

WRDC-TR-90-8007
Volume V
Part 11

AD-A252 452



INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 11 - Neutral Data Manipulation Language (NDML) Precompiler
Parse Application Procedure Division Product Specification

M. Apicella, J. Slaton, B. Levi

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

September 1990



Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

92-14315



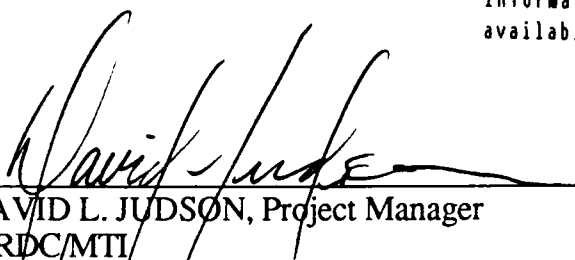
MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

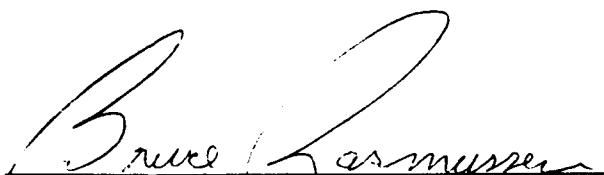
This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations


DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:


BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) PS 620341211			5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR- 90-8007 Vol. V, Part 11	
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services		6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI	
6c. ADDRESS (City, State, and ZIP Code) 2970 Presidential Drive Fairborn, OH 45324-6209			7b. ADDRESS (City, State, and ZIP Code) WPAFB, OH 45433-6533	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF		8b. OFFICE SYMBOL (if applicable) WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464	
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, Ohio 45433-6533			10. SOURCE OF FUNDING NOS.	
11. TITLE (Include Security Classification) See block 19			PROGRAM ELEMENT NO. 78011F	PROJECT NO. 595600
			TASK NO. F95600	WORK UNIT NO. 20950607
12. PERSONAL AUTHOR(S) Control Data Corporation: Apicella, M. L., Slaton, J., Levi, B.				
13a. TYPE OF REPORT Final Report		13b. TIME COVERED 4 / 1 / 87 - 12 / 31 / 90	14. DATE OF REPORT (Yr., Mo., Day) 1990 September 30	
15. PAGE COUNT 52				
16. SUPPLEMENTARY NOTATION WRDC/MTI Project Priority 6203				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify block no.)	
FIELD	GROUP	SUB GR.		
1308	0905			
19. ABSTRACT (Continue on reverse if necessary and identify block number) This document establishes the design of Function PRE1, "Parcel Application Program", one of the major functions of the Configuration Item "Precompiler" to be built and formally accepted by the ICAM program office. BLOCK 11: INTEGRATED INFORMATION SUPPORT SYSTEM Vol V - Common Data Model Subsystem Part 11 - Neutral Data Manipulation Language (NDML) Precompiler Parse Application Procedure Division Product Specification				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson			22b. TELEPHONE NO. (Include Area Code) (513) 255-7371	22c. OFFICE SYMBOL WRDC/MTI

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

<u>SUBCONTRACTOR</u>	<u>ROLE</u>
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface, and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

		<u>Page</u>
SECTION 1.0	SCOPE	1-1
1.1	Identification	1-1
1.2	Functional Summary	1-1
SECTION 2.0	DOCUMENTS	2-1
2.1	Reference Documents	2-1
2.2	Terms and Abbreviations	2-1
SECTION 3.0	REQUIREMENTS	3-1
3.1	Structural Description	3-1
3.2	Functional Flow	3-1
3.3	Interfaces	3-1
3.3.1	Inputs/Outputs	3-2
3.4	Program Interrupts	3-2
3.5	Timing and Sequencing Description ...	3-2
3.6	Special Control Features	3-2
3.7	Storage Allocation	3-2
3.7.1	Database Definition	3-2
3.7.1.1	File Description	3-2
3.7.1.2	Table Description	3-3
3.7.1.3	Item Description	3-3
3.8	Object Code Creation	3-3
3.9	Adaptation Data	3-3
3.10	Detail Design Description	3-3
3.10.1	Where Include File Used List	3-3
3.10.2	Where External Routine Used List ..	3-7
3.10.3	Main Program Parts List	3-9
3.10.4	Module Documentation	3-11
3.10.5	Include File Description	3-37
3.10.6	Hierarchy Chart	3-39
3.11	Program Listings Comments	3-45
SECTION 4.0	QUALITY ASSURANCE PROVISIONS	4-1
4.1	Introduction and Definitions	4-1
4.2	Computer Programming and Test Evaluation	4-1

SECTION 1

SCOPE

1.1 Identification

This specification establishes the design of Function PRE1, "Parcel Application Program", one of the major functions of the Configuration Item "Precompiler" to be built and formally accepted by the ICAM Program Office. This CI constitutes one of the subsystems of the Common Data Model Processor (CDMP).

1.2 Functional Summary

The purpose of this Computer Program Configuration Item (CPCI) is to partition the input Application Program into parcels that other Precompiler components will access and update.

The following functions will be performed by this CPCI:

1. Determine source code of the Application Program.
2. Determine the Application Program name.
3. Divide the Application Program into the correct number of parcels depending on the program language.



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

SECTION 2

DOCUMENTS

2.1 Reference Documents

1. ICAM Documentation Standards: IDS15012000A, 28 December 1981.
2. D. Appleton Co., CDM Administrators Manual: UM620141000, March 1984.
3. D. Appleton Co., CDM1-IDEF, Model of the Common Data Model; CCS620141000, 15 May 1985.
4. D. Appleton Co., Computer Program Development Specification (DS) for ICAM Integrated Support System (IISS) Configuration Item: NDML Precompiler; DS620141200, October 1984.
5. D. Appleton Co., Embedded NDML Programmer's Reference Manual; PRM620141200, March 1985.
6. Softech Inc., NTM Programmer's Guide: UM620140001, July 1984.
7. Control Data Corp., Computer Program Development Specification (DS) for ICAM Integrated Support System (IISS) Configuration Item: NDDL Command Processor; DS620141100, June 1985.

2.2 Terms and Abbreviations

Attribute Use Class: (AUC)

Conceptual Schema: (CS)

Common Data Model Processor: (CDMP)

Common Data Model: (CDM) Describes common data application process formats, form definitions, etc, of the IISS and includes conceptual schema, external, internal schemas, and schema transformation operators.

Data Field: (DF) An element of data in the external schema. It is by this name that an NDML programmer references data.

Database Management System: (DBMS)

Distributed Request Supervisor: (DRS) This IISS CDM subsystem configuration item controls the execution of distributed NDML queries and non distributed updates.

Domain: A logical definition of legal attribute class values.

Domain Constraint: Predicate that applies to a single domain.

External Schema: (ES)

Forms: Structured views which may be imposed on windows or other forms. A form is composed of fields where each field is a form, item, or window.

Forms Processor: (FP) A set of callable execution time routines available to an application program for form processing.

Internal Schema: (IS)

Integrated Information Support System: (IISS) A test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous databases supported by heterogeneous computers interconnected via a local Area Network.

Mapping: The correspondence of independent objects in two schemas: ES to CS or CS to IS.

Network Transaction Manager: (NTM) Performs the coordination, communication and housekeeping functions required to integrate the application processes and system services resident on the various hosts into a cohesive system.

Neutral Data Manipulation Language: (NDML) A language developed by the IISS project to provide uniform access to common data, regardless of database manager or distribution criteria. It provides distributed retrieved and single node updates.

ORACLE: Relational DBMS based on the SQL (Structured Query Language, a product of ORACLE Corp, Menlo Park, CA). The CDM is an ORACLE database.

Parcel: A sequential file containing sections source code of the input application program.

Request Processor: (RP) A COBOL program that will satisfy a retrieval or update NDML subtransaction against a particular Database Management System.

User Interface: (UI) Controls the user's terminal and interfaces with the rest of the system.

Virtual Terminal Interface: (VTI) Performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by UI software which constitutes the Virtual Terminal Definition. Specific terminals are then mapped against the Virtual Terminal software by specific software modules written for each type of real terminal supported.

SECTION 3

REQUIREMENTS

3.1 Structural Description

A graphic portrayal of this CPCI is included in Section 3.10. This chart shows the hierarchical relationships of each module making up this CPCI.

This CPCI uses a number of lower level modules to handle specific operations such as:

1. Open associated files and parcels (OPNFIL).
2. Read associated files and parcels (INPFIL).
3. Write records to associated files and parcels (OUTFIL).
4. Close associated files and parcels (CLSFIL).

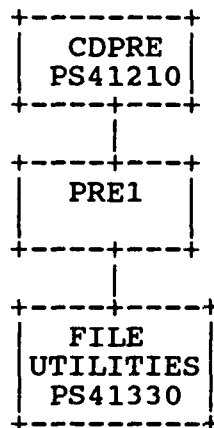
3.2 Functional Flow

This CPCI implements the logic defined in the Development Specification for this CPCI. Details of inputs/outputs and relationships between modules are to found in Section 3.10.

This CPCI has been designated to operate in a batch or interactive mode. It must use the ORACLE DBMS installed on a DEC VAX computer.

3.3 Interfaces

The following diagram depicts the interface of PRE1 with other CPCI's in the system.



3.3.1 Inputs/Outputs

The following table depicts the inputs and outputs of this CPCI. A detail description for each item can be found in the DS for this CPCI.

Function: PRE1

<u>INPUT</u>	<u>OUTPUT</u>
FILE Host	Parcel 1 File Name
Error File Pointer	Parcel 2 File Name
	Parcel 3 File Name
	Parcel 4 File Name
Input-File Pointer	Parcel 1 File Pointer
User Module Name	Parcel 2 File Pointer
Source Language	Parcel 3 File Pointer
EMbedded Language	Parcel 4 File Pointer
	Curley Count
	File Status
	SQL Variable Table
	Function Status

3.4 Program Interrupts

Not applicable to this CPCI.

3.5 Timing and Sequencing Description

This CPCI is called upon by the control module of the precompiler, CDPRE, for every user module encountered in the user's input file of modules to be precompiled.

3.6 Special Control Features

Not applicable to this CPCI.

3.7 Storage Allocation

3.7.1 Database Definition

The database used by this CPCI is the Common Data Model (CDM) database. This model is defined by the CDM1, the IDEF-1 model of the CDM, Reference Document Number 3. The database was constructed using the ORACLE DBMS.

3.7.1.1 File Description

No permanent files have been defined for this CPCI. It uses temporary scratch files for program source code.

3.7.1.2 Table Description

All tables used by this CPCI have been defined by the Development Specification for this CPCI.

3.7.1.3 Item Description

Not applicable to this CPCI.

3.8 Object Code Creation

The object code for this CPCI will be created by the system integration test team by using defined IISS Software Configuration Management procedures. This CPCI will use the COBOL and "C" language compilers.

3.9 Adaptation Data

This CPCI has been coded using ANSI COBOL and a "standard" subset of the "C" language. The intent was to provide a transportable system. Any system environment supporting these languages, a virtual memory management scheme, and the ORACLE Database Management System should be able to support this CPCI. Every possible attempt has been made to localize and identify any machine or environment dependent modules through the original design of the IISS and application of Configuration Management Procedures.

3.10 Detail Design Description

The following sections have been computer generated for this CPCI.

3.10.1 Where Include File Used List

The following lists each include file in the documentation group and all the modules documented in this specification which include them. The purpose of each module is listed as well.

DOCGROUP PS41211 Where-include-file-used List

Include File -----	Module Name -----
ERRCDM	
FILSTAT	CDPRE1
ERRFS	CDPRE1
SBSTLST	CDPRE1
SQLVAR	CDPRE1
ERRPRO	CDPRE1
STDIO	CDPRE1
	CLSERR
	CLSFILE
	CLSINPT
	CLSPAR1
	CLSPAR2
	CLSPAR3
	CLSPAR4
	OPNERR
	OPNFILE
	OPNINPT
	OPNPAR1
	OPNPAR2
	OPNPAR3
	OPNPAR4
	REDINPT
	REDLINE
	REDPAR1
	REDPAR2

DOCGROUP PS41211 Where-include-file-used List

Include File -----	Module Name -----
	REDPAR3
	REDPAR4
	UNPLINE
	WRITERR
	WRTLINE
	WRTPAR1
	WRTPAR2
	WRTPAR3
	WRTPAR4
CTYPE	
	CLSERR
	CLSFILE
	CLSINPT
	CLSPAR1
	CLSPAR2
	CLSPAR3
	CLSPAR4
	OPNERR
	OPNFILE
	OPNINPT
	OPNPAR1
	OPNPAR2
	OPNPAR3
	OPNPAR4
	REDINPT
	REDLINE
	REDPAR1
	REDPAR2
	REDPAR3
	REDPAR4
	UNPLINE

DOCGROUP PS41211 Where-include-file-used List

Include File -----	Module Name -----
	WRITERR
	WRTLINE
	WRTPAR1
	WRTPAR2
	WRTPAR3
	WRTPAR4
FCBSTRC	
	CLSERR
	CLSFILE
	CLSINPT
	CLSPAR1
	CLSPAR2
	CLSPAR3
	CLSPAR4
	OPNERR
	OPNFILE
	OPNINPT
	OPNPAR1
	OPNPAR2
	OPNPAR3
	OPNPAR4
	REDINPT
	REDLINE
	REDPAR1
	REDPAR2
	REDPAR3
	REDPAR4
	UNPLINE
	WRITERR
	WRTLINE
	WRTPAR1

DOCGROUP PS41211 Where-include-file-used List

Include File -----	Module Name -----
	WRTPAR2 WRTPAR3 WRTPAR4

3.10.2 Where External Routine Used List

The following lists each external function or routine in the documentation group and all the documented modules which call it. The purpose of each module is listed as well.

DOCGROUP PS41211 Where-external-routine-used List

System Module -----	Module Name -----
GENFIL	CDPRE1
OPNFIL	CDPRE1
PROEXEC	CDPRE1
PROCVAR	CDPRE1
CURLCNT	CDPRE1
INPFIL	CDPRE1
OUTFIL	CDPRE1
TOKEN	CDPRE1
CDMACR	CDPRE1
RPTERR	CDPRE1
ERRPRO	CDPRE1
ESCPY	OPNERR OPNFILE OPNINPT OPNPAR1 OPNPAR2 OPNPAR3 OPNPAR4
ISUPPER	

DOCGROUP PS41211 Where-external-routine-used List

System Module -----	Module Name -----
	OPNINPT
TOLOWER	OPNINPT
FOPEN	OPNERR OPNFILE OPNINPT OPNPAR1 OPNPAR2 OPNPAR3 OPNPAR4
GETC	REDLINE
PUTC	WRTLINE
FCLOSE	CLSERR CLSFILE CLSINPT CLSPAR1 CLSPAR2 CLSPAR3 CLSPAR4
FSEEK	UNPLINE
STRNCPY	UNPLINE

3.10.3 Main Program Parts List

The following lists each Main Program in the documentation group and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more than once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external routine". The Purpose of the Main Program module is listed as well.

DOCGROUP PS41211 Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
CDPRE1	GENFIL	External routine
	OPNFIL	External routine
	PROEXEC	External routine
	PROCVAR	External routine
	CURLCNT	External routine
	UNPLINE	Well-defined module
	INPFIL	External routine
	OUTFIL	External routine
	TOKEN	External routine
	CDMACR	External routine
	RPTERR	External routine
	ERRPRO	External routine
CLSERR		
	FCLOSE	External routine
CLSFILE		
	FCLOSE	External routine
CLSINPT		
	FCLOSE	External routine
CLSPAR1		
	FCLOSE	External routine
CLSPAR2		
	FCLOSE	External routine
CLSPAR3		
	FCLOSE	External routine
CLSPAR4		
	FCLOSE	External routine
OPNERR		
	ESCPY	External routine
	FOPEN	External routine
OPNFILE		

DOCGROUP PS41211 Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
OPNINPT	ESCPY	External routine
	FOPEN	External routine
OPNPAR1	ESCPY	External routine
	ISUPPER	External routine
	TOLOWER	External routine
	FOPEN	External routine
OPNPAR2	ESCPY	External routine
	FOPEN	External routine
OPNPAR3	ESCPY	External routine
	FOPEN	External routine
OPNPAR4	ESCPY	External routine
	FOPEN	External routine
REDINPT	ESCPY	External routine
	FOPEN	External routine
REDLINE	REDLINE	Well-defined module
REDPAR1	GETC	External routine
REDPAR2	REDLINE	External routine
REDPAR3	REDLINE	External routine
REDPAR4	REDLINE	External routine
	REDLINE	External routine

DOCGROUP PS41211 Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
UNPLINE	FSEEK	External routine
	STRNCPY	External routine
WRITERR		
	WRTLINE	Well-defined module
WRTLINE		
	PUTC	External routine
WRTPAR1		
	WRTLINE	External routine
WRTPAR2		
	WRTLINE	External routine
WRTPAR3		
	WRTLINE	External routine
WRTPAR4		
	WRTLINE	External routine

3.10.4 Module Documentation

The following documentation describes information which is specific to each individual module in the documentation group being documented in this specification. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

NAME: Name of program Module.

PURPOSE: Purpose of Module as detailed in the source code.

LANGUAGE: Programming language source code is written in.
The choices are:
VAX-11 FORTRAN
C (I/S-1 Workbench 'C')
VAX-11 COBOL

MODULE TYPE: Whether a Program, Subroutine, or Function.

SOURCE FILE: Name of Source File from file specification.

SOURCE FILE TYPE: Source File Extension from file specification.

HOST: Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.

SUBSYSTEM: IISS sub-system this file resides in.

SUBDIRECTORY: Sub-directory of that subsystem in which this file resides.

DOCUMENTATION GROUP: Name of documentation group of which this source file is a member.

DESCRIPTION: A description of the module as obtained from the source code.

ARGUMENTS: The arguments with which this routine is called if it is a Subroutine or a Function.

INCLUDE FILES: A list of all the files that are included into this module as well as their purposes.

ROUTINES CALLED: Subroutines or Functions, either documented or external, called by this module, if any.

CALLED DIRECTLY BY: The documented routines which call this module, if any.

USED IN MAIN PROGRAM(S): The documented Main Programs which contain this module in their parts list according to the list in section 3.10.3.

The Module Documentation is arranged alphabetically according to Module Name.

DOCGROUP PS41211 Module Documentation

NAME: CDPRE1
PURPOSE: PARCEL APPLICATION PROGRAM
LANGUAGE: VAX-11 COBOL
SOURCE FILE: CDPRE1
SOURCE FILE TYPE: COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: NDML

DESCRIPTION:

THIS PROGRAM EXTRACTS THE PROGRAM ID FROM THE APPLICATION,
AND PARTITIONS AN INPUT AP INTO FOUR PARCELS THAT WILL
BE ADDED TO BY OTHER PRECOMPILER COMPONENTS.
MODIFIED 10/88 - RELEASE 2.5
CHANGED FORTRAN TO OPEN ALL PARCELS FOR USE AND WRITE
THE USER VARIABLE DECLARATIONS TO THE 3RD PARCEL (WS)
DELETED IOS-SECTION PARAMETER - 2 MAY 1989
Modified 7/89 - Release 2.7
Changed to allow C precompiles, as well as modifying
to allow for embedded SQL precompiles.

ARGUMENTS:

FCB-I DSPLY[S9(9)]
FILE-HOST DSPLY[X(3)]
FCB-E DSPLY[S9(9)]
PARCL1 DSPLY[X(80)]
PARCL2 DSPLY[X(80)]
PARCL3 DSPLY[X(80)]
PARCL4 DSPLY[X(80)]
FCB-1 DSPLY[S9(9)]
FCB-2 DSPLY[S9(9)]
FCB-3 DSPLY[S9(9)]
FCB-4 DSPLY[S9(9)]
USER-MODULE-NAME DSPLY[X(10)]
SOURCE-LANGUAGE DSPLY[X(10)]
EMBEDDED-LANGUAGE DSPLY[X(10)]
CURLEY-COUNT DSPLY[S9(9)]
FILE-STATUS DSPLY[S9(9)]
SQL-VARIABLE-TABLE RECRD
RET-STATUS DSPLY[X(5)]

INCLUDE FILES:

ERRCDM
FILSTAT
ERRFS
SBSTLST
SQLVAR
ERRPRO

ROUTINES CALLED:

GENFIL
OPNFIL
PROEXEC
PROCVAR
CURLCNT
UNPLINE
INPFIL
OUTFIL
TOKEN
CDMACR
RPTERR
ERRPRO

DOCGROUP PS41211 Module Documentation

NAME: OPNINPT
PURPOSE: OPEN THE FILE CONTAING THE USERS APPLICATION PROGRAMS
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

MODIFIED 04/01/86 TO NULL TERMINATE THE FILENAME ARGUMENTS

SYNOPSIS

C -- OPINPT(FILE_NAME, MODE, RET_STATUS) ;
COBOL -- CALL "OPNINPT" USING FILE-NAME,
MODE,
RET-CODE.

FORTTRAN -- CALL OPNINPT(FILENAME, MODE, RETCODE)

INPUT:

CHAR *FILE_NAME ;
CHAR *MODE ;

OUTPUT:

INT *RET_STATUS ;
DESCRIPTION

ARGUMENTS:

FILE_NAME CHAR *
MODE CHAR *
STATUS INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

ESCPY
ISUPPER
TOLOWER
FOPEN

DOCGROUP PS41211 Module Documentation

NAME: OPNERR
PURPOSE: OPEN THE ERROR FILE FOR THIS PRECOMPILATION
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- OPNERR(FILE_NAME, MODE, RET_STATUS) ;

COBOL -- CALL "OPNERR" USING FILE-NAME,
MODE,
RET-STATUS.

FORTRAN -- CALL OPNERR(FILENAME, MODE, RETCODE)

INPUT:

CHAR *FILE_NAME ;
CHAR *MODE ;

OUTPUT:

INT *RET_STATUS ;

DESCRIPTION

ARGUMENTS:

FILE_NAME	CHAR *
MODE	CHAR *
STATUS	INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

ESCPY
FOPEN

DOCGROUP PS41211 Module Documentation

NAME: OPNPAR1

PURPOSE: OPEN THE FILE CONTAINING PARCEL 1 OF THE AP

LANGUAGE: C

SOURCE FILE: NDMLCIO

SOURCE FILE TYPE: C

HOST:

SUBSYSTEM: CDM

SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- OPNPAR1(FILE_NAME, MODE, RET_STATUS) ;

COBOL -- CALL "OPNPAR1" USING FILE-NAME,
MODE,
RET-STATUS.

FORTRAN -- CALL OPNPAR1(FILENAME, MODE, RETCODE)

INPUT:

CHAR *FILE_NAME ;

CHAR *MODE ;

OUTPUT:

INT *RET_STATUS ;

DESCRIPTION

ARGUMENTS:

FILE_NAME

CHAR *

MODE

CHAR *

STATUS

INT *

INCLUDE FILES:

STDIO

CTYPE

FCBSTRC

ROUTINES CALLED:

ESCPY
FOPEN

DOCGROUP PS41211 Module Documentation

NAME: OPNPAR2
PURPOSE: OPEN THE FILE CONTAINING PARCEL 2 OF THE AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- OPNPAR2(FILE_NAME, MODE, RET_STATUS) ;

COBOL -- CALL "OPNPAR2" USING FILE-NAME,
 MODE,
 RET-STATUS.

FORTRAN -- CALL OPNPAR2(FILENAME, MODE, RETCODE)

INPUT:

CHAR *FILE_NAME ;
CHAR *MODE ;

OUTPUT:

INT *RET_STATUS ;

DESCRIPTION

ARGUMENTS:

FILE_NAME CHAR *
MODE CHAR *
STATUS INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

ESCPY
FOPEN

DOCGROUP PS41211 Module Documentation

NAME: OPNPAR3
PURPOSE: OPEN THE FILE CONTAINING PARCEL 3 OF THE AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
      C      -- OPNPAR3(FILE_NAME, MODE, RET_STATUS) ;  
  
      COBOL  -- CALL "OPNPAR3" USING    FILE-NAME,  
                      MODE,  
                      RET-STATUS.  
      FORTRAN -- CALL OPNPAR3(FILENAME, MODE, RETCODE)
```

INPUT:

```
CHAR  *FILE_NAME ;  
CHAR  *MODE_;
```

OUTPUT:

```
INT   *RET_STATUS ;  
DESCRIPTION
```

ARGUMENTS:

FILE_NAME	CHAR *
MODE	CHAR *
STATUS	INT *

INCLUDE FILES:

```
STDIO  
CTYPE  
FCBSTRC
```

ROUTINES CALLED:

```
ESCPY  
FOPEN
```

DOCGROUP PS41211 Module Documentation

NAME: OPNPAR4

PURPOSE: OPEN THE FILE CONTAINING PARCEL 4 OF THE AP

LANGUAGE: C

SOURCE FILE: NDMLCIO

SOURCE FILE TYPE: C

HOST:

SUBSYSTEM: CDM

SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
      C      -- OPNPAR4(FILE_NAME, MODE, RET_STATUS) ;
```

COBOL -- CALL "OPNPAR4" USING FILE-NAME,
MODE,
RET-STATUS.
FORTRAN -- CALL OPNPAR4(FILENAME, MODE, RETCODE)

INPUT:
CHAR *FILE_NAME ;
CHAR *MODE ;
OUTPUT:
INT *RET_STATUS ;
DESCRIPTION

ARGUMENTS:

FILE_NAME CHAR *
MODE CHAR *
STATUS INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

ESCPY
FOPEN

DOCGROUP PS41211 Module Documentation

NAME: OPNFILE
PURPOSE: OPENS NAMED FILE FOR INPUT
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- OPNFILE(FILE_NAME, MODE, FILE_POINTER,
RET_STATUS) ;

COBOL -- CALL "OPNFILE" USING FILE-NAME,
MODE,
FILE-POINTER,
RET-STATUS.

FORTTRAN -- CALL OPNFILE(FILENAME, MODE, FILEPTR,
RETCODE)
INPUT:
CHAR *FILE_NAME ;
CHAR *MODE ;
INT *FILE_POINTER;
OUTPUT
INT *RET_STATUS ;
DESCRIPTION

ARGUMENTS:

FILE_NAME CHAR *
MODE CHAR *
FILE_POINTER FILE *
STATUS INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

ESCPY
FOPEN

DOCGROUP PS41211 Module Documentation

NAME: REDINPT
PURPOSE: READ A LINE IN THE FILE CONTAINING USERS APPLICATION
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- REDINPT(LINE, LINE_LEN, RET_STATUS) ;
COBOL -- CALL "REDINPT" USING LINE,
LINE-LEN,
RET-STATUS.

FORTTRAN -- CALL REDINPT(LINE, LINE-LEN, RETCODE)

INPUT:
CHAR *LINE ;
INT *LINE_LEN ;
OUTPUT:
INT *RET_STATUS ;
DESCRIPTION

ARGUMENTS:

LINE CHAR *
LINE_LEN INT *
STATUS INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

REDLINE

DOCGROUP PS41211 Module Documentation

NAME: REDPAR1
PURPOSE: READ A LINE IN THE FILE CONTAINING PARCEL 1 OF THE AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- REDPAR1(LINE, LINE_LEN, RET_STATUS) ;

COBOL -- CALL "REDPAR1" USING LINE,
LINE-LEN,
RET-STATUS.

FORTTRAN -- CALL REDPAR1(LINE, LINE-LEN, RETCODE)

INPUT:
CHAR *LINE ;
INT *LINE_LEN ;
OUTPUT:
INT *RET_STATUS ;
DESCRIPTION

ARGUMENTS:

```
-----  
LINE                CHAR *  
LINE_LEN            INT *  
STATUS              INT *
```

INCLUDE FILES:

```
-----  
STDIO  
CTYPE  
FCBSTRC
```

ROUTINES CALLED:

```
-----  
REDLINE
```

DOCGROUP PS41211 Module Documentation

NAME: REDPAR2
PURPOSE: READ A LINE IN THE FILE CONTAINING PARCEL 2 OF THE AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
      C      -- REDPAR2(LINE, LINE_LEN, RET_STATUS) ;  
  
      COBOL  -- CALL "REDPAR2" USING      LINE,  
                        LINE-LEN,  
                        RET-STATUS.  
  
      FORTRAN -- CALL REDPAR2(LINE, LINE-LEN, RETCODE)
```

INPUT:

```
CHAR      *LINE ;  
INT       *LINE_LEN ;  
OUTPUT:  
INT       *RET_STATUS ;  
DESCRIPTION
```

ARGUMENTS:

```
-----  
LINE                CHAR *  
LINE_LEN            INT *  
STATUS              INT *
```

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

REDLINE

DOCGROUP PS41211 Module Documentation

NAME: REDPAR3

PURPOSE: READ A LINE IN THE FILE CONTAINING PARCEL 3 OF THE AP

LANGUAGE: C

SOURCE FILE: NDMLCIO

SOURCE FILE TYPE: C

HOST:

SUBSYSTEM: CDM

SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- REDPAR3(LINE, LINE_LEN, RET_STATUS) ;

COBOL -- CALL "REDPAR3" USING LINE,
LINE-LEN,
RET-STATUS.

FORTRAN -- CALL REDPAR3(LINE, LINE-LEN, RETCODE)

INPUT:

CHAR *LINE ;

INT *LINE_LEN ;

OUTPUT:

INT *RET_STATUS ;

DESCRIPTION

ARGUMENTS:

LINE

CHAR *

LINE_LEN

INT *

STATUS

INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

REDLINE

DOCGROUP PS41211 Module Documentation

NAME: REDPAR4
PURPOSE: READ A LINE IN THE FILE CONTAINING PARCEL 4 OF THE AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- REDPAR4(LINE, LINE_LEN, RET_STATUS) ;

COBOL -- CALL "REDPAR4" USING LINE,
LINE-LEN,
RET-STATUS.

FORTRAN -- CALL REDPAR4(LINE, LINE-LEN, RETCODE)
INPUT:
CHAR *LINE ;
INT *LINE_LEN ;
OUTPUT:
INT *RET_STATUS ;
DESCRIPTION

ARGUMENTS:

LINE CHAR *
LINE_LEN INT *
STATUS INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

REDLINE

DOCGROUP PS41211 Module Documentation

NAME: REDLINE
PURPOSE: READ A LINE IN THE NAMED FILE
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
      C      -- REDLINE(LINE, LINE_LEN, RET_STATUS) ;  
  
      COBOL  -- CALL "REDLINE" USING      LINE,  
                        LINE-LEN,  
                        RET-STATUS.  
      FORTRAN -- CALL REDLINE(LINE, LINE-LEN, RETCODE)  
INPUT:  
CHAR      *LINE ;  
INT       *LINE_LEN ;  
OUTPUT:  
INT       *RET_STATUS ;  
DESCRIPTION
```

ARGUMENTS:

FILE_POINTER	FILE *
LINE	CHAR *
LINE_LEN	INT *
STATUS	INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

GETC

DOCGROUP PS41211 Module Documentation

NAME: WRITERR
PURPOSE: WRITE A LINE TO THE ERROR FILE OF THIS PRECOMPILATION
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- WRITERR(LINE, LINE_LEN) ;

COBOL -- CALL "WRITERR" USING LINE,
 LINE-LEN.
FORTRAN -- CALL WRITERR(LINE, LINE-LEN)

INPUT:

CHAR *LINE ;
INT *LINE_LEN ;

OUTPUT:
 NONE

DESCRIPTION

ARGUMENTS:

LINE	CHAR *
LINE_LEN	INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

WRTLINE

DOCGROUP PS41211 Module Documentation

NAME: WRTPAR4
PURPOSE: WRITE A LINE TO THE FILE CONTAINING PARCEL 4 OF THE
 AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
      C      -- WRTPAR4(LINE, LINE_LEN) ;  
  
      COBOL  -- CALL "WRTPAR4" USING    LINE,  
                LINE-LEN.  
      FORTRAN  -- CALL WRTPAR4(LINE, LINE-LEN )
```

INPUT:

```
CHAR    *LINE ;  
INT     *LINE_LEN ;
```

OUTPUT:

NONE

DESCRIPTION

ARGUMENTS:

```
-----  
LINE                                CHAR *  
LINE_LEN                           INT *
```

INCLUDE FILES:

```
-----  
STDIO  
CTYPE  
FCBSTRC
```

ROUTINES CALLED:

WRTLINE

DOCGROUP PS41211 Module Documentation

NAME: WRTPAR1

PURPOSE: WRITE A LINE TO THE FILE CONTAINING PARCEL 1 OF THE
AP

LANGUAGE: C

SOURCE FILE: NDMLCIO

SOURCE FILE TYPE: C

HOST:

SUBSYSTEM: CDM

SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
      C      -- WRTPAR1(LINE, LINE_LEN) ;
```

```
      COBOL  -- CALL "WRTPAR1"  USING      LINE,
                    LINE-LEN.
      FORTRAN  --  CALL WRTPAR1(LINE, LINE-LEN )
INPUT:
CHAR      *LINE ;
INT       *LINE_LEN ;
OUTPUT:
          NONE
DESCRIPTION
```

ARGUMENTS:

```
-----
LINE      CHAR *
LINE_LEN  INT  *
```

INCLUDE FILES:

```
-----
STDIO
CTYPE
FCBSTRC
```

ROUTINES CALLED:

```
-----
WRTLINE
```

DOCGROUP PS41211 Module Documentation

NAME: WRTPAR2
PURPOSE: WRITE A LINE TO THE FILE CONTAINING PARCEL 2 OF THE
 AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
      C      -- WRTPAR2(LINE, LINE_LEN) ;

      COBOL  -- CALL "WRTPAR2"  USING      LINE,
                    LINE-LEN.
      FORTRAN  --  CALL WRTPAR2(LINE, LINE-LEN )
INPUT:
CHAR      *LINE ;
INT       *LINE_LEN ;
OUTPUT:
```

DESCRIPTION NONE

ARGUMENTS:

LINE	CHAR *
LINE_LEN	INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

WRTLINE

DOCGROUP PS41211 Include Documentation

NAME: WRTPAR3

PURPOSE: WRITE A LINE TO THE FILE CONTAINING PARCEL 3 OF THE
AP

LANGUAGE: C

SOURCE FILE: NDMLCIO

SOURCE FILE TYPE: C

HOST:

SUBSYSTEM: CDM

SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- WRTPAR3 (LINE, LINE_LEN) ;

COBOL -- CALL "WRTPAR3" USING LINE,
 LINE-LEN.

FORTTRAN -- CALL WRTPAR3 (LINE, LINE-LEN)

INPUT:

CHAR *LINE ;

INT *LINE_LEN ;

OUTPUT:

NONE

DESCRIPTION

ARGUMENTS:

LINE	CHAR *
LINE_LEN	INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

WRTLINE

DOCGROUP PS41211 Module Documentation

NAME: WRTLINE
PURPOSE: WRITE A LINE TO THE NAMED FILE
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C	--	WRTLINE(FILE_POINTER, LINE, LINE_LEN) ;
COBOL	--	CALL "WRTLINE" USING FILE-POINTER, LINE, LINE-LEN.
FORTTRAN	--	CALL WRTLINE(FILEPTR,LINE, LINE-LEN)

INPUT:

	INT	*FILE_POINTER ;
CHAR	*LINE ;	
INT	*LINE_LEN ;	

OUTPUT:

NONE

DESCRIPTION

ARGUMENTS:

FILE_POINTER	FILE *
LINE	CHAR *
LINE_LEN	INT *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

PUTC

DOCGROUP PS41211 Module Documentation

NAME: CLSINPT
PURPOSE: CLOSE THE FILE CONTAINING THE USERS AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- CLSINPT () ;

COBOL -- CALL "CLSINPT".

FORTRAN -- CALL CLSINPT.

INPUT:

NONE

OUTPUT:

NONE

DESCRIPTION

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

FCLOSE

DOCGROUP PS41211 Module Documentation

NAME: CLSERR
PURPOSE: CLOSE THE FILE CONTAINING THE ERROR LISTING
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- CLSERR () ;

COBOL -- CALL "CLSERR".
FORTRAN -- CALL CLSERR.

INPUT:

NONE

OUTPUT:

NONE

DESCRIPTION

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

FCLOSE

DOCGROUP PS41211 Module Documentation

NAME: CLSPAR1
PURPOSE: CLOSE THE FILE CONTAINING PARCEL 1 OF THE AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
C          -- CLSPAR1 ( ) ;

          COBOL -- CALL "CLSPAR1".
          FORTRAN -- CALL CLSPAR1.
```

INPUT:

NONE

OUTPUT:

NONE

DESCRIPTION

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

FCLOSE

DOCGROUP PS41211 Module Documentation

NAME: CLSPAR2

PURPOSE: CLOSE THE FILE CONTAINING PARCEL 2 OF THE AP

LANGUAGE: C

SOURCE FILE: NDMLCIO

SOURCE FILE TYPE: C

HOST:

SUBSYSTEM: CDM

SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
C          -- CLSPAR2 ( ) ;

          COBOL -- CALL "CLSPAR2".
          FORTRAN -- CALL CLSPAR2.
```

INPUT:

NONE

OUTPUT:

NONE

DESCRIPTION

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

FCLOSE

DOCGROUP PS41211 Module Documentation

NAME: CLSPAR3
PURPOSE: CLOSE THE FILE CONTAINING PARCEL 3 OF THE AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- CLSPAR3 () ;

COBOL -- CALL "CLSPAR3".
FORTRAN -- CALL CLSPAR3.

INPUT:

NONE

OUTPUT:

NONE

DESCRIPTION

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

FCLOSE

DOCGROUP PS41211 Module Documentation

NAME: CLSPAR4
PURPOSE: CLOSE THE FILE CONTAINING PARCEL 4 OF THE AP
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

C -- CLSPAR4 () ;

COBOL -- CALL "CLSPAR4".

FORTTRAN -- CALL CLSPAR4.

INPUT:

NONE

OUTPUT:

NONE

DESCRIPTION

INCLUDE FILES:

STDIO

CTYPE

FCBSTRC

ROUTINES CALLED:

FCLOSE

DOCGROUP PS41211 Module Documentation

NAME: CLSFILE
PURPOSE: CLOSE THE NAMED FILE
LANGUAGE: C
SOURCE FILE: NDMLCIO
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
C          -- CLSFILE(FILE_POINTER) ;

          COBOL -- CALL "CLSFILE" USING FILE-POINTER.
          FORTRAN -- CALL CLSFILE(FILEPTR)
```

INPUT:

INT *FILE_POINTER ;

OUTPUT:

NONE

DESCRIPTION

ARGUMENTS:

FILE_POINTER

INT *

INCLUDE FILES:

STDIO

CTYPE

FCBSTRC

ROUTINES CALLED:

FCLOSE

DOCGROUP PS41211 Module Documentation

NAME: UNPLINE

PURPOSE: THIS ROUTINE UNPUTS A LINE BACK TO INPUT_FP

LANGUAGE: C

SOURCE FILE: NDMLCIO

SOURCE FILE TYPE: C

HOST:

SUBSYSTEM: CDM

SUBDIRECTORY: SHARE

DESCRIPTION:

SYNOPSIS

```
C          -- UNPLINE(FCB, LINE, BUF_LEN, REC_LEN,
RET_STATUS) ;
```

```
          COBOL -- CALL "UNPLINE" USING FCB,
                                         LINE,
```

```

                                BUF-LEN,
                                REC-LEN,
                                RET-STATUS.
      FORTRAN  --  CALL UNPLINE(FCB, LINE, BUFLen, RECLen,
RETCODE)
INPUT:
      INT      *FCB;
CHAR    *LINE ;
LONG    *BUF_LEN ;
LONG    *REC_LEN ;
OUTPUT:
CHAR    *RET_STATUS ;
DESCRIPTION
```

ARGUMENTS:

FCB	FCB **
INREC	CHAR *
BUF_LEN_IN	INT *
REC_LEN_IN	INT *
RET_STATUS	CHAR *

INCLUDE FILES:

STDIO
CTYPE
FCBSTRC

ROUTINES CALLED:

FSEEK
STRNCPY

3.10.5 Include File Descriptions

The following list contains a purpose and description of each include file in the documentation group as specified in the source code. The language it is written in is also given.

DOCGROUP PS41211 Include File Description

FILE NAME: ERRCDM
PURPOSE: IISS ERROR STATUS CODES FOR CDMP MODULES
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

CONTAINS ALL ERROR CODES USED BY CDMP *

DOCGROUP PS41211 Include File Description

FILE NAME: ERRFS
PURPOSE: ERRFS.INC - FILE I/O PRIMITIVES (FILE SERVICES)
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

DOCGROUP PS41211 Include File Description

FILE NAME: ERRPRO
PURPOSE: PROCESS ERROR INCLUDE FILE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

DOCGROUP PS41211 Include File Description

FILE NAME: FILSTAT
PURPOSE: VARIABLE DEFINITION FOR FILE STATUS
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

FILE USAGE FILE STATUS PARAMETER

FILSTAT

FILE USAGE FILE STATUS PARAMETER
SIZE AND THE 88 VALUE ARE PROBABLY MACHINE

DEPENDENT
(THIS IS FOR VAX-11 COBOL)

DOCGROUP PS41211 Include File Description

FILE NAME: SBSTLST
PURPOSE: WS DEFINITION FOR THE SUBSTITUTION LIST TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

SUBSTITUTION-LIST REPRESENTS THE INPUT TABLE
OF SUBSTITUTION PARAMETERS FOR THE CDMACR
MACRO EXPANSION SUBROUTINE

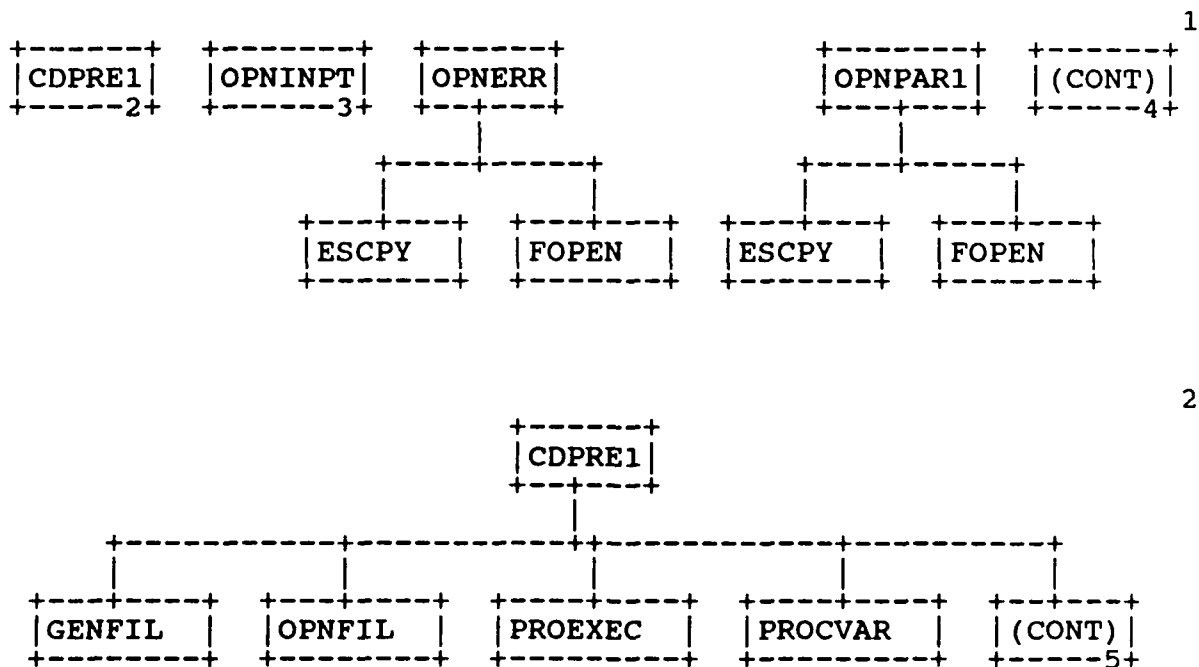
DOCGROUP PS41211 Include File Description

FILE NAME: SQLVAR
PURPOSE: SQL VARIABLE TABLE
LANGUAGE: VAX-11 COBOL

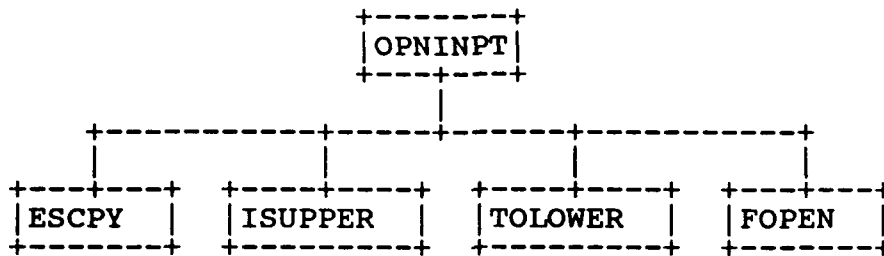
DESCRIPTION:

THIS TABLE HOLDS THE VARIABLES DECLARED IN THE
EXEC SQL BEGIN DECLARE SECTION.

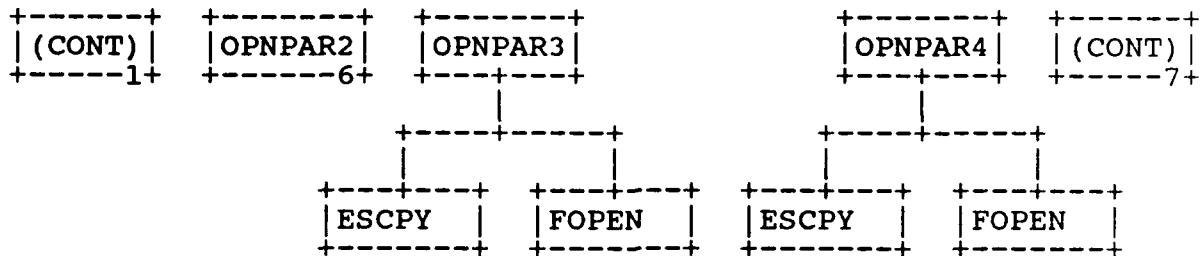
3.10.6 Hierarchy Chart



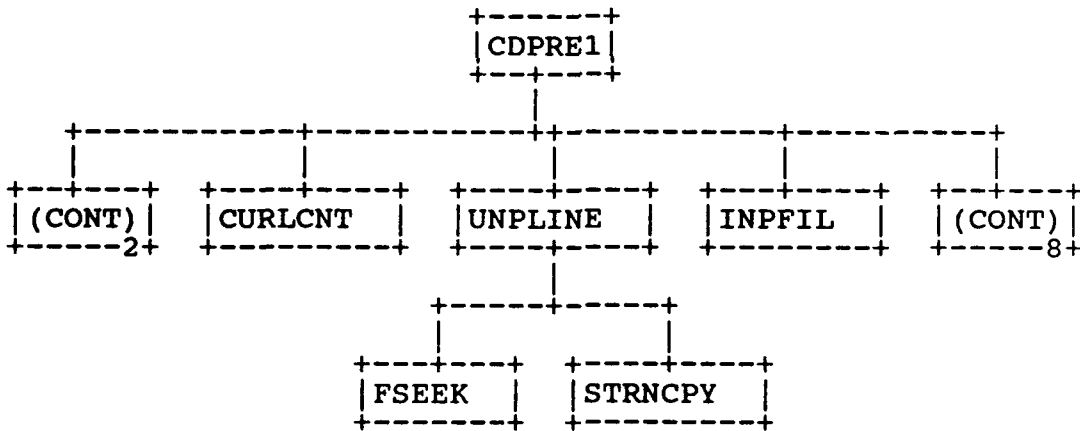
3



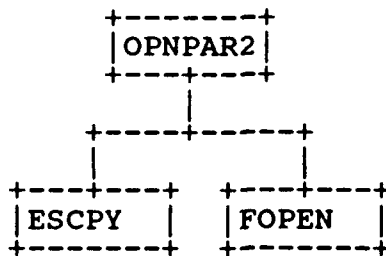
4

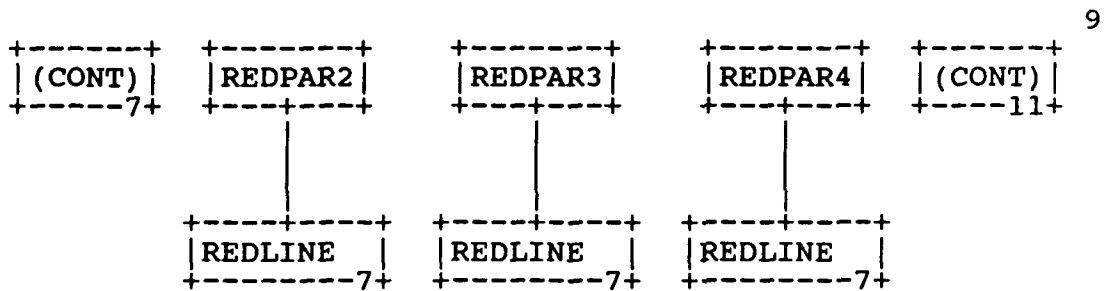
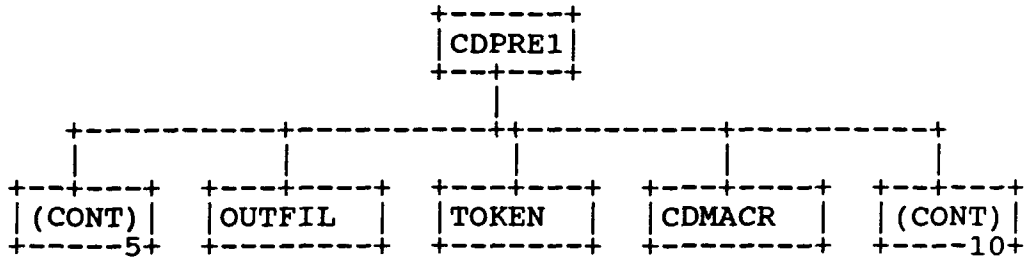
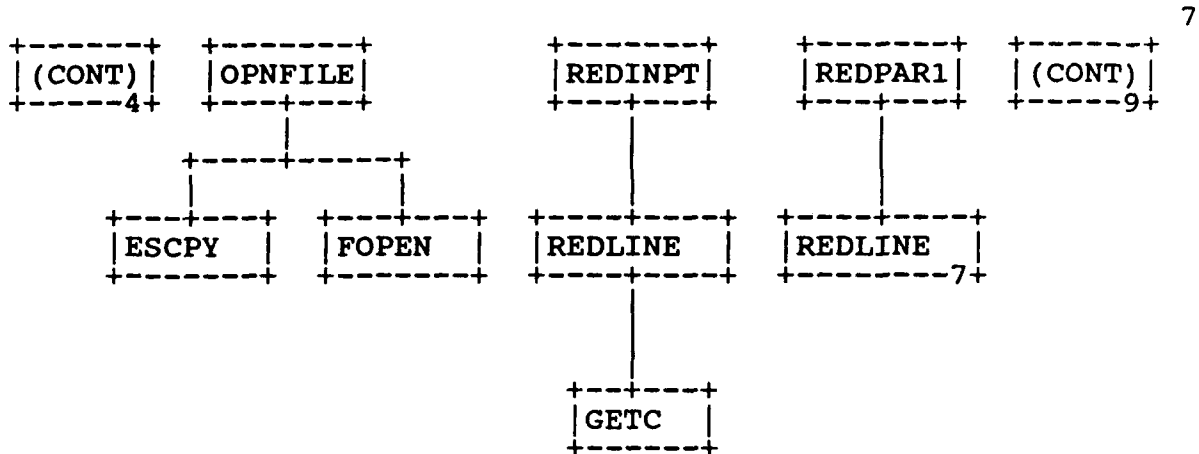


5

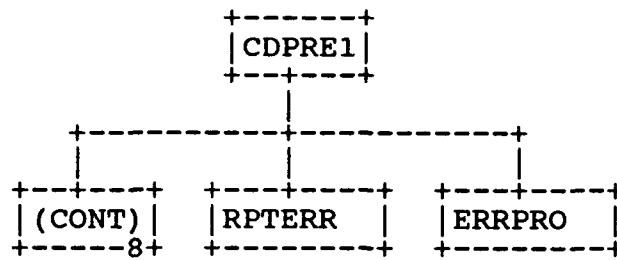


6

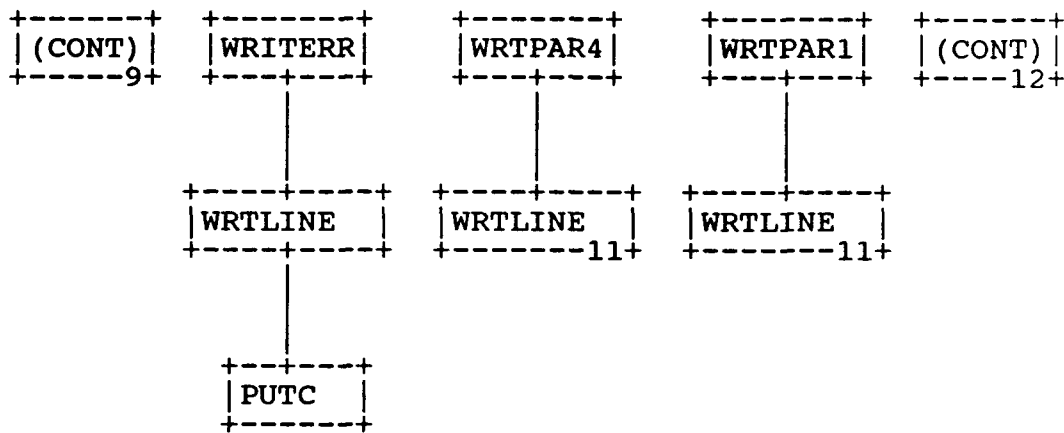




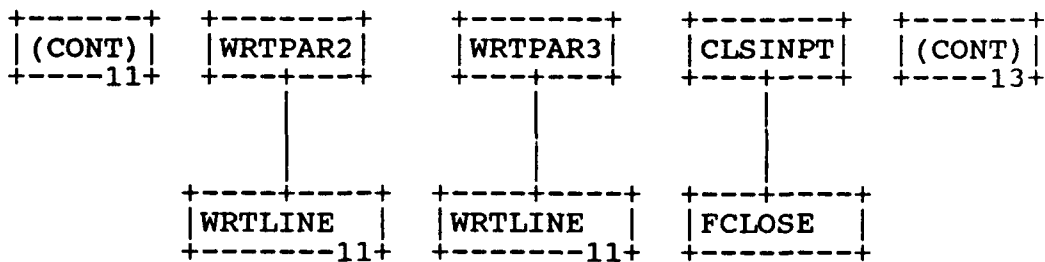
10

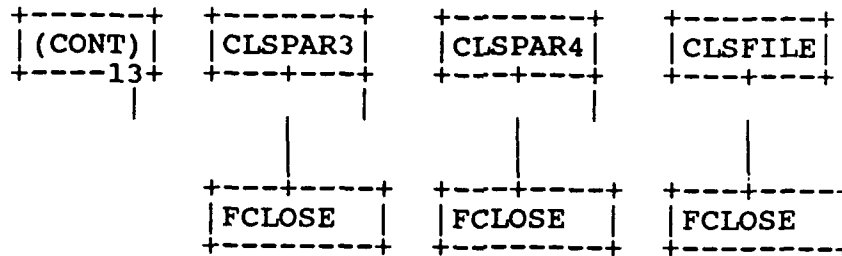
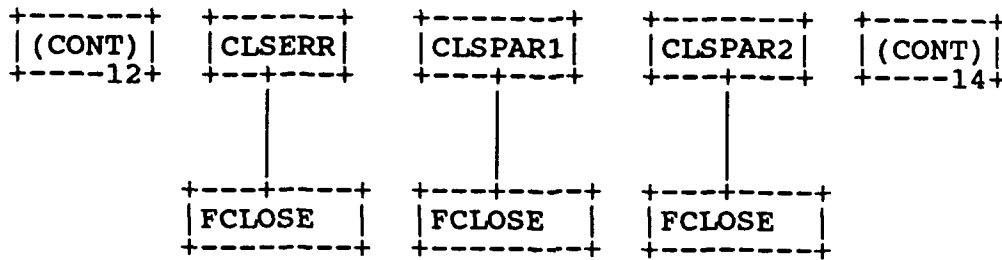


11



12





CDMACR
CDPRE1.....2
CLSERR.....13
CLSFILE.....14
CLSINPT.....12
CLSPAR1.....13
CLSPAR2.....13
CLSPAR3.....14
CLSPAR4.....14
CURLCNT
ERRPRO
ESCPY
FCLOSE
FOPEN
FSEEK
GENFIL
GETC
INPFIL
ISUPPER
OPNERR.....1
OPNFIL
OPNFILE.....7
OPNINPT.....3
OPNPAR1.....1
OPNPAR2.....6
OPNPAR3.....4
OPNPAR4.....4
OUTFIL
PROCVAR
PROEXEC
PUTC

WRTPAR4.....11

PS 620341211
30 September 1990

REDINPT.....7
REDLINE7
REDPAR1.....7
REDPAR2.....9
REDPAR3.....9
REDPAR4.....9
RPTERR
STRNCPY
TOKEN
TOLOWER
UNPLINE5
WRITERR.....11
WRTLINE11
WRTPAR1.....11
WRTPAR2.....12
WRTPAR3.....12

3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.

SECTION 4

QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."