaction Processing

DATE: 7/21/78

We are preparing plans for CYBER 180 transaction processing at this time. Please review this second revision of CYBER 180 Transaction Processing and return any comments you may have by September 1. This document should be viewed as part of a long-range plan, and phasing will be required across several releases.

G. M. Herman

G. M. Herman
AD&C

paj

Attachment

Distribution:

C. T. Barr	MNA02A
R. J. Devers	HQS118
T. C. Boos	MNA02A
K. W. Brinkman	HQW09G
S. W. Fewer	ARH293
L. R. Gottschalk	ARH244
D. R. Haycraft	ARM08
D. B. Horning	SVL102
R. L. Howell	HQW05I
R. J. Johnson	ARH293
W. F. Keatts	ARH257
E. H. Michehl	ARH293
M. A. Morandi	ARH206
M. J. Mykkanen	HQW09G
D. P. Owen	SVL107
S. M. Radcliffe	SVL100
R. R. Ragan	SVL160
G. N. Robertson	SVL102
M. R. Swenson	HQS098
J. Thibodeau	ARH217
F. P. Vince	HQS100
J. L. Wardell	ARH244
J. E. Zelenka	HQW09G

TRANSACTION PROCESSING

CYBER 180

DCS ID ARH2497

Submitted: G.M. Herman

Approved: ____________________

____

____

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PURPOSE

This document describes transaction processing concepts for CYBER 180. These concepts will be reflected in the AO/R Revision D after review with development and marketing divisions.

The document is organized into four sections:

1) Introduction, references, and assumptions
2) Overview of CYBER 180 transaction processing concepts
3) Features required to support the major concepts
4) Summary of major requirements for each NOS/180 component

1.2 REFERENCES

CYBER 180 transaction requirements were influenced by the following source materials:

1) "Advanced 170 and CYBER 80 Forecast Assumptions, 1980-1990", F.P.Vince, 12 August 1977.

2) "Transaction Processing on the CYBER 18, A Competitive Analysis", R.Langley, M.Mykkanen, S.Pacholski, 30 June 1977.

3) "CYBER 170 Transaction Processing, A Competitive Analysis", M.J.Mykkanen, 15 March 1977.

4) "Transaction Strategy", F.P.Vince, 14 March 1977.

5) "Transaction Processing", R.A. Manning, 24 September 1976.

6) "Transaction Processing Requirements", D.E.Stahl, 4 March 1976.

7) "Final Report, Communications Team, Technical Migration Strategy Study Team", D.Lieberman, J.Lovgren, D.Owen, R.Zemlin, 27 January 1976.

8) "Transaction Processing Strategy", L.R.Gottschalk, 16 December 1974.

9) "Distributed Processing Technical Study", K.A.Lucke, 14 December 1977.

1.3 ASSUMPTIONS

CYBER 180 transaction requirements are based upon the following assumptions:

1) Control Data's ability to successfully market large general-purpose computer systems will be significantly enhanced in the years to come if it maintains a competitive transaction software system.

2) Transaction processing products should not be independent or self-contained, but should utilize standard system interfaces.

3) CYBER 180 must provide a transaction processing capability which addresses the needs of both systems and services: both overseas and stateside.

4) CYBER 180 should provide a generalized transaction processing capability using TAF/NOS as a design base.

5) It is not a requirement that CYBER 180 be compatible with TAF/NOS. Migration support will be limited to the preparation of difference documents.

6) CYBER 180 should offer in NOS/180 R2 a transaction processing system which is at least as capable as TAF/NOS.

7) The Message Control System (MCS) must be supported on CYBER 180.

8) CYBER 180 systems will frequently interface competitor systems within computer networks.

1.4 ENVIRONMENT

Transaction processing, sometimes referred to as teleprocessing, communications processing, or data base communications, is a mode of computer processing which enables multiple users at remote locations to simultaneously access a computer system by inputting structured sets of data. Usually, the user is not a trained computer professional (and does not want to become one), but a person performing some work function with the help of computers.

The data input is processed in a structured way by pre-defined program modules, and normally involves manipulation of one or more databases. Frequently, the completion of a terminal function (e.g., evaluating status of an inventoried part) requires several interactions with the computer system. The terminal function is called a "business transaction", and each interaction is called a "system transaction" or simply "transaction." The environment is time-sensitive, with specific constraints placed upon response times (usually in the range of 1 to 5 seconds).

Transaction volumes vary between sites, but rates in a range between 2 and 10 transaction per second are common. Transactions characteristics too vary, but generally each transaction (i.e., system transaction) requires execution of about 20,000 CYBER 180 instructions, for application, and 5 logical (12 physical) I/O operations, excluding data management logging. The logical processes of transactions are usually simple and repetitive, and the number of types of transactions is usually small (50-100) for a given application. Transactions are usually broken into several tasks which are shared among transaction types; each task dedicated to a simple specific function.

Preliminary response time objectives for NOS/180 on a target S2 configuration is 2 seconds average and 95% less than 5 seconds for loads of 10 transactions per second.

2.0 TRANSACTION PROCESSING OVERVIEW

The following describes some of CYBER 180's transaction processing concepts from the viewpoint of a user.

2.1 MARKETING OBJECTIVES

While recognizing the need for high-performance transaction processing, CYBER 180 will emphasize the low to mid peformance range in commercially-oriented applications. A basic transaction processing capability will be provided in NOS/180 R2, and a competitive transaction processing capability will be provided in NOS/180 R3. CYBER 180 will provide the tasking, the data management, and the communication facilities necessary to efficiently perform transaction processing while processing interactive and batch jobs. It will be possible to site tune system performance to benefit selected modes of processing, e.g., it will be possible to define scheduling priorities and aging rates for transaction tasks which are unique among those for all tasks. Transaction processes will use standard NOS/180 interfaces, and will access system resources in the same manner as do interactive and batch processes.

Transaction processing will be supported in multi-mainframe configurations, and in multiple hosts within a computer network. Load-leveling will be provided to permit efficient utilization of computer resources. This will not be dynamic load-leveling. All transactions from a given terminal are processed on a single mainframe. Terminal connection is made at LOGIN by NHP's Network Validation Facility (NVF). Several transaction applications may run concurrently. Each application will be isolated from another as one job is isolated from another, and may share resources as one job with another.

2.2 TASKING

2.2.1 APPLICATION CONTROL

A transaction application will be a set of tasks, procedures, files, and databases utilized for a common purpose or organization. Examples would be inventory control, state crime bureau, or TECHNOTEC applications. NOS/180 will ensure that transaction application names are unique.

An application will have a single owner. This owner will also own all resources of the application, and will be accountable for all resources consumed by the application. The owner may delegate privileges as he chooses (e.g., via access control entries) to others to administrate the application. Application Administrators will be able to develop, test, initialize, terminate, recover, and modify applications entirely from remote terminals. No host operator intervention will be required. It will be possible for the owner to delegate privileges so that those responsible for the application's daily operation (i.e., initialization, termination, and recovery) will not have application modification privileges. Programming errors in application tasks will not cause the application to abort.

All access to applications will be controlled from remote terminals by Applcation Administrators. Login of selected terminals may be implicit at application initialization. It will be possible to grant some terminals/users application access on request (e.g., TRANS,APPL), and to define other terminals as dedicated to an application. A front-desk terminal in a hotel lobby would be an example of a terminal which might be forced to a dedicated application.

All terminal LOGIN's must pass system validation. When access to an application is granted by the system, it will be possible for the application to further validate access if necessary. For instance, the application may make further inquiry of the terminal user, or periodically reissue a request for an application password.

2.2.2 TRANSACTION TASKS

Transactions will be processed by the execution of one or more independently compiled programs called tasks. Host language call interfaces will be provided for COBOL and FORTRAN to make it

possible to establish a new transaction (NEWTRN), to construct a new task chain (NEWCHN), to insert a list of tasks into the current task chain (ADDCHN), and to enqueue a transaction to a network queue (SEND). Transactions will not be necessarily processed on a first-in-first-out basis. It will be possible to process transactions on the basis of transaction priority, e.g., certain classes of transactions may be given priority over others in the same application.

Associated with each transaction active in an application will be a variable-length Communication Block containing a terminal identifier, a unique transaction identifier, control flags, date and time stamps, and an input message. See Figure 1. This Communication Block will be passed between tasks during execution of a transaction. Passing will be performed efficiently for performance reasons. Blocks will not be moved physically, but instead will be switched logically by the Transaction Executive as the "move" from task to task.

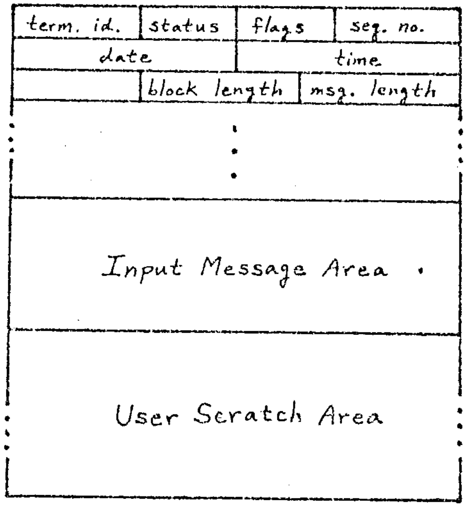


Figure 1

2.2.3 TRANSACTION I/O

Application tasks may receive two kinds of input: solicited and unsolicited. Transaction tasks will be able to perform file I/O using standard NOS/180 interfaces, including terminal I/O using the files INPUT and OUTPUT. A READ from the file INPUT would be an example of a solicited input. Solicited inputs are routed directly to the requesting task.

Examples of unsolicited input would be: a notification of a terminal LOGIN, a notification of terminal timeout, and a terminal input to cancel an active transaction. When an unsolicited input is received, a Communication Block will be prepared with the appropriate entries, and the Initial-Task (ITASK) will be initiated. Unique terminal identifiers will be assigned to NOS/180 and NAM inputs. If a fatal error occurs during execution of a transaction, the cause will be identified in the Communication Block, and the Error-Task (ETASK) will be initiated.

Figure 2 shows a transaction which generates a terminal response, and also creates a new transaction which it enqueues to a TAX application.

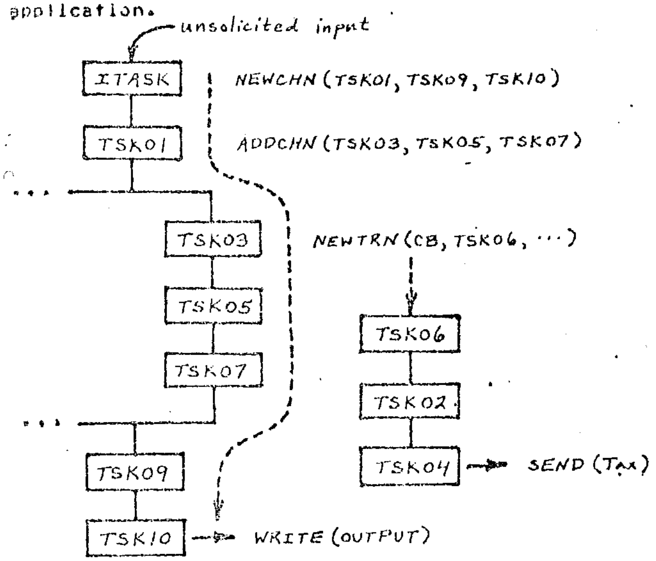


Figure 2

2.2.4 TRANSACTION MANAGEMENT

An optional Transaction Management Module will be provided which will, for a given application:

- verify that transaction types are valid for the terminals sending the transactions.
- verify that tasks called are valid for the requesting terminal.
- construct task chains needed to process transactions.
- enable and disable selected groups of terminals and transaction types.

Utilities will be provided which will enable Application Administrators to prepare and maintain the tables used by the Transaction Management Module, and thereby manage the application's functions.

2.3 DATA MANAGEMENT

2.3.1 DBMS INTERFACE

DMS-180 will provide data management services on CYBER 180. The system will not, however, prevent the use of foreign data management systems, e.g., TOTAL. Transaction applications will run as NOS/180 jobs, and will access DMS-180 using standard job and host language interfaces. Some features, e.g. recovery, will be available only through DMS-180 and selected AAM structures.

It will be possible to share databases between jobs of all modes: batch, interactive, and transaction. In multi-mainframe configurations, databases may be shared between jobs in all mainframes. Data security to the record and element levels will be provided by DMS-180. To control concurrent database access, DMS-180 will provide lock capabilities at both the record-type and record levels. Records and record-types which remain locked but not accessed for some installation-defined timeout period will be automatically unlocked. This timeout period may be based upon parameters other than time only: e.g., job-type, job-state, or number of records locked. Selected classes may be dually recorded. In the event an access to a dually recorded class fails, DMS-180 will automatically access the second copy and restore the failing copy. A class is an EDMS structure which defines attributes associated with elemental items or records.

At the completion of each request, DMS-180 will return a status to the caller. This status will indicate that the request was completed successfully, or indicate why the request failed. Periodically, jobs will be required to issue QUIET-POINT requests to DMS-180. These requests will cause DMS-180 to lock selected databases, copy these databases or recovery-needed data to back-up copies, and then unlock these databases. This will not effect the operation of other databases. Most database failures will be recovered by DMS-180 without user application intervention or knowledge. For instance, DMS-180 will be capable of regenerating databases using back-up copies and after-image records. In the unlikely event that this is not possible, DMS-180 will recover to the last back-up copy, and will notify jobs via returned status that recovery is necessary.

2.3.2 ALL OR NONE

Many transactions will perform more than one database update during their execution. For instance, one may update an employee's classification, establish a new salary rate, and write an access journal log entry. An important DMS-180 concept is that of "All or None". Either all updates will be performed correctly, or none will be performed. DMS-180, therefore, will provide the BEGIN_PARCEL, ROLLBACK, and END_PARCEL requests.

A DMS-180 Parcel is delimited by BEGIN_PARCEL and END_PARCEL requests. Database updates to locked records which occur within a Parcel will be temporary until an END_PARCEL request is processed. This request will cause DMS-180 to make permanent all updated records currently locked for a transaction, write after-image records necessary for recovery, and then release all locks. Temporary updates will be lost if an END_PARCEL request is not received before unlock occurs.

The ROLLBACK request will cause DMS-180 to undo to the last BEGIN_PARCEL all updates to records currently locked for a transaction, and then release all locks. This request may be issued, for instance, when a transaction is to be aborted. Updates occurring outside a Parcel or to records not locked will be immediate, and may not be cancelled by ROLLBACK requests.

2.3.3 TEST MODE

It will be possible for Application Administrators to test selected transactions in a "live" environment without endangering databases. DMS-180 requests will have a Test Mode parameter which may be set by transaction tasks. When this parameter is set, DMS-180 will direct the request to a test database for processing.

2.3.4 RECOVERY

System recovery, and the processing necessary during normal operations, to make recovery possible, may effect system performance. Successful recovery will require the coordination of DMS-180, AAM, and applications. Usually DMS-180 will be able to recover all permanent updates, and will require minimal operator intervention. Temporary updates will be lost during recovery.

In the event a database is not fully recoverable, DMS-180 will restore it to the last Quiet-Point and place the database in a Recovery State. This will not effect the operation of other databases. All DMS-180 requests will have Recovery Mode parameters which may be set by application tasks, indicating they are in Recovery Mode. Only requests with the Recovery Mode parameter set will be processed against databases in a Recovery State. This will prevent normal operations with the database until it is fully recovered.

All applications which used this database since the last QUIET-POINT must rerun their transactions in Recovery Mode and then notify the Database Administrator that recovery is complete. Any other jobs, e.g., batch jobs, must also be rerun if they modified the database since the last Quiet-Point. NCS/180 will provide facilities so applications can write message logs and rerun transactions and jobs. Careful coordination between applications and jobs sharing databases will be required to guarantee full recovery during rerun.

After the Database Administrator determines that all transactions and jobs necessary for recovery are complete, the database will be placed in Operational State. This will disable Recovery Mode processing on the database, and will begin normal database operations.

2.4 COMMUNICATIONS

2.4.1 MESSAGE ROUTING

A fundamental requirement of any transaction system is the ability to effectively communicate with terminals in a network. NOS/180 will satisfy this requirement with a basic communication facility utilizing standard interfaces. It will be possible for users to replace or augment this facility to meet individual needs.

Transactions will be able to send messages or transmit files to a single destination, or broadcast to a number of destinations. Each destination may be a device, a user, or a network queue; and may be referenced by logical name. This facility will be CDC's Message Control System (MCS) offering. When MCS development compromises are necessary, priority will be given to the needs of CDC's transaction processing capability. The physical destination, its location in the network, and its characteristics will be interpreted by NOS/180 in a manner which is transparent to application tasks. Senders may optionally await status indicating whether all data was properly routed. A network queue is identified by: a queue name, a job name, and a logical host identifier.

Selected terminals and users may be identified as capable of receiving unsolicited output, i.e., messages unexpected or not requested. When messages are sent unsolicited, they will be written to message files associated with the terminal or the user. If the terminal is active or the user is active on a terminal capable of receiving unsolicited output, an alert is sent to the terminal. The terminal operator may then request messages to be output to the terminal.

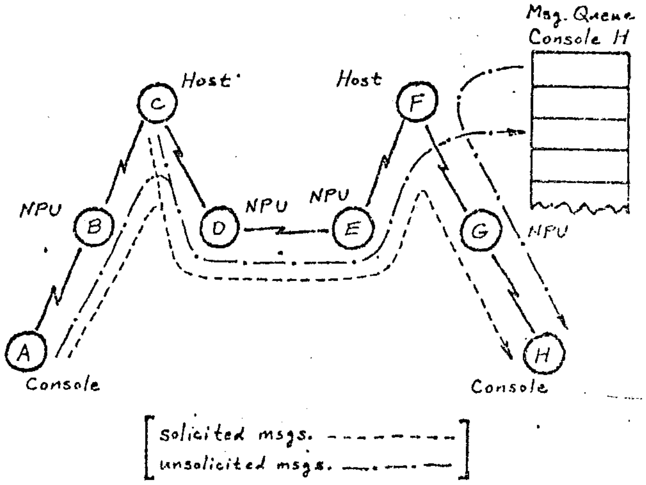


Figure 3

2.4.2 PAGE BROWSING

NCS/180 will support character, line, and block mode communication with display terminals. Outputs which exceed one page (screen) will be queued, and an alert will be given at the terminal indicating more pages are available. The operator may access these pages randomly or sequentially. Using proper application tasks, users will be able to chain inquiry outputs, i.e., save an output from one inquiry while making another.

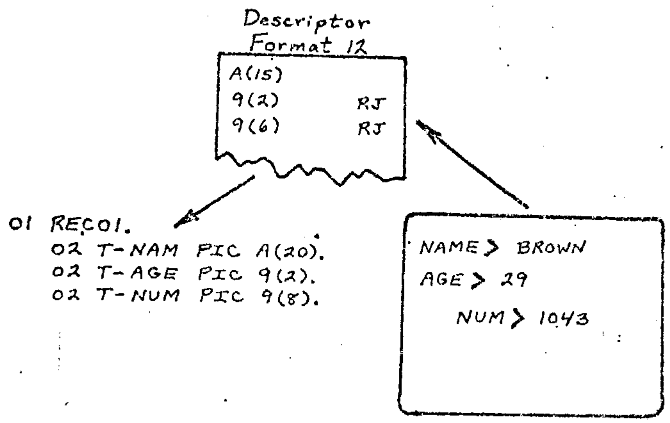
2.4.3 FORMATTED SCREENS

Transaction applications may optionally perform formatted-screen I/O to/from display terminals. Such I/O will:

- reduce the time to generate and modify display images
- isolate screen formats and application program formats
- eliminate terminal-type dependencies
- map terminal data to/from application program data
- perform input validation and justification

Using Format Services, an Application Administrator will be able to create at a remote console a "picture" of a screen image, indicating protected fields and variable formats. Format Services will transform this image into descriptors suitable for performing formatted-screen I/O. In this manner an Application Administrator will create new or modify existing screen images.

Host language interfaces to FORTRAN and COBOL will use the descriptors generated by Format Services when performing formatted-screen I/O. An error, e.g., an alphabetic character in a numeric field, during a formatted-screen READ or RECEIVE will be indicated on the user's terminal by a highlighted field and positioned cursor. An error during a formatted-screen WRITE or SEND will be indicated by returned status to the caller.



```
01 REC01.
   02 T-NAM PIC A(20).
   02 T-AGE PIC 9(2).
   02 T-NUM PIC 9(8).
```

ENTER FREAD USING FMT12

Figure 4

2.4.4 OFF-LINE SPOOLING

Terminal operators will be able to perform display-to-tape cassette operations in an off-line (local) mode, and later transmit the cassette messages to a host computer. Application tasks will be able to request tape cassette load and perform terminal tape cassette I/O using standard NCS/189 virtual terminal interfaces, e.g., READ and WRITE.

2.4.5 TERMINAL SUPPORT

Network products will support the following terminals and their functions:

- CDC 751 Terminal Subsystem
- CDC 752 Terminal Subsystem
- CDC 756 Display Terminal
- IBM 3270 Display Stations

The IBM 3270 protocol is a quasi-standard among many plug-compatible terminal vendors. Thus, NOS/180 will support these plug-compatible equipment. Asynchronous block mode, and synchronous mode SDLC, HDLC, and X.25 transmissions will be supported. Reference CYBER 180 AD/R Section 12.3.

The following terminal features will be supported:

- function keys
- tab control
- highlighted fields
- protected fields

NOS/180 network products will provide auto-baud and auto-terminal type detection, and will support auto-dial up of terminals. Idle terminals may be logged out after an installation-defined timeout period. This timeout period may be a function of terminal type, line type, and mode of operation.

2.4.6 DISTRIBUTED PROCESSING

Application Administrators will be able to submit jobs to initialize applications throughout a computer network, and to transmit object libraries between logical hosts. By routing messages to network queues using standard NOS/180 interfaces (MCS), transaction applications will be able to distribute their function and database throughout a computer network. This will be a Level 1 capability as defined in reference [9].

The actual distribution of processing will be controlled by the Network Administrator. Transaction applications will reference network queues using logical identifiers, and need not be concerned with the network configuration. The Network Administrator, using the Network Definition Language, will allocate logical hosts to physical hosts within the network, and thus distribute processing.

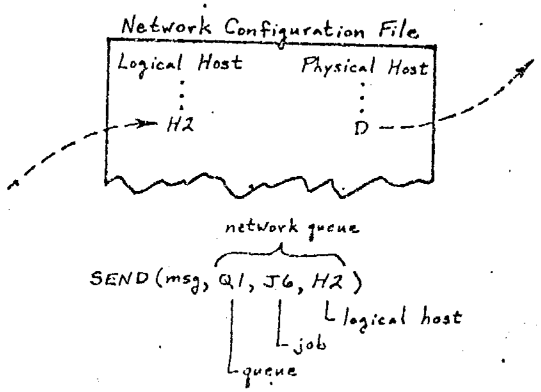


Figure 5

2.5 APPLICATION SUPPORT SOFTWARE

CDC will provide a number of software packages which will assist customers in a CYBER 180 transaction environment. These packages will be developed by implementing a transaction application for a CYBER 180 customer. This customer may be internal to CDC (e.g., COMSOURCE), or it may be external.

Software for the following functions will be provided. These are identified as a Special Programming (S) responsibility in Sections 3 and 4:

1) Initial Task (ITASK). An application task called by the Transaction Executive as the first user task executed for each transaction. Interprets unsolicited messages and routes to appropriate tasks for processing.

2) <u>Error Task (ETASK).</u> An application task called by the Transaction Executive when a transaction fatally aborts. Performs appropriate processing, e.g., ROLLBACK of updates and notifying the terminal user or Application Administrator.

3) <u>Administrator Interface.</u> Provides visibility and control of an application. An Application Administrator may, for instance, suspend selected transaction types, initiate Test Mode processing, or see the resources currently owned by an application.

4) <u>Transaction Rerun.</u> Resubmits transactions and batch jobs for rerun after database fallback to a Quiet-Point.

5) <u>Resource Reports.</u> Prepares reports showing the resources used by an application. Options permit reports by application, by transaction, by terminal, and by user.

6) <u>Cassette Submit.</u> Reads transaction messages from tape cassettes, and submits these as transactions for execution.

7) <u>Message Parsing.</u> Parses messages for use by application tasks. Edits free field terminal inputs into forms more suitable for task processing.

3.0 <u>TRANSACTION PROCESSING REQUIREMENTS</u>

The following describe 20 transaction processing concepts, and the NOS/180 features required to support each. Responsibilities are designated as follows:

NOS/180	(O)	=	CYBER 180 Operating System
DMS-180	(D)	=	Data Management
Net.Prod.	(N)	=	Network Products
Comp.Prod.	(C)	=	Compiler Products
Tran.Exec.	(E)	=	Transaction Executive
Spec.Prog.	(S)	=	Special Programming
User-Appl.	(U)	=	User Application

1) <u>Transaction Priorities.</u> It will be possible to process transactions on the basis of transaction priority within an application. It will be possible to alter a transaction's priority during its execution. This means:

NOS/180 -- must provide a program management command to change the priority of a task.

-- must enable a called task to inherit the priority of a caller task.

Comp.Prod. -- must provide a language call interface for COBOL and FORTRAN to change a task's priority.

2) <u>MMF Load-Leveling.</u> It will be possible to achieve load-leveling in a multi-mainframe configuration by sharing an application's transaction load between mainframes. This will <u>not</u> be dynamic load-leveling. All transactions from a given terminal are processed on a single mainframe. Terminal connection is made at LOGIN by NHP's Network Validation Facility (NVF). This means:

NOS/180 -- must permit a transaction application to execute concurrently in more than one mainframe in a multi-mainframe configuration.

-- must support concurrent access to shared files from all mainframes in a multi-mainframe configuration.

DMS-180 -- must support concurrent access and update of shared databases from all mainframes in a multi-mainframe configuration.

Net.Prod. -- must permit implicit login of selected terminals at application initialization.

-- must be able to select the mainframe for terminal connection at LOGIN on the basis of: NDL definition, hardware status, application status, or host operator direction.

-- must reroute terminal connections in the event of mainframe or link failure (need not be transparent to the terminal user).

3) Single Owner. Each transaction application will have a single owner. This owner will also own all resources of the application, and will be accountable for all resources consumed by the application. This means:

NOS/180 -- must run each application as a job.

-- must support multiple network terminals per job.

-- must permit jobs to perform job library list maintenance during on-line operation.

-- must permit users to request the resources consumed on a transaction basis.

-- must permit users to submit jobs to logical hosts in a computer network.

4) Task Chains. It will be possible for one task to initiate another task or task chain, with the option of continuing execution or awaiting completion of the called task or task chain. This means:

Tran.Exec. -- must maintain a task list for each active transaction.

-- must process NEWTRN, NEWCHN, ADDCHN, and ENDTSK requests.

-- must initiate the next task in the task list when the previous task completes (ENDTSK), or if none, terminate the transaction.

-- must release all locks at the termination of a transaction.

Comp.Prod. -- must provide COBOL and FORTRAN language call interfaces for NEWTRN, NEWCHN, ADDCHN, and ENDTSK requests.

5) Communication Block. A variable-length data block will be passed from one task to another during execution of a transaction. This block may be saved between transactions. This means:

Tran.Exec. -- must fetch and prepare a communication block for each unsolicited input.

-- must pass the communication block between transaction tasks or ENDTSK.

-- must maintain communication block storage space for each active terminal.

-- must optionally save the communication block at the completion of a transaction.

-- must save communication blocks when recovering.

Comp.Prod. -- must make communication blocks visible to COBOL and FORTRAN users.

6) Unsolicited Input. When an unsolicited input is received, a communication block will be prepared with the appropriate entries, and the task ITASK will be initiated. Applications will be capable of accepting unsolicited input while a transaction is in progress. This means:

NOS/180 -- must route solicited inputs to requesting tasks, and unsolicited inputs to the transaction executive.

Net.Prod. -- must permit input from terminals while transactions are active for the same terminals.

Tran.Exec. -- must fetch and prepare a communication block for each unsolicited input.

-- must establish an initial task list and then initiate ITASK.

-- must indicate cause and initiate the task ETASK when a fatal error occurs during execution of a transaction.

-- must be able to terminate all transactions for a given terminal (other than the caller), and then issue the necessary DMS-180 ROLLBACK requests.

Comp.Prod. -- must provide COBOL and FORTRAN language call interfaces for requests to terminate all transactions active for a given terminal.

7) Transaction Management. An optional Transaction Management Module and associated utilities will be provided which enable Application Administrators to manage an application's functions. This means:

Tran.Exec. -- must provide utilities to prepare and maintain module tables.

-- must verify, using module tables, that transaction types are valid for the terminals sending the messages.

-- must construct task chains needed to process transactions.

-- must process requests to enable and disable selected groups of terminals and transaction types.

8) Terminal Status. It will be possible for a terminal user to status the system at any time. A terminal user may receive the input and output messages associated with the last successfully completed transaction for the terminal. This means:

NOS/180 -- must route unsolicited inputs to the transaction executive.

Net.Prod. -- must permit input from terminals while transactions are active for the same terminals.

Tran.Exec. -- must fetch and prepare a communication block for each unsolicited input.

-- must establish an initial task list and then initiate ITASK.

User-Appl. -- must maintain terminal activity records.

-- must prepare and send status response to the terminal: last messages are available from the communication block.

9) NAM Messages. It will be possible to initiate execution of a task as a result of a terminal being newly connected, reconnected during recovery, disconnected, or logged out from an application. This means:

NOS/180 -- must route unsolicited inputs to the transaction executive.

Net.Prod. -- must prepare and send an unsolicited message to an application on terminal connect, reconnect during recovery, disconnect, or logout.

Tran.Exec. -- must fetch and prepare a communication block for each unsolicited input.

-- must establish an initial task list and then initiate ITASK.

10) Standard Interfaces. Transaction applications will use standard NOS/180 interfaces, and will have the same access to system resources (e.g., tapes, files, databases, and network products) as do other applications. This means:

NOS/180 -- must process transaction requests using the same interfaces as requests from other modes.

DMS-180 -- must process transaction requests using the same interfaces as requests from other modes.

Net.Prod. -- must process transaction requests using the same interfaces as requests from other modes.

Comp.Prod. -- must provide common interfaces to all modes of processing.

Tran.Exec. -- must require no special transaction-only services from NOS/180 or its product set.

11) Lock Control. DMS-180 will provide lock capabilities at both the record-type and record levels. Record-types and records which remain locked but not accessed for some installation-defined timeout period will be unlocked. This means:

NOS/180 -- must provide a clock or interval timer for DMS-180 use.

DMS-180 -- must process requests to lock and unlock record-types and records.

-- must enable transactions to be the "owner" of locks.

-- must monitor inactive locked areas, and unlock these after a timeout period.

Comp.Prod. -- must provide COBOL and FORTRAN language interfaces to lock and unlock record-types and records of DMS-180 databases.

Tran.Exec. -- must release all locks at the termination of a transaction.

12) Quiet-Point. DMS-180 will process QUIET_POINT requests. Most database failures will be recovered by DMS-180 without user application intervention or knowledge. This means:

DMS-180 -- must copy selected databases or recovery-needed data to back-up copies during Quiet-Point processing.

-- must lock databases during Quiet-Point operations.

-- must be able to recover databases, or portions of databases, concurrently with other system operations.

-- must provide the necessary Database Administrator interface necessary to manage database back-up and recovery.

-- must provide COBOL and FORTRAN language call interfaces for QUIET_POINT requests.

Tran.Exec. -- must idle and restart its application when requested by a task (no-op if already idled or restarted).

User-Appl. -- must periodically request DMS-180 to Quiet-Point.

-- must request the application to idle before issuing QUIET_POINT requests.

-- must periodically request Quiet-Point status from DMS-180.

-- must request the application to restart after completion of Quiet-Point processing.

13) ROLLBACK/END PARCEL. DMS-180 will process BEGIN_PARCEL, ROLLBACK, and END_PARCEL requests, and will ensure that "all or none" of each parcel of updates are performed. This means:

DMS-180 -- must treat all updates to locked record-types and records which occur within Parcels as temporary until an END_PARCEL request is received.

-- must make permanent "all or none" of the temporary updates for a transaction and release locks when an END_PARCEL request is received.

-- must write after-image records for the purposes of recovering from the last Quiet-Point.

-- must undo all temporary updates for a transaction and release locks when a ROLLBACK request is received.

-- must undo all temporary updates to record-types and records when unlocked, with an unlock option to make updates permanent.

-- must provide COBOL and FORTRAN language call interfaces to DMS-180 BEGIN_PARCEL, ROLLBACK, and END_PARCEL requests.

User-Appl. -- must issue DMS-180 BEGIN_PARCEL, ROLLBACK, and END_PARCEL requests.

14) Test Mode. It will be possible for Application Administrators to test selected transactions in a "live" environment without endangering databases. This means:

DMS-180 -- must evaluate the Test Mode parameter on DMS-180 requests.

-- must support test databases.

-- must provide COBOL and FORTRAN language call interfaces for DMS-180 requests which include Test Mode parameters.

User-Appl. -- must provide any logic necessary to select which transactions are to run in Test Mode.

-- must set the Test Mode parameter on DMS-180 requests.

-- must reroute I/O on the basis of Test Mode, if desired (network products will not evaluate Test Mode parameters).

15) Database Recovery. In the event a database is not fully recoverable, it will be possible to rerun applications and jobs concurrently with other system operations. This means:

DMS-180 -- must always be able to restore a database to the last Quiet-Point.

-- must process requests to restore a database to the last Quiet-Point.

-- must process for databases in Recovery State only requests with the Recovery Mode parameter set.

-- must return databases to Operational State when directed by the Database Administrator.

-- must process for databases in Operational State no requests with the Recovery Mode

parameter set.

-- must provide the Database Administrator interface necessary to control the recovery process.

-- must provide COBOL and FORTRAN language call interfaces for DMS-180 requests which include Recovery Mode parameters.

-- must provide COBOL and FORTRAN language call interfaces for requests to restore databases to the last Quiet-Point.

-- must provide recovery of AAM files to the last Quiet-Point.

Spec.Prog. -- must resubmit transactions (via NEWTRN or SEND requests) with Recovery Mode parameters set on DMS-180 requests.

-- must signal the Database Administrator when recovery is complete for a job or application.

-- must periodically request the recovery status from DMS-180.

-- must request the application to restart after operational status is returned by DMS-180.

User-Appl. -- must log transactions and jobs so they may be resubmitted for recovery.

-- must request the application to idle when recovery status is returned from DMS-180.

-- must initiate application and job rerun procedures.

-- must reroute I/O on the basis of Recovery mode, if desired (network products will not evaluate Recovery Mode parameters).

16) Message Routing. Tasks will be able to send messages and transmit files to a single destination, or broadcast to a number of destinations. Each destination may be a device, a user, or a network queue; and may be referenced by logical name. This facility will be CDC's Message Control

System (MCS) offering. This means:

NOS/180 -- must provide message queues which may be associated with selected tasks.

-- must provide an SCL statement to send messages or transmit files to single devices or users.

Net.Prod. -- must transmit messages and files to network queues, users, and devices located throughout a computer network.

-- must transform logical device and queue identifiers to physical devices and queues.

-- must accept both single destination and broadcast requests.

-- must identify those users and devices capable of receiving unsolicited output.

-- must maintain an unsolicited message queue for each user and device capable of receiving unsolicited output.

-- must enqueue unsolicited messages for users and devices, and send alerts if currently active.

-- must send an alert at terminal login time if unsolicited messages await the user or device.

-- must transmit unsolicited message queues to terminals on request.

-- must discard unsolicited messages after an installation-defined timeout period.

Comp.Prod. -- must provide COBOL and FORTRAN language interfaces to send and broadcast messages and files.

User-Appl. -- must send broadcast messages, and all messages to network queues.

17) Page Browsing. Display terminal outputs which exceed one ! page (screen) will be queued, and an alert will be given at the terminal indicating more pages are available. The

operator may access these pages randomly or sequentially. This means:

Net.Prod. -- must be able to enqueue pages for display terminals.

-- must alert terminals when multiple page (screen) outputs are sent.

-- must permit users to select pages for display: next, back, first, last, forward 3, back 6, etc..

-- must optionally display the current page number with each page.

-- must identify page-end by either line count or delimiter.

18) Formatted I/O. Application Administrators will be able to ! create new or modify existing screen image definitions from remote consoles using Format Services. These image definitions will be used during formatted-screen I/O. This means:

Net.Prod. -- must support highlighted fields and cursor control on display terminals.

Comp.Prod. -- must provide COBOL and FORTRAN language call ! interfaces necessary to perform ! formatted-screen I/O.

-- must map terminal data to/from application programs.

-- must provide a Format Services utility which will transform remote console "pictures" into image definitions.

-- must maintain image definitions for use during formatted-screen I/O.

-- must type validate and right/left justify I/O data.

-- must send to the console indications of input errors.

19) Off-Line Spooling. Terminal operators will be able to ! perform display-to-tape cassette operations in an off-line

(local) mode, and later transmit the cassette messages to a host computer. This means:

NOS/180 -- must support tape cassette I/O using standard I/O interfaces.

Net.Prod. -- must support terminals with off-line (local) mode of operation.

-- must identify tape cassettes as network devices.

-- must support tape cassette I/O using standard network I/O procedures.

Comp.Prod. -- must provide COBOL and FORTRAN language interfaces to perform I/O, e.g., READ and RECEIVE.

Spec.Prog. -- must perform tape cassette I/O using standard NOS/180 interfaces.

-- must submit transactions for execution.

User-Appl. -- must initiate the Cassatte Submit utility.

20) <u>Terminal Support.</u> NOS/180 network products will support the following terminals and their functions:

- CDC 751 Terminal Subsystem
- CDC 752 Terminal Subsystem
- CDC 756 Display Terminal
- IBM 3270 Display Stations

This means:

Net.Prod. -- must provide auto-baud detection and auto-terminal type detection.

-- must support auto-dial up of terminals.

-- must permit logout of terminals idle for an installation-defined timeout period.

-- must support BSC and SDLC communication protocols.

21) <u>Distributed Processing.</u> Transaction applications will be able to distribute their function and database throughout a computer network by routing messages to network queues.

Message routing will be performed using standard NOS/180 interfaces (MCS). This means:

NOS/180 -- must provide message queues which may be associated with selected tasks.

-- must permit jobs to perform job library list maintenance during on-line operation.

-- must permit users to submit jobs to logical hosts in a computer network.

Net.Prod. -- must be able to transmit files as transparent text between hosts in a computer network.

-- must transmit messages and files to network queues located throughout a computer network.

-- must transform logical queue identifiers to physical queues.

Comp.Prod. -- must provide MCS and the associated interfaces to COBOL and FORTRAN.

Tran.Exec. -- must initialize application files and queues, and then schedule ITASK.

User-Appl. -- must provide the jobs which initialize an application, its object libraries and files, in each logical host in a computer network.

22) <u>Application Support Software.</u> CDC will provide a number of software packages which will assist customers in a CYBER 180 transaction processing environment. This means:

NOS/180 -- must permit t asks to request resources consumed on a transaction basis.

Spec.Prog. -- must provide a basic initial task (ITASK).

-- must provide a basic error processing task (ETASK).

-- must provide a transaction application interface to the Application Administrator.

-- must provide a utility which will resubmit transactions for rerun after fallback to a

Quiet-Point.

- -- must provide a utility which reports the resources used; by application, transaction, terminal, and user.
- -- must provide a utility which submits transactions stored on tape cassettes.
- -- must provide a utility which parses messages for use by application tasks.

Summary

No.	Concept	Requirement	Resp.
1	Transaction Priorities	Tasks inherit caller's priority	O
		Request to change task's priority	O-C
2	MMF Load-Leveling	Same application in multi-mainframes	O
		Concurrent access in MMF	O-D
		Implicit terminal login	N
		Selectable mainframe connect in MMF	N
		Reroute terminal connections	N
3	Single Owner	Applications run as jobs	O
		Multiple terminals per job	O
		On-line library maintenance	O
		Request transaction resources	O
		Submit to logical hosts	O
4	Task Chains	Task list per transaction	E
		Process task list requests	E-C
		Initiate next task	E
		Release locks at termination of transaction	E
5	Communication Block	Prepare communication blocks	E
		Pass communication blocks	E
		Maintain CB storage areas	E
		Save CB at transaction end	E
		Save CB's when recovering	E
		Commun. block visibility	C
6	Unsolicited Input	Route input messages	O
		Permit busy terminal input	N
		Prepare communication blocks	E
		Establish initial task list	E
		Initiate ETASK on errors	E
		Terminate active transactions	E-C
7	Transaction Management	Module tables utilities	E
		Validate transaction types	E

```
!    !                            !Construct task chains    !E      !   !
!    !                            !Enable and disable groups!E      !   !
!    !                            !                         !       !
! 8  !Terminal Status             !Route input messages     !O      !   !
!    !                            !Permit busy terminal input!N     !
!    !                            !Prepare communication blocks!E   !
!    !                            !Establish initial task list!E    !
!    !                            !Maintain terminal busy   !U      !   !
!    !                            !  status                 !       !   !
!    !                            !Prepare and send status  !U      !
!    !                            !  resoponse              !       !
!    !                            !                         !       !
! 9  !NAM Messages                !Route input messages     !O      !   !
!    !                            !Send messages re terminal!N      !
!    !                            !  connection             !       !
!    !                            !Prepare communication blocks!E   !
!    !                            !Establish initial task list!E    !
!    !                            !                         !       !
!10  !Standard Interfaces         !Same processing for      !O-O-N! !
!    !                            !  all modes              !       !
!    !                            !Common interfaces for all!O-C    !   !
!    !                            !  modes                  !       !
!    !                            !No transaction-only      !E      !
!    !                            !  requirements           !       !
!    !                            !Record level security    !O      !   !
!    !                            !                         !       !
!11  !Lock Control                !Provide clock or interval!O      !   !
!    !                            !  timer                  !       !
!    !                            !Lock/Unlock record-types !O      !   !
!    !                            !  and records            !       !   !
!    !                            !Transaction locks ownership!O    !   !
!    !                            !Unlock after timeout period!O    !   !
!    !                            !Release locks at termination!E   !   !
!    !                            !  of transaction         !       !   !
!    !                            !                         !       !
!12  !Quiet-Point                 !Copy to back-up copies   !O      !   !
!    !                            !Lock databases during    !O      !
!    !                            !  Quiet-Point            !       !
!    !                            !Concurrent database recovery!O   !   !
!    !                            !Database Administrator   !O      !
!    !                            !  Interface              !       !
!    !                            !Periodically request Quiet-!O-U  !   !
!    !                            !  Point                  !       !   !
!    !                            !Idle application before  !O-E-U! !
!    !                            !  Quiet-Point            !       !   !
!    !                            !Periodically status      !O-U    !   !
!    !                            !  Quiet-Point            !       !   !
!    !                            !Restart application after!O-E-U! !
!    !                            !  Quiet-Point            !       !   !
!    !                            !                         !       !
```

```
!13  !ROLLBACK/END_PARCEL         !Treat locked updates as  !O      !   !
!    !                            !  temporary              !       !
!    !                            !Update "all or none" on  !O      !
!    !                            !  END_PARCEL             !       !   !
!    !                            !Write after-images for   !O      !
!    !                            !  recovery               !       !
!    !                            !Undo temporary updates on!O      !
!    !                            !  ROLLBACK               !       !   !
!    !                            !Undo temporary updates on!O      !
!    !                            !  Unlock                 !       !
!    !                            !Request BEGIN_PARCEL,    !O-U    !   !
!    !                            !  ROLLBACK, and END_PARCEL!      !   !
!    !                            !                         !       !
!14  !Test Mode                   !Evaluate Test Mode       !O      !   !
!    !                            !  parameters             !       !   !
!    !                            !Support test databases   !O      !   !
!    !                            !Transaction selection logic!U    !
!    !                            !Set Test Mode parameters !U      !
!    !                            !Reroute Test Mode I/O    !U      !
!    !                            !                         !       !
!15  !Database Recovery           !Last Quiet-Point restorable!O    !   !
!    !                            !Requests to restore databasesO   !   !
!    !                            !Recovery State/Mode requests!O   !   !
!    !                            !Operational State after  !O      !
!    !                            !  recovery               !       !
!    !                            !Operational State/Mode   !O      !   !
!    !                            !  requests               !       !
!    !                            !Database Administrator   !O      !
!    !                            !  interface              !       !
!    !                            !Log transactions and jobs!U      !   !
!    !                            !  for recovery           !       !
!    !                            !Idle application for recoveryC-E-U! !
!    !                            !Initiate rerun process   !U      !   !
!    !                            !Resubmit transactions in !S      !   !
!    !                            !  Recovery Mode          !       !   !
!    !                            !Reroute Recovery Mode I/O!U      !   !
!    !                            !Signal DB Admin.  after  !S      !   !
!    !                            !  recovery               !       !   !
!    !                            !Periodically status      !O-S    !   !
!    !                            !  recovery               !       !   !
!    !                            !Restart application      !C-E-S! !
!    !                            !  after recovery         !       !   !
!    !                            !AAM files recovery       !O      !   !
!    !                            !                         !       !
!16  !Message Routing             !Associate queues with tasks!O    !   !
!    !                            !SCL statement to send    !O      !
!    !                            !Transmit messages and files!N    !
!    !                            !Transform logical to     !N      !
!    !                            !  physical devices       !       !
!    !                            !Accept single and        !N-C    !
```

		broadcast requests			
		Identify unsolicited destinations	N		
		Maintain unsolicited message queues	N		
		Alert on unsolicited output	N		
		Alert at terminal login	N		
		Send unsolicited messages to output	N		
		Discard messages after timeout	N		
		Send broadcast and queue messages	C-U		!
17	Page Browsing	Enqueue multiple pages output	N		!
		Alert when multiple pages	N		
		Process page selection requests	N		
		Display page number optionally	N		
		Line count & page delimiter	N		
18	Formatted I/O	Highlighted fields and cursor control	N		!
		Interfaces for formatted-screen I/O	C		
		Map formatted-screen data	C		
		Provide Format Services utility	C		
		Maintain image definitions	C		
		Validate and justify formatted data	C		
		Send error indications to consoles	C		
19	Cff-Line Spooling	Initiate utility	U		!
		Cassette I/O via standard interfaces	O-C-S		!
		Submit transactions	S		!
		Local-mode terminal support	N		
		Tape cassette identification	N		
		Cassette I/O via standard protocols	N		
20	Terminal Support	Auto-baud and auto-type	N		!
		Auto-dial up support	N		
		Idle terminal logout	N		
		BSC and SDLC protocols	N		

21	Distributed Processing	Associate queues with tasks	O		!
		On-line library maintenance	O		
		Submit jobs to logical hosts	O		
		Route files between logical hosts	O		
		Transmit files between hosts	N		
		Transmit messages & files	N		
		Transform logical to physical devices	N		
		MCS and interfaces	C		!
		Initialize application files	E		
		Provide jobs to initiate applications	U		
22	Application Support	Basic Initial-Task (ITASK)	S		!
		Basic Error-Task (ETASK)	S		!
		Administrator interface	S		!
		Transactions rerun utility	S		!
		Resource report generator	O-S		!
		Cassette submit utility	S		!
		Message parsing utility	S		!

4.0 NOS/180 COMPONENT REQUIREMENT

Summarized below are the requirements necessary to support transaction processing. The YES/NO codes indicate whether requirements are defined in current CYBER 180 design or objective documents.

Resp.	No.	Requirement	Defined Capab.
NOS/180	1	Tasks inherit caller's priority	Yes
	2	Request to change task's priority	Yes
	3	Same application in multi-mainframes	Yes
	4	Concurrent file access in MMF	Yes
	5	Applications run as jobs	Yes
	6	Multiple terminals per job	No
	7	On-line library maintenance	No
	8	Route input messages	No
	9	Same processing for all modes	Yes
	10	Provide clock or interval timer	Yes
	11	Associate queues with tasks	No
	12	SCL statement to send messages	No
	13	Perform I/O via standard interfaces	Yes
	14	Request resources on transaction basis	No
	15	Submit jobs to logical hosts	Yes
	15	Route files between logical hosts	Yes
DMS-180	1	Concurrent database access in MMF	No
	2	Same processing for all modes	Yes
	3	Lock/Unlock record-types and records	No
	4	Transaction ownership of locks	No
	5	Unlock after timeout period	No
	6	Copy data to back-up copies	No
	7	Lock databases during Quiet-Point	No
	8	Concurrent database recovery	Yes
	9	Database Administrator interface	No
	10	Treat locked updates as temporary	Yes
	11	Update "all or none" on END_PARCEL	No
	12	Write after-images for recovery	Yes
	13	Undo temporary updates on ROLLBACK	Yes
	14	Undo temporary updates on Unlock	No
	15	Evaluate Test Mode parameters	No

	16	Support test databases	No
	17	Last Quiet-Point restorable	Yes
	18	Recovery State/Mode requests	No
	19	Operational State after recovery	No
	20	Operational State/Mode requests	No
	21	Record level security	No
	22	Interfaces for the following requests:	
		- Lock/Unlock record-types & records	No
		- QUIET_POINT	No
		- BEGIN_PARCEL	No
		- ROLLBACK	No
		- END_PARCEL	No
		- Restore database	No
	23	AAM files recovery	No
Nat.Prod.	1	Selectable mainframe connect in MMF	No
	2	Permit busy terminal input	Yes
	3	Send messages re terminal connection	Yes
	4	Same processing for all modes	Yes
	5	Transmit messages and files	Yes
	6	Transform logical to physical devices	Yes
	7	Accept single and broadcast request	Yes
	8	Identify unsolicited destinations	No
	9	Maintain unsolicited message queues	No
	10	Alert on unsolicited output	No
	11	Alert at terminal login	No
	12	Send unsolicited message to output	No
	13	Discard messages after timeout	No
	14	Enqueue multiple pages output	Yes
	15	Alert when multiple pages	Yes
	16	Process page selection requests	No
	17	Display page number optionally	No
	18	Line count and page delimiter	Yes
	19	Highlighted fields & cursor control	No
	20	Local-mode terminal support	Yes
	21	Tape cassette identification	No
	22	Cassette I/O via standard protocols	No
	23	Auto-baud and auto-type	No
	24	Auto-dial up support	Yes
	25	Idle terminal logout	Yes
	26	BSC and SDLC protocols	No
	27	Implicit terminal login	No
	28	Reroute terminal connections	No
Comp.Prod.	1	Common interfaces for all modes	Yes
	2	Interfaces the following requests:	
		- Change task priority	No
		- New chain (NEWCHN)	No
		- Add chain (ADDCHN)	No
		- New transaction (NEWTRN)	No

		- End task (ENDTSK)	No
		- Terminate active transactions	No
		- Idle/Restart application	No
		- Formatted-screen I/O	No
		- Send messages and files	No
		- File I/O	Yes
		- Record I/O	Yes
	3	Provide Format Services utility	No
	4	Map formatted-screen data	No
	5	Maintain image definition	No
	6	Validate & justify formatted data	No
	7	Send error indications to consoles	No
	8	Communication block visibility	No
	9	Provide Message Control System (MCS)	Yes
Tran.Exec.	1	No transaction-only requirements	Yes
	2	Task list per transaction	No
	3	Process task list requests	No
	4	Initiate next task	No
	5	Prepare communication blocks	No
	6	Pass communication blocks	No
	7	Maintain CB storage areas	No
	8	Save CB at transaction end and recovery	No
	9	Establish initial task list	No
	10	Initiate ETASK on fatal errors	No
	11	Terminate active transactions	No
	12	Idle and restart applications	No
	13	Initialize application files	No
	14	Release locks at transaction termination	No
	15	Module tables utilities	No
	16	Validate transaction types	No
	17	Construct task chains	No
	18	Enable and disable selected groups	No
Spec.Prog.	1	Resubmit transactions in Recovery Mode	No
	2	Signal DB Admin. after recovery	No
	3	Periodically status recovery	No
	4	Restart application after recovery	No
	5	Issue cassette I/O requests	No
	6	Submit cassette transactions	No
	7	Provide basic Initial-Task (ITASK)	No
	8	Provide basic Error-Task (ETASK)	No
	9	Provide application admin. interface	No
	10	Provide transactions rerun utility	No
	11	Provide resource report generator	No
	12	Provide cassette submit utility	No
	13	Provide message parsing utility	No
User-Appl.	1	Maintain terminal busy status	No
	2	Prepare and send status response	No

	3	Periodically request Quiet-Point	No
	4	Idle application before Quiet-Point	No
	5	Periodically status Quiet-Point	No
	6	Restart application after Quiet-Point	No
	7	Process DMS-180 messages	No
	8	Issue DMS-180 recovery requests	No
	9	Provide Test Mode selection logic	No
	10	Set Test Mode parameters	No
	11	Reroute Test Mode I/O	No
	12	Log transactions for recovery	No
	13	Idle application for recovery	No
	14	Initiate rerun process	No
	15	Reroute Recovery Mode I/O	No
	16	Initiate Cassette Submit utility	No
	17	Send broadcast and queue messages	Yes
	18	Provide jobs to initiate applications	Yes