

AD-A181 674

INTEGRATED INFORMATION SUPPORT SYSTEM (IIS) VOLUME 5 1/3

COMMON DATA MODEL 5 (U) GENERAL ELECTRIC CO

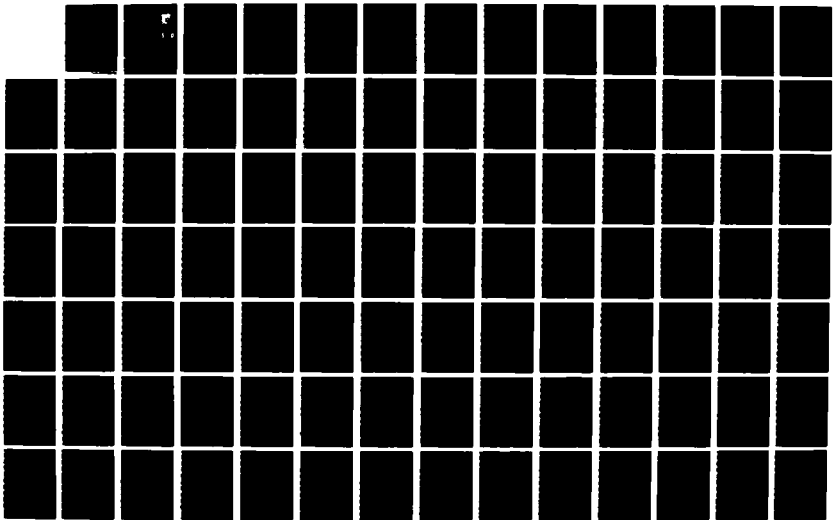
SCHENECTADY NY PRODUCTION RESOURCES CONSU

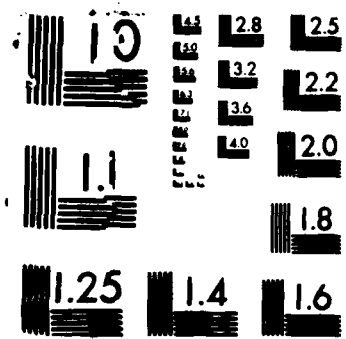
UNCLASSIFIED

5 SINGH ET AL 01 NOV 85 PS-620141100

F/G 9/2

NL



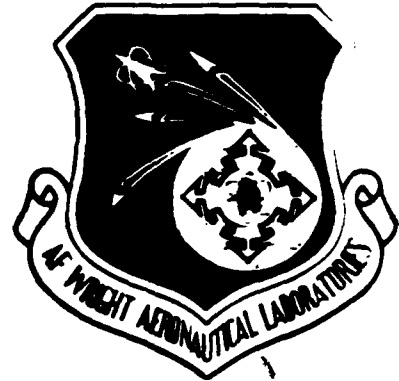


**DTIC FILE COPY**

2

AFWAL-TR-86-4006  
Volume V  
Part 6

**AD-A181 674**



INTEGRATED INFORMATION  
SUPPORT SYSTEM (IISS)  
Volume V - Common Data Model Subsystem  
Part 6 - NDDL Processor Product Specification  
Section 3.10.8 CPFONE through INITRDL

General Electric Company  
Production Resources Consulting  
One River Road  
Schenectady, New York 12345

DTIC  
ELECTE  
JUN 25 1987  
S D

Final Report for Period 22 September 1980 - 31 July 1985  
November 1985

Approved for public release; distribution is unlimited.

MATERIALS LABORATORY  
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES  
AIR FORCE SYSTEMS COMMAND  
WRIGHT-PATTERSON AFB, OH 45433-6533


87 6 24 044

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, and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto


This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public including foreign nations.

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

  
\_\_\_\_\_  
DAVID L. JUDSON, PROJECT MANAGER  
AFWAL/MLTC  
WRIGHT PATTERSON AFB OH 45433

5 Aug 1986  
\_\_\_\_\_  
DATE

FOR THE COMMANDER.

  
\_\_\_\_\_  
GERALD C. SHUMAKER, BRANCH CHIEF  
AFWAL/MLTC  
WRIGHT PATTERSON AFB OH 45433

7 Aug 86  
\_\_\_\_\_  
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 AFWAL/MLTC, W-PAFB, OH 45433 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



REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION <b>Unclassified</b>		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)	
5. NAME OF PERFORMING ORGANIZATION <b>General Electric Company Production Resources Consulting</b>		6a OFFICE SYMBOL <i>(if applicable)</i>	7a NAME OF MONITORING ORGANIZATION <b>AFVAL/MLTC</b>
6c ADDRESS (City, State and ZIP Code) <b>1 River Road Schenectady, NY 12345</b>		7b ADDRESS (City, State and ZIP Code) <b>WPAFB, OH 45433-6533</b>	
8a NAME OF FUNDING/SPONSORING ORGANIZATION <b>Materials Laboratory Air Force Systems Command, USAF</b>	8b OFFICE SYMBOL <i>(if applicable)</i> <b>AFVAL/MLTC</b>	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER <b>783618-80-C-5188</b>	
8c ADDRESS (City, State and ZIP Code) <b>-Wright-Patterson AFB, Ohio 45433</b>		10 SOURCE OF FUNDING NOS	
11 TITLE (Include Security Classification) <b>(See Reverse)</b>		PROGRAM ELEMENT NO. <b>780117</b>	PROJECT NO. <b>7500</b>
12 PERSONAL AUTHOR(S) <b>Singh, S., Althoff, J. L., and Apicella, M. L.</b>		TASK NO. <b>62</b>	WORK UNIT NO. <b>01</b>
13a TYPE OF REPORT <b>Final Technical Report</b>	13b TIME COVERED <b>22 Sept 1980 - 31 July 1985</b>	14. DATE OF REPORT (Yr., Mo., Day) <b>1985 November</b>	15. PAGE COUNT <b>411</b>
16. SUPPLEMENTARY NOTES <b>ICAN Project Priority 6201</b> The computer software contained herein are theoretical and/or references that in no way reflect Air Force-owned or -developed computer software.			
17. COSAT CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB GR	
1808	0805		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>This document is the product specification establishing the design implementation of the IISS Configuration Item Neutral Data Definition Language NDDL which is the primary tool used for maintaining the Common Data Model (CDM) data base. (6 pages VII) →</p>			
20 DISTRIBUTION AVAILABILITY OF ABSTRACT <b>UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/></b>		21 ABSTRACT SECURITY CLASSIFICATION <b>Unclassified</b>	
22a NAME OF RESPONSIBLE INDIVIDUAL <b>David L. Judson</b>		22b TELEPHONE NUMBER (Include Area Code) <b>512-255-6876</b>	22c OFFICE SYMBOL <b>AFVAL/MLTC</b>

11. Title

Integrated Information Support System (IISS)  
Vol V - Common Data Model Subsystem  
Part 6 - MDDL Processor Product Specification  
Section 3.10.8 CPFONE through INITRDL

A S D 86 1482  
17 Jul 1986



Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

PREFACE

This product specification covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows:

TASK 4.2

Subcontractors

Role

Boeing Military Aircraft  
Company (BMAC)

Reviewer

D. Appleton Company  
(DACOM)

Responsible for IDEF support,  
state-of-the-art literature  
search

General Dynamics/  
Ft. Worth

Responsible for factory view  
function and information  
models

Subcontractors

Role

Illinois Institute of  
Technology

Responsible for factory view  
function research (IITRI)  
and information models of  
small and medium-size business

North American Rockwell

Reviewer

Northrop Corporation

Responsible for factory view  
function and information  
models

Pritsker and Associates

Responsible for IDEF2 support

SofTech

Responsible for IDEFO support

TASKS 4.3 - 4.9 (TEST BED)

Subcontractors

Role

Boeing Military Aircraft  
Company (BMAC)

Responsible for consultation on  
applications of the technology  
and on IBM computer technology.

Computer Technology  
Associates (CTA)

Assisted in the areas of  
communications systems, system  
design and integration  
methodology, and design of the  
Network Transaction Manager.

Control Data Corporation  
(CDC)

Responsible for the Common Data  
Model (CDM) implementation and  
part of the CDM design (shared  
with DACOM).

D. Appleton Company  
(DACOM)

Responsible for the overall CDM  
Subsystem design integration and  
test plan, as well as part of  
the design of the CDM (shared  
with CDC). DACOM also  
developed the Integration  
Methodology and did the schema  
mappings for the Application  
Subsystems.

PS 620141100  
1 November 1985

Subcontractors

Role

Digital Equipment Corporation (DEC)

Consulting and support of the performance testing and on DEC software and computer systems operation.

McDonnell Douglas Automation Company (McAuto)

Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.

On-Line Software International (OSI)

Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.

Rath and Strong Systems Products (RSSP) (In 1985 became McCormack & Dodge)

Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.

SofTech, Inc.

Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.

Software Performance Engineering (SPE)

Responsible for directing the work on performance evaluation and analysis.

Structural Dynamics Research Corporation (SDRC)

Responsible for the User Interface and Virtual Terminal Interface Subsystems.

Other prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Boeing Military Aircraft Company (BMAC)	1701, 2201, 2202	Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC)

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Control Data Corporation (CDC)	1502, 1701	IISS enhancements to Common Data Model Processor (CDMP)
D. Appleton Company (DACOM)	1502	IISS enhancements to Integration Methodology
General Electric	1502	Operation of the Test Bed and communications equipment.
Hughes Aircraft Company (HAC)	1701	Test Bed enhancements
Structural Dynamics Research Corporation (SDRC)	1502, 1701, 1703	IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI)
Systran	1502	Test Bed enhancements. Operation of Test Bed.

TABLE OF CONTENTS

	<u>Page</u>
<u>BOUND SEGMENT 1</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 ..... 3-2
3.3.1	Inputs/Outputs ..... 3-3
3.4	Program Interrupts ..... 3-3
3.5	Timing and Sequencing Description ..... 3-3
3.6	Special Control Features ..... 3-3
3.7	Storage Allocation ..... 3-4
3.7.1	Database Definition ..... 3-4
3.7.1.1	File Description ..... 3-4
3.7.1.2	Table Description ..... 3-4
3.7.1.3	Item Description ..... 3-4
3.8	Object Code Creation ..... 3-4
3.9	Adaptation Data ..... 3-4
3.10	Detail Design Description ..... 3-4
3.10.1	Main Program List ..... 3-6
3.10.2	Module List ..... 3-8
3.10.3	External Routines List ..... 3-39
3.10.4	Include File List ..... 3-42
3.10.5	Where Include File Used List ..... 3-44
3.10.6	Where External Routine Used List .. 3-75
3.10.7	Main Program Parts List ..... 3-94
3.10.8	Module Documentation ..... 3-109
<u>BOUND SEGMENT 2</u>	
3.10.8	Module Documentation - CPFONE ..... 3-315
<u>BOUND SEGMENT 3</u>	
3.10.8	Module Documentation - INITSES .... 3-717

TABLE OF CONTENTS (Continued)

	<u>Page</u>
<b><u>BOUND SEGMENT 4</u></b>	
3.10.9	Include File Description ..... 3-1084
3.10.10	Hierarchy Chart ..... 3-1116
3.11	Program Listings Comments ..... 3-1518
<b>SECTION 4.0</b>	
	<b>QUALITY ASSURANCE PROVISIONS ..... 4-1</b>
4.1	Introduction and Definitions ..... 4-1
4.2	Computer Programming and Test Evaluation ..... 4-1



PS 620141100  
1 November 1985

FOREWORD

This is the second bound segment of four comprising Volume V, Part 6 of the Final Technical Report. It contains Sections 3.10.8 CPFONE through INITRDL.

NDDL COMMAND PROCESSOR Module Documentation

NAME: CPFONE  
PURPOSE: EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: CPF  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- CPFONE(LST, TYPE, STRING, COUNT)

COBOL -- CALL "CPFONE" USING

LIST-NO,

TYPE-ROW,

TOKEN,

LIST-COUNT.

FORTRAN -- CALL CPFONE(LST, TYP, TOKEN, COUNT)

INPUT:

LST -- AN INTEGER IDENTIFYING THE LIST TO BE  
ACCESSED.

OUTPUT:

**TYPE**  
- FOR IISS, THIS WILL ONLY HAVE A VALUE OF "ROW"

**STRING** - A BLANK PADDED CHARACTER STRING, ITS LENGTH  
DEPENDS ON THE LIST BEING ACCESSED. THIS  
WILL CONTAIN THE TOKEN PLACED ON THE  
LIST BY THE PARSER.

**COUNT**  
- AN INTEGER REPRESENTING THE NUMBER OF  
TOKENS THAT WERE PLACED ON THE LIST,  
NOTE THIS COULD BE 0.

**DESCRIPTION**

FOR THE GIVEN LIST NUMBER, SET UP THE OUTPUT COUNT  
VARIABLE, THE TYPE AND THE TOKEN IF FOUND. NOTE THE  
OUTPUT TOKEN IS BLANK PADDED.

**ARGUMENTS:**

-----  
LST = INT \*  
TYPE = CHAR \*  
STRING = CHAR \*  
COUNT = INT \*

**INCLUDE FILES:**

-----  
LISTS - PROVIDES THE DIMENSIONS OF THE NDDL LISTS  
NDDL - \*\*\*\* PURPOSE NOT FOUND BY STRIPPER \*\*\*\*

**ROUTINES CALLED:**

-----  
STRNCPY

**CALLED DIRECTLY BY:**

-----  
ADDATT - ASSOCIATES EXISTING ATT WITH ENTITY IN CREAT  
ENTITY COMMAND  
ADDKC - CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR  
CREATE/ALTER ENTITY  
ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD

KEYWORD\*(OPTIONAL)

ADDMIG	- PROCESS THE ADD MIGRATES...SET..CLAUSE
ALтали	- CONTROLS ALTER ALIAS PROCESSING (ALIAS TO PRIM OR VICE VERSA)
ALTCARD	- PROCESS CARDINALITY FO USER SPECIFIED RELATION
ALтDOM	- PROGRAM NAME
ALтENT	- CONTROL PROCESSING FOR ALTER ENTITY CLASS COMMAND.
ALтMOD	- CONTROLS PROCESSING FOR ALTER MODEL COMMANDS
ALтREL	- CONTROLS PROCESSING LOGIC FOR THE ALTER RELATION COMMAND
BLVWLST	- CREATE BUILD VIEW LISTS FOR THE CREATE VIEW COMMAND
CHKCARD	- OBTAINS THE USER-SPECIFIED CARDINALTY (IF ANY) FOR THE RELATION.
CHKMOD	- DETERMINES WHETHER CERTAIN RULES ARE FULFILLED,
CMBENT	- CONROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.
CMPMOD	- CONTROLS THE PROCESSING LOGIC TO COMPARE TWO MODELS.
COPATT	- CONTROLS THE PROCESSING LOGIC FOR COPYING AN ATTRIBUTE.
COPENT	- CONTROLS THE PROCESSING LOGIC FOR THE COPY ENTITY COMMAND.
CPYMOD	- CONTROLS THE PROCESSING LOGIC FOR THE COPY MODEL COMMAND.
CRTALI	- CREATES ALIASES FOR AN ENTITY OR ATTRIBUTE.
CRTATT	- CONTROLS THE PROCESSING LOGIC FOR CREATING AN ATTRIBUTE.
CRTDOM	- PROGRAM NAME CRTDOM
CRTENT	- CONTROL THE PROCESSING LOGIC FOR CREATING A NEW ENTITY CLASS.
CRTMOD	- CONTROL THE PROCESSING LOGIC IF CREATING A MODEL WITHIN THE SYSTE
CRTREL	- CONTROLS THE LOGIC FOR VALIDATING AND CREATING A NEW RELATION CLA
CRTVIEW	- CONTROLS THE PROCESSING LOGIC FOR THE CREATE VIEW COMMAND.
DEFAREA	- PROCESSES THE AREA INFORMATION IF THE DBMS IS IDS-II, IDMS AND VA
DEFODL	- PROCESSES THE DBMS TYPES: VAX-11, IDMS, IDS-II.
DEFDB	- CONTROLS THE PROCESSING LOGIC FOR DEFINING A DATABASE TO THE SYST
DEFFLD	- PROCESSES THE FIELD INFORMATION FOR THE DEFINE RECORD COMMAND.
DEFIMS	- PROCESSES THE DBMS TYPE: IMS.
DEFIMSS	- PROCESSES THE SEGMENT INFORMATION IF THE DBMS

IS IMS.

DEFKEY	- PROCESSES THE KEY INFORMATION FOR THE DEFINE RECORD COMMAND.
DEFORCL	- PROCESSES THE DBMS TYPE: ORACLE.
DEFSET	- CONTROLS THE PROCESSING LOGIC FOR THE DEFINE SET COMMAND.
DEFTOT	- PROCESSES THE DBMS TYPE: TOTAL.
DRPAC	- DELETE OWNED ATTRIBUTES ASSOCIATED WITH ENTITY
DRPALI	- DROP THE ALIAS FOR AN ENTITY OR ATTRIBUTE.
DRPATT	- CONTROLS THE DROPPING OF USER SPECIFIED ATTRIBUTE CLASSES FROM TH
DRPDB	- CONTROLS THE PROCESSING LOGIC FOR DELETING THE DATA BASE.
DRPDOM	- PROGRAM NAME
DRPENT	- CONTROL THE PROCESSING LOGIC FOR DELETING ENTITIES.
DRPFLD	- CONTROLS THE PROCESSING LOGIC FOR DROPPING A DATA FIELD.
DRPKC	- CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY CLASS".
DRPKW	- DROP A KEYWORD ASSOCIATION FROM EITHER AN ATTRIBUTE, ENTITY OR REL
DRPKWC	- OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP THEIR ASSOCIATIONS.
DRPMIG	- CONTROLS THE PROCESSING LOGIC FOR THE "DROP MIGRATES" CLAUSE.
DRPMOD	- CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.
DRPREL	- CONTROLS THE PROCESSING LOGIC FOR THE "DROP RELATION" COMMAND.
DRPSET	- CONTROLS THE PROCESSING LOGIC FOR DELETING A SET FROM THE DATABAS
DRPVIEW	- DROP THE VIEW.
EXCFLAG	- DETERMINE IF KEYWORDS, ALIASES, DESCR ARE TO BE EXCLUDED
GETDBST	- RETURN INFORMATION ABOUT THE CURRENT SESSIONS' DATABASE
GETDRT	- FETCH DATA BASE, RECORD TYPE INFO FROM PARSER LISTS
GETRDH	- RETURN WITH CURRENT SESSIONS' DATA BASE INFO
MKRNLST	- FETCH LIST OF RENAME PAIRS FOR MIGRATES..SET..CLAUSE
MRGMOD	- MERGE TWO IDEF MODELS INTO ONE
PROCDT	- PROGRAM NAME
RENAME	- UPDATE EXISTING OBJECT NAME WITH NEW OBJECT NAME
ADDMAP	- ADD A CS-IS MAPPING
ALTATT	- THE ALTER ATTRIBUTE COMMAND PROCESSOR ALTERS
ALTMAP	- ALTER MAP COMMAND PROCESSOR

PS 620141100  
1 November 1985

CHGDOM	- CHANGE THE DOMAIN FOR AN ALTERED ATTRIBUTE
CPYDES	- COMMAND PROCESSOR FOR COPY DESCRIPTION
CRTMAP	- CREATE MAP COMMAND PROCESSOR
DESCRB	- COMMAND PROCESSOR FOR THE NDDL DESCRIBE COMMAND
DRPMAP	- COMMAND PROCESSOR FOR THE DROP MAP COMMAND
HALT	- HALT WITH 'COMMIT' OR 'ROLLBACK'.
TXTTYP	- DETERMINE SOURCE OF DESCRIPTION TEXT

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100  
1 November 1985

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** CPFVAL  
**PURPOSE:** RETURN THE COUNTER OF LIST1 BASED ON ROW  
IN LIST2  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** CPF  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
**SYNOPSIS**

C -- CPFVAL(LIST1, LIST2, VALUE)

COBOL

-- CALL "CPFVAL" USING

LIST1,

LIST2,

VALUE.

FORTRAN

CALL CPFVAL(LIST1, LIST2, VAL)

**INPUT:**

LIST1 --

THE LIST CONTAINING THE COUNTER.

LIST2 --

THE LIST WHOSE CURRENT ROW WILL

BE USED AS THE INDEX INTO LIST1.

**OUTPUT:**

VALUE -- AN INTEGER REPRESENTING THE COUNTER THAT  
WAS KEPT IN LIST1.

DESCRIPTION

ACCESS THE FIRST ITEM FROM THE ROW OF LIST1  
CORRESPONDING TO THE CURRENT ROW OF LIST2.

ARGUMENTS:

-----  
LIST1 = INT \*  
LIST2 = INT \*  
VALUE = INT \*

INCLUDE FILES:

-----  
LISTS - PROVIDES THE DIMENSIONS OF THE NDDL LISTS  
NDDL - \*\*\*\* PURPOSE NOT FOUND BY STRIPPER \*\*\*\*

CALLED DIRECTLY BY:

-----  
ADDKC - CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR  
CREATE/ALTER ENTITY  
DEFKEY - PROCESSES THE KEY INFORMATION FOR THE DEFINE  
RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: CPYDES  
PURPOSE: COMMAND PROCESSOR FOR COPY DESCRIPTION  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: CPYDES  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- CPYDES(%CURRENT\_MODEL\_NO)  
COBOL -- CALL "CPYDES" USING  
CURRENT\_MODEL\_NO.  
FORTRAN -- CALL CPYDES(CURRENT\_MODEL\_NO)

INPUT:

INT \*CURRENT\_MODEL\_NO :

OUTPUT:

DESCRIPTION:

THIS IS A PARTIAL DESCRIPTION COPY. ONLY THE DESCRIPTION  
LINES OF THE  
IDENTIFIED DESCRIPTION TYPE OF THE GIVEN OBJECT WILL BE  
COPIED, RATHER  
THAN ALL DESCRIPTION TYPES.

ARGUMENTS:

-----

CURRENT\_MODEL\_NO - INT \*

INCLUDE FILES:

-----  
**LISTID** - PROVIDES LIST OF PARSED OBJECTS

**ROUTINES CALLED:**  
-----

**CPFONE** - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
**VEROBJ** - VERIFY THAT THE OBJECT EXISTS.  
**VERDSTX** - VERIFY THE EXISTENCE OF DESC. TEXT FOR A GIVEN  
OBJECT  
**VERDSTP** - VALIDATE A USER ENTERED DESCRIPTION TYPE  
**VERMOD** - verify the existence of a model and return the  
unique number.  
**SPRINTF**  
**UERROR** - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
**WRTDSC4** - SELECT A RECORD FROM DESC\_TEXT ENITY

**CALLED DIRECTLY BY:**  
-----

**BRANCHR** - PERFORMS MULTI-WAY CALL TO THE

**USED IN MAIN PROGRAM(S):**  
-----

**NDDL/MAIN** - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** CPYMOD  
**PURPOSE:** CONTROLS THE PROCESSING LOGIC FOR THE  
COPY MODEL COMMAND.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** CPYMOD  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----

CONTROLS THE PROCESSING LOGIC FOR THE COPY MODEL COMMAND.  
GENERATES ALL THE NDDL STATEMENTS NECESSARY TO POPULATE  
THE TARGET MODEL WITH THE ATTRIBUTES, ENTITIES, RELATIONS,  
KEY CLASSES, KEYWORDS, ALIASES AND DESCRIPTIONS CONTAINED  
IN THE FROM-MODEL.

ARE TO BE COPIED.

**ARGUMENTS:**

-----

MODEL-NO = DSPLY [S9(9)]  
MOD-NAME = DSPLY [X(30)]

**INCLUDE FILES:**

-----

LISTNOS - VALID LIST NUMBERS  
RCDEPKC - LIST OF KEYS MIGRATED VIA A RELATION  
CHKMODL - DETERMINE IF CURRENT MODEL EXISTS FOR A SESSION

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERMOD - verify the existence of a model and return the  
          unique number.  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
OPNFIL  
EXCFLAG - DETERMINE IF KEYWORDS, ALIASES, DESCR ARE TO BE  
          EXCLUDED  
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
          CHECKS/DELIMITER  
ALLATT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND  
          GENERATE  
ALLENT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND  
          GENERATE  
BLKCLST - SELECT AND STORE KEY CLASS INFO FOR A GIVEN  
          ENTITY  
TOPNODE - SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT  
          ENTITY  
ALLREL - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL  
          GENERATE  
CLSFIL - THIS ROUTINE CLOSSES AN OUTPUT FILE. THE FILE  
          WILL

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** CRTALI  
**PURPOSE:** CREATES ALIASES FOR AN ENTITY OR  
ATTRIBUTE.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** CRTALI  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----

CREATES ALIASES FOR AN ENTITY OR ATTRIBUTE

DEPENDING ON WHETHER AN ENTITY OR  
ATTRIBUTE HAS BE CREATED, VERIFY THAT  
IT EXISTS, AND THEN INSERT THE ALIAS.

**ARGUMENTS:**

-----

MODEL-NO = DSPY [S9(9)]

**INCLUDE FILES:**

-----

LISTNOS - VALID LIST NUMBERS

**ROUTINES CALLED:**

-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
LOWUPP - CONVERT A STRING TO UPPER CASE CHARACTERS  
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A  
MODEL  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A  
MODEL  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
INSECM - INSERT A RECORD INTO ENTIYT\_NAME

PS 620141100  
1 November 1985

**INSACNM - INSERT A RECORD INTO ATTRIBUTE\_NAME**

**CALLED DIRECTLY BY:**

-----  
**BRANCHR - PERFORMS MULTI-WAY CALL TO THE**

**USED IN MAIN PROGRAM(S):**

-----  
**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** CRTATT  
**PURPOSE:** CONTROLS THE PROCESSING LOGIC FOR  
CREATING AN ATTRIBUTE.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** CRTATT  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
CONTROLS THE PROCESSING LOGIC FOR CREATING  
AN ATTRIBUTE AND IT RELATED DATA IN A MODEL.  
ONLY A SINGLE ATTRIBUTE HAVING ONE DOMAIN IS  
IS CREATED. A KEYWORD LIST MAY OR MAY NOT BE  
SPECIFIED FOR THE ATTRIBUTE.

**ARGUMENTS:**

-----  
MODEL-NO = DSPLY [S9(9)]

**INCLUDE FILES:**

-----  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
LISTNOS - VALID LIST NUMBERS  
CHKMODL - DETERMINE IF CURRENT MODEL EXISTS FOR A SESSION

**ROUTINES CALLED:**

-----  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
CPHONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERDOM - VERIFY THE EXISTENCE OF A DOMAIN CLASS IN THE  
CDM  
BLDATT - CREATE ATT CLASS AND ATT NAME FOR A  
MODEL(CREATE/COPY ATT)  
INSACNM - INSERT A RECORD INTO ATTRIBUTE\_NAME

PS 620141100  
1 November 1985

ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD  
KEYWORD" (OPTIONAL)

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** CRTDOM  
**PURPOSE:** PROGRAM NAME CRTDOM  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** PROGRAM  
**SOURCE FILE:** CRTDOM  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

GIVEN A DOMAIN NAME VERIFY THAT THE DOMAIN DOES NOT ALREADY EXIST, AND INSERT THE DOMAIN AND A DATA TYPE CLAUSE

OBTAIN THE DOMAIN NAME, VERIFY THAT IT DID NOT PREVIOUSLY EXIST, FETCH AN UNIQUE DOMAIN NUMBER, AND INSERT. ALSO ADD THE DATE TYPE CLAUSE.

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

**ROUTINES CALLED:**  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERDOM - VERIFY THE EXISTENCE OF A DOMAIN CLASS IN THE CDM  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
GETNUM -  
INSDOM - INSERT A RECORD INTO DOMAIN\_CLASS  
PROCDT - PROGRAM NAME

**CALLED DIRECTLY BY:**  
-----

PS 620141100  
1 November 1985

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** CRTENT  
**PURPOSE:** CONTROL THE PROCESSING LOGIC FOR CREATING  
A NEW ENTITY CLAS  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** CRTENT  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
CONTROL THE PROCESSING LOGIC FOR CREATING A NEW  
ENTITY CLASS AND RELATED DATA IN A MODEL.

**ARGUMENTS:**

-----  
MODEL-NO = DSPLY [S9(9)]  
EC-NO = DSPLY [S9(9)]

**INCLUDE FILES:**

-----  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
LISTNOS - VALID LIST NUMBERS  
CHKMODL - DETERMINE IF CURRENT MODEL EXISTS FOR A SESSION

**ROUTINES CALLED:**

-----  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A  
MODEL  
GETNUM -  
INSEC - INSERT A RECORD INTO ENTITY\_CLASS  
INSECM - INSERT A RECORD INTO ENTIYT\_NAME

PS 620141100  
1 November 1985

ADDATT - ASSOCIATES EXISTING ATT WITH ENTITY IN GREAT  
ENTITY COMMAND  
ADDKC - CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR  
CREATE/ALTER ENTITY  
ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD  
KEYWORD"(OPTIONAL)

-----  
CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CRTMAP  
PURPOSE: CREATE MAP COMMAND PROCESSOR  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: CRTMAP  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- CRTMAP  
COBOL -- CALL CRTMAP.  
  
FORTRAN -- CALL CRTMAP

INPUT:

OUTPUT:

DESCRIPTION

CRTMAP IS THE COMMAND PROCESSOR FOR THE CREATE MAP  
COMMAND. IT  
VERIFIES THAT EITHER THE AUC NAME OR THE EC RC EC NAMES  
ARE VALID  
AND THEN CALLS ADDMAP TO ADD THE MAP.

**INCLUDE FILES:**

-----  
STDTYP - STANDARD TYPE DEFINITIONS  
LISTID - PROVIDES LIST OF PARSED OBJECTS

**ROUTINES CALLED:**

-----  
CPPONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A  
MODEL  
SPRINTF  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ADDMAP - ADD A CS-IS MAPPING  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
VERREL - VERIFY THE DEPENDENT AND INDEPENDENT ENTITIES  
IN THE RELATION  
VERAUC - VERIFY THE EXISTENCE OF AN ATTRIBUTE USE CLASS

**CALLED DIRECTLY BY:**

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** CRTMOD  
**PURPOSE:** CONTROL THE PROCESSING LOGIC IF CREATING  
A MODEL WITHIN THE  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** CRTMOD  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
THE CREATE MODEL COMMAND PROCESSOR WILL  
CONTROL THE PROCESSING LOGIC IF CREATING  
A MODEL WITHIN THE SYSTEM

**ARGUMENTS:**

-----  
CURRENT-MODEL-NO = DSPLY [S9(9)]  
CUR-MOD-NAME = DSPLY [X(30)]

**INCLUDE FILES:**

-----  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERMOD - verify the existence of a model and return the  
unique number.  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
GETNNUM -  
INSMOD - INSERT A RECORD INTO MODEL CLASS  
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----  
**BRANCHR - PERFORMS MULTI-WAY CALL TO THE**

**USED IN MAIN PROGRAM(S):**

-----  
**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**



**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** CRTREL  
**PURPOSE:** CONTROLS THE LOGIC FOR VALIDATING AND CREATING A NEW RELATI  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** CRTREL  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
CONTROLS THE LOGIC FOR VALIDATING AND CREATING A NEW RELATION CLASS WITHIN A MODEL. CREATE RELATION COMMAND CONSISTS OF A RELATION CLASS CLAUSE, AN OPTIONAL MIGRATES CLAUSE AND AN OPTIONAL KEYWORD CLAUSE.

**ARGUMENTS:**

-----  
MODEL-NO = DSPLY [S9(9)]

**INCLUDE FILES:**

-----  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
LISTNOS - VALID LIST NUMBERS  
CHKMODL - DETERMINE IF CURRENT MODEL EXISTS FOR A SESSION

**ROUTINES CALLED:**

-----  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
VERREL - VERIFY THE DEPENDENT AND INDEPENDENT ENTITIES  
IN THE RELATION  
GETNUM -

PS 620141100  
1 November 1985

CHKCARD - OBTAINS THE USER-SPECIFIED CARDINALTY (IF ANY)  
FOR THE RELATION.  
INSRC - INSERT A RECORD INTO RELATION CLASS  
ADDMIG - PROCESS THE ADD MIGRATES...SET...CLAUSE  
ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD  
KEYWORD"(OPTIONAL)

-----  
CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** CRTVIEW  
**PURPOSE:** CONTROLS THE PROCESSING LOGIC FOR THE  
CREATE VIEW COMMAND.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** PROGRAM  
**SOURCE FILE:** CRTVIEW  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR  
THE CREATE VIEW COMMAND

EXTRACT A VIEW-NAME, VERIFY THAT IT DOES  
EXIST; THEN FETCH A UNIQUE VIEW NO.  
FROM THE REUSABLE POOL AND  
INSERT THE VIEW -  
THEN PROCESS THE REMAINDER OF THE  
VIEW COMMAND.

**INCLUDE FILES:**

-----  
LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
VWFROM - LIST OF ENTITIES SPECIFIED IN A VIEW  
VWRETR - LIST OF ENTITIES AND TAGS SPECIFIED IN A VIEW  
VWDI - LIST OF DATA ITEMS IN A VIEW  
VWRC - LIST OF RELATION CLASSES INHERENT TO A VIEW

**ROUTINES CALLED:**

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERVIEW - verify the existence of a view in the system.  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
GETNUM -  
INSSEC - INSERT A RECORD INTO SEC

PS 620141100  
1 November 1985

BLVWLST - CREATE BUILD VIEW LISTS FOR THE CREATE VIEW  
COMMAND  
REVIEW - POPULATE VIEW DEFINITION AND ES-CS-MAPPING

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DEFAREA  
**PURPOSE:** PROCESSES THE AREA INFORMATION IF THE  
DBMS IS IDS-II, IDMS  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DEFAREA  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE PROCESSES THE AREA INFORMATION IF THE DBMS IS  
IDS-II, IDMS AND VAX-11.

**ARGUMENTS:**  
-----

DB-ID = DSPLY [S9(9)]  
RT-ID = DSPLY [X(30)]

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERAREA - VERIFY THE EXISTENCE OF AN AREA FOR A DATA BASE  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
INSDAA - INSERT A RECORD INTO THE DB\_AREA\_ASSIGNMENT  
ENTITY. IF  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
ERRPRO

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----  
**DEFREC** - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
RECORD FOR THE SYSTE

**USED IN MAIN PROGRAM(S):**

-----  
**NDDL/MAIN** - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DEFCODL  
**PURPOSE:** PROCESSES THE DBMS TYPES: VAX-11, IDMS,  
IDS-II.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DEFCODL  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

**THIS ROUTINE PROCESSES THE DBMS TYPES:**  
VAX-11  
IDMS  
IDS-II

First, extract the schema, subschema and area names. After verifying that the area name does not exist, insert into the cdm. Repeat till no more area names.

**ARGUMENTS:**  
-----

DB-ID = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST

PS 320141100  
1 November 1985

UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
INSSCH - INSERT A RECORD INTO THE SCHEMA\_NAMES ENTITY.  
IF  
VERAREA - VERIFY THE EXISTENCE OF AN AREA FOR A DATA BASE  
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
INSAREA - INSERT A RECORD INTO THE DATA\_BASE AREA ENTITY.  
IF  
ERRPRO

-----  
CALLED DIRECTLY BY:

DEFDB - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
DATABASE TO THE SYST

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DEFDB  
**PURPOSE:** CONTROLS THE PROCESSING LOGIC FOR  
DEFINING A DATABASE TO TH  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** PROGRAM  
**SOURCE FILE:** DEFDB  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
DATABASE TO THE SYSTEM.

extract the dbms type ,data base name and host id  
from the parser lists.  
After verifying that the data base does not  
already exist, insert it into the CDM. Now process the DBMS  
type depending on whether it is Total, IMS, ORACLE, VAX-11,  
IDMS or IDS-II.

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
LOWUPP - CONVERT A STRING TO UPPER CASE CHARACTERS  
VERDB - VERIFY THE EXISTENCE OF A DATA BASE IN THE CDM  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

PS 620141100  
1 November 1985

GETNNUM -  
INSDB - INSERT A RECORD INTO THE DATA\_BASE ENTITY. IF  
DEFORCL - PROCESSES THE DBMS TYPE: ORACLE.  
DEFTOT - PROCESSES THE DBMS TYPE: TOTAL.  
DEFIMS - PROCESSES THE DBMS TYPE: IMS.  
DEFCODL - PROCESSES THE DBMS TYPES: VAX-11, IDMS, IDS-II.  
CHGLOBL - CHANGES THE GLOBAL DB PARAMETERS  
ERRPRO

-----  
CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DEFFLD  
**PURPOSE:** PROCESSES THE FIELD INFORMATION FOR THE  
DEFINE RECORD COMMA  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DEFFLD  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE PROCESSES THE FIELD INFORMATION FOR THE DEFINE  
RECORD COMMAND.

**ARGUMENTS:**  
-----

DB-ID = DSPLY [S9(9)]  
RT-ID = DSPLY [X(30)]  
DBMS-NAME = DSPLY [X(30)]

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
CPFCOR - ACCESS A TOKEN IN A CORRESPONDING NAMED LIST  
LOWUPP - CONVERT A STRING TO UPPER CASE CHARACTERS  
INSDFLD - INSERT A RECORD INTO THE DATA FIELD ENTITY.  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
INSSDFL - INSERT A RECORD INTO THE SEGMENT\_DATA\_FIELD

PS 620141100  
1 November 1985

ENTITY. IF  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
ERRPRO

-----  
CALLED DIRECTLY BY:

DEFREC - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
RECORD FOR THE SYSTE

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DEFIMS  
PURPOSE: PROCESSES THE DBMS TYPE: IMS.  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DEFIMS  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

THIS ROUTINE PROCESSES THE DBMS TYPE: IMS

First extract the PSB sequence number, key feedback length, and PSB name from the parser lists. After verifying that the psb name does not previously exist, insert into PSB and PSB\_PCB.

ARGUMENTS:

-----

DB-ID = DSNLY [S9(9)]  
HOST-ID = DSNLY [X(30)]  
RETURN-STATUS = DSNLY [S9(9)]

INCLUDE FILES:

-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.

PS 620141100  
1 November 1985

VERPSB - VERIFIES THE EXISTANCE OF A PROGRAM STATUS  
BLOCK FOR AN IMS DATAB  
INSPSB - INSERT A RECORD INTO THE PSB ENTITY. IF  
INSPCB - INSERT A RECORD INTO THE PSB\_PCB ENTITY. IF  
ERRPRO

-----  
CALLED DIRECTLY BY:

DEFDB - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
DATABASE TO THE SYST

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100  
1 November 1985

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DEFIMSS  
**PURPOSE:** PROCESSES THE SEGMENT INFORMATION IF THE  
DBMS IS IMS.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DEFIMSS  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE PROCESSES THE SEGMENT INFORMATION IF THE DBMS  
IS IMS.

**ARGUMENTS:**  
-----

DB-ID = DSPLY [S9(9)]  
RT-ID = DSPLY [X(30)]

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
INSISS - INSERT A RECORD INTO THE IMS\_SEGMENT\_SIZE  
ENTITY. IF  
ERRPRO

PS 620141100  
1 November 1985

**-----  
CALLED DIRECTLY BY:**

**DEFREC - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
RECORD FOR THE SYSTE**

**-----  
USED IN MAIN PROGRAM(S):**

**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**



NDDL COMMAND PROCESSOR Module Documentation

NAME: DEFKEY  
PURPOSE: PROCESSES THE KEY INFORMATION FOR THE  
DEFINE RECORD COMMAND  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DEFKEY  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE PROCESSES THE KEY INFORMATION FOR THE DEFINE  
RECORD COMMAND.

ARGUMENTS:  
-----

DB-ID = DSPLY [S9(9)]  
RT-ID = DSPLY [X(30)]

INCLUDE FILES:  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
CPFHOR - ACCESS A TOKEN IN A CORRESPONDING NAMED LIST  
VERRK - verifies the existance of a key for a record.  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A IF  
INSRKEY - INSERT A RECORD INTO THE RECORD KEY ENTITY.  
CPFVAL - RETURN THE COUNTER OF LIST1 BASED ON ROW IN  
LIST2

PS 620141100  
1 November 1985

CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
VERRKM - verifies the existence of a data field for a  
record.  
INSRKM - INSERT A RECORD INTO THE RECORD\_KEY\_MEMBER  
ENTITY. IF  
ERRPRO

**CALLED DIRECTLY BY:**

-----  
DEFREC - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
RECORD FOR THE SYSTE

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DEFORCL  
PURPOSE: PROCESSES THE DBMS TYPE: ORACLE.  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DEFORCL  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE PROCESSES THE DBMS TYPE: ORACLE

Extract the password from the parser lists, then insert the oracle database id and password.

ARGUMENTS:  
-----

DB-ID = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
INSPWRD - INSERT A RECORD INTO THE DB\_PASSWORD. IF  
SUCCESSFUL,  
ERRPRO

PS 620141100  
1 November 1985

**-----**  
**CALLED DIRECTLY BY:**

**DEFDB - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
DATABASE TO THE SYST**

**-----**  
**USED IN MAIN PROGRAM(S):**

**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**

NDDL COMMAND PROCESSOR Module Documentation

NAME: DEFREC  
PURPOSE: CONTROLS THE PROCESSING LOGIC FOR  
DEFINING A RECORD FOR THE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: DEFREC  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
RECORD FOR THE SYSTEM.

AFTER EXTRACTING THE DB-ID AND RT-ID, VERIFY THAT THE RECORD  
TYPE DID NOT PREVIOUSLY EXIST. IF IT DID NOT, PROCESS THE  
AREA INFORMATION, IMS SEGMENT INFORMATION, FIELD INFORMATION  
AND KEY INFORMATION.

INCLUDE FILES:  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

GETRDH - RETURN WITH CURRENT SESSIONS' DATA BASE INFO  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERRT - VERIFIES THE EXISTANCE OF A RECORD TYPE.  
INSRTYP - INSERT A RECORD INTO THE RECORD TYPE ENTITY. IF  
DEFAREA - PROCESSES THE AREA INFORMATION IF THE DBMS IS  
IDS-II, IDMS AND VA  
DEFIMSS - PROCESSES THE SEGMENT INFORMATION IF THE DBMS  
IS IMS.

PS 620141100  
1 November 1985

DEFFLD - PROCESSES THE FIELD INFORMATION FOR THE DEFINE  
RECORD COMMAND.  
DEFKEY - PROCESSES THE KEY INFORMATION FOR THE DEFINE  
RECORD COMMAND.  
ERRPRO

-----  
CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DEFSET  
 PURPOSE: CONTROLS THE PROCESSING LOGIC FOR THE  
 DEFINE SET COMMAND.  
 LANGUAGE: VAX-11 COBOL  
 MODULE TYPE: PROGRAM  
 SOURCE FILE: DEFSET  
 SOURCE FILE TYPE: .COB  
 HOST:  
 SUBSYSTEM: CDM  
 SUBDIRECTORY:  
 DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
 CONTROLS THE PROCESSING LOGIC FOR THE  
 DEFINE SET COMMAND.  
 THE FOLLOWING DBMS ARE LEGAL FOR THIS PROCESS:  
 TOTAL, IMS, VAX-11, IDMS, IDS

DBMS	RULES	TOTAL	IMS	* VAX-11	IDMS	IDS
* CODASYL DBMS'S				*		
RULE	*					
SETS	NO	YES	YES	YES	YES	YES
EQUIRED/ PTIONAL OR MEMBERS	NO	NO	NO	YES	YES	YES
INKAGE/ ATA FIELD	NO	YES	NO	NO	NO	NO
INGLE EMBER LLOWED	NO	YES	YES	YES	YES	YES
ULTIPLE EMBERS LLOWED	NO	NO	NO	YES	YES	YES

INCLUDE FILES:

-----  
LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED:

-----  
GETDBST - RETURN INFORMATION ABOUT THE CURRENT SESSIONS'  
DATABASE  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERRSET - VERIFIES THE EXISTANCE OF A SET.  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERRT - VERIFIES THE EXISTANCE OF A RECORD TYPE.  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
INSSTM - INSERT A RECORD INTO THE SET\_TYPE\_MEMBER  
ENTITY. IF  
GETNNUM -  
INSRSET - INSERT A RECORD INTO THE RECORD\_SET ENTITY. IF  
VERDFLD - VERIFIES THE EXISTENCE OF A DATA FIELD FOR A  
GIVEN DATA BASE  
INSDSL - INSERT A RECORD INTO THE DF\_SET\_LINKAGE ENTITY.  
IF  
CPFCOR - ACCESS A TOKEN IN A CORRESPONDING NAMED LIST  
LOWUPP - CONVERT A STRING TO UPPER CASE CHARACTERS

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DEFTOT  
PURPOSE: PROCESSES THE DBMS TYPE: TOTAL.  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DEFTOT  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE PROCESSES THE DBMS TYPE: TOTAL

Extract the first file name from the parser lists. Then, after verifying that the file does not previously exist, insert the Total DBID and file nam into the CDM. Now, repeat process till no more file names.

ARGUMENTS:  
-----

DB-ID = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERAREA - VERIFY THE EXISTENCE OF AN AREA FOR A DATA BASE  
INSAREA - INSERT A RECORD INTO THE DATA\_BASE AREA ENTITY.

PS 620141100  
1 November 1985

IF  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
ERRPRO

CALLED DIRECTLY BY:

-----  
DEFDB - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
DATABASE TO THE SYST

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DEL1PDF  
PURPOSE: DELETE A RECORD FROM PROJECT\_DATA\_FIELD  
ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DEL1PDF  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- DEL1PDF(®DB\_ID,RT\_ID,DF\_ID,TAG\_NO,  
®STATUS)  
COBOL -- CALL "DELPDFT" USING  
DB-ID,  
RT-ID,  
DF-ID,  
TAG-NO,  
STATUS.  
FORTRAN -- CALL DELPDFT(DBID,RTID,DFID,TAGNO,STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;  
CHAR \*DF\_ID ;  
INT \*TAG\_NO ;  
CHAR \*RT\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
PROJECT\_DATA\_FIELD, IF SUCCESSFULLY DELETED THE RCORD ,

THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS :

-----  
DB\_ID = INT \*  
RT\_ID = CHAR \*  
DF\_ID = CHAR \*  
TAG\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
PDFDB - RETRIEVES PROJECT DATA FIELDS FOR A GIVEN DATA  
BASE  
PDFDF - RETRIVES ALL THE PROJ DATA FIELD OR DF BEING  
DROPPED  
PDFREC - RETRIEVES ALL PORJECT DATA FIELDS FOR A GIVEN  
RECORD

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELAC  
PURPOSE: DELETE A RECORD FROM ATTRIBUTE\_CLASS  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELAC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE AC\_NO = "INPUT PARAMETER 1" ;

SYNOPSIS

C -- DELAC(%AC\_NO,%STATUS)  
COBOL -- CALL "DELAC" USING  
AC-NO,  
STATUS.  
FORTRAN -- CALL DELAC(ACNO,STATUS)

INPUT:

INT \*AC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
ATTRIBUTE\_CLASS  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

PS 620141100  
1 November 1985

AC\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED  
ATTRIBUTE CLASSES FROM TH  
FNDACH - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELACAL  
PURPOSE: DELETE RECORD CLASS FROM ATTRIBUTE\_NAME  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELACAL  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
WHERE AC\_NAME =: "INPUT PARAMETER 1 " AND  
AC\_NO := "INPUT PARAMETER 2" ;

SYNOPSIS

C -- DELACAL(AC\_NAME, SAC\_NO, SSTATUS)  
COBOL -- CALL "DELACAL" USING  
AC-NAME,  
AC-NO,  
STATUS.  
FORTRAN -- CALL DELACAL(ACNAME, ACNO, STATUS)

INPUT:

CHAR \*AC\_NAME ;  
INT \*AC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
ATTRIBUTE\_NAME  
WHERE AC\_NAME EQUALS TO "INPUT PARAMETER 1" AND AC\_NO  
EQUALS  
TO "INPUT PARAMETER 2".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH

PS 620141100  
1 November 1985

WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
AC\_NAME =            CHAR \*  
AC\_NO =             INT \*  
STATUS =            INT \*

ROUTINES CALLED:

-----  
COSQL3       - ORACLE ROUTINE  
ERRRPT       - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC       - ORACLE ROUTINE  
COBINDN      - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
GETACAL      - VERIFY THE EXISTANCE OF AN ALIAS FOR AN  
              ATTRIBUTE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN    - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELACKW  
PURPOSE: DELETE A RECORD FROM AC\_KEYWORD  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELACKW  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE AC\_NO = "INPUT PARAMETER 1" ;

SYNOPSIS

C -- DELACKW(%AC\_NO,%STATUS)  
COBOL -- CALL "DELACKW" USING  
AC-NO,  
STATUS.  
FORTRAN -- CALL DELACKW(ACNO,STATUS)

INPUT:

INT \*AC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
AC\_KEYWORD  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

PS 620141100  
1 November 1985

AC\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
GOBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED  
ATTRIBUTE CLASSES FROM TH  
FNDACM - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELACNM  
PURPOSE: DELETE RECORD CLASS FROM ATTRIBUTE\_NAME  
WHERE AC\_NO  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELACNM  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

:= "INPUT PARAMETER" ;

SYNOPSIS

C -- DELACNM( &AC\_NO, &STATUS)  
COBOL -- CALL "DELACNM" USING  
AC-NO,  
STATUS.  
FORTRAN -- CALL DELACNM(ACNO,STATUS)

INPUT:

INT \*AC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
ATTRIBUTE\_NAME  
WHERE AC\_NO EQUAL TO THE INPUT PARAMETER(AC\_NO).  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

PS 620141100  
1 November 1985

-----  
AC\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED  
ATTRIBUTE CLASSES FROM TH  
FNDACH - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELASM  
PURPOSE: DELETE FROM ATTRIBUTE USE CLASS MAPPING  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELASM  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
WHERE TAG\_NO =: "INPUT PARAMETER 1 "  
SYNOPSIS  
C -- DELASM(TAG\_NO, , @STATUS)  
COBOL -- CALL "DELASM" USING  
TAG-NO,  
STATUS.  
FORTRAN -- CALL DELASM(TAGNO, STATUS)

INPUT:  
INT \*TAG\_NO ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:  
THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
AUC ST\_MAPPING  
WHERE TAG NO EQUALS TO "INPUT PARAMETER 1"  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
TAG\_NO = INT \*

PS 620141100  
1 November 1985

STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPMAP - COMMAND PROCESSOR FOR THE DROP MAP COMMAND

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELASM1  
PURPOSE: DELETE A RECORD FROM AUC\_ST\_MAPPING ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELASM1  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELASM1(%DB\_ID,%STATUS)  
COBOL -- CALL "DELASM1" USING  
DB-ID,  
STATUS.  
FORTRAN -- CALL DELASM1(DBID,STATUS)

INPUT:

INT \*DB\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
AUC\_ST\_MAPPING. IF SUCCESSFULLY DELETED THE RECORD ,  
THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID =  
STATUS =

INT \*  
INT \*

PS 620141100  
1 November 1985

ROUTINES CALLED:

-----  
COSQL3        - ORACLE ROUTINE  
ERRRPT       - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC       - ORACLE ROUTINE  
COBINDN      - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPDB        - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
                      DATA BASE.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN    - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELASM2  
PURPOSE: DELETE A RECORD FROM AUC\_ST\_MAPPING ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELASM2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELASM2(ØDB\_ID, ØSET\_ID, ØSTATUS)  
COBOL -- CALL "DELASM2" USING  
DB-ID,  
SET-ID,  
STATUS.  
FORTRAN -- CALL DELASM2(DBID, SETID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*SET\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
AUC ST MAPPING  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*

PS 620141100  
1 November 1985

SET\_ID - CHAR \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.  
DRPSET - CONTROLS THE PROCESSING LOGIC FOR DELETING A  
SET FROM THE DATABAS  
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET  
BEING DROPPED  
SELSTM - DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE  
BEING DROPPED

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELAUC  
PURPOSE: DELETE INHERITED\_ATT\_USE,  
KEY\_CLASS\_MEMBER,  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELAUC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

ATTRIBUTE\_USE\_CL WHERE THE TAG NUMBER  
EQUALS  
THE INPUT PARAMETER. ALSO DELETE  
AUC\_COPY\_XREF  
WHERE THE INPUT PARAMETER EQUALS  
TO AUC\_NO OR  
FROM\_AUC\_NO.

SYNOPSIS

C	--	DELAUC(&TAG_NO, &STATUS)	
COBOL	--	CALL "DELAUC"	USING TAG-NO STATUS.
FORTRAN	--	CALL DELAUC (TAGNO, STATUS)	

INPUT:

INT \*TAG\_NO

OUTPUT:

INT \*STATUS

DESCRIPTION

THIS ROUTINE CALLS ROUTINES WHICH DELETE THE

INHERITED ATTRIBUTE  
USE, KEY CLASS MEMBER, ATTRIBUTE USE CLASS AND THE  
ATTRIBUTE USE  
CLASS COPY CROSS REFERENCE ENTITY CLASSES WHERE THE  
TAG NUMBERS  
MATCH. THIS ROUTINE WILL ALSO DELETE TEXTUAL  
DESCRIPTION AND  
ADD THE REUSABLE TAG-NO BACK TO THE CDM FOR THE TAG-NO.

ARGUMENTS:

-----  
TAG\_NO = INT \*  
STATUS = INT \*

INCLUDE FILES:

-----  
STDTYP - STANDARD TYPE DEFINITIONS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED:

-----  
DELIAUC - DELETE A REORD FROM INHERITED\_ATT\_USE  
SPRINTF  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENITY  
DELAUCL - DELETE A RECORD FROM ATTRIBUTE\_USE\_CL  
DELAUCK - DELETE ATTRIBUTE USE CLASS KEY\_MEMBER GIVEN

CALLED DIRECTLY BY:

-----  
DELOAC - DELETE ALL OWNED AND INHERITED ATTRIBUTES  
DLMDAUC - DELETE ATTRIBUTE USE CLASSES ASSOCIATED WITH  
ENTITY  
FNDAUC - DELETE ALL ATTRIBUTE USE CLASSES FOR A GIVEN  
ENTITY CLASS

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELAUCK  
PURPOSE: DELETE ATTRIBUTE USE CLASS KEY MEMBER  
GIVEN  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELAUCK  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

THE TAG NUMBER

SYNOPSIS

C	--	DELAUCK(%TAG_NO, %STATUS)	
COBOL	--	CALL "DELAUCK"	USING TAG-NO STATUS.
FORTRAN	--	CALL DELAUCK(TAGNO, STATUS)	

INPUT:

INT \*TAG\_NO

OUTPUT:

INT \*STATUS

DESCRIPTION

THIS ROUTINE CALLS DELKCMT TO DELETE KEY CLASS MEMBERS  
GIVEN  
A TAG NUMBER FOLLOWED BY DELMIGK WHICH DELETES ALL  
ATTRIBUTE  
USE CLASSES FORMED FROM A MIGRATION OF THAT TAG NUMBER  
AND

PS 620141100  
1 November 1985

ANY MIGRATION OF THOSE ATTRIBUTE USE CLASSES.

ARGUMENTS:

-----  
TAG\_NO = INT \*  
STATUS = INT \*

INCLUDE FILES:

-----  
STDYTP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

-----  
DELKMT - DELETE A RECORD FROM KEY\_CLASS\_MEMBER  
SPRINTF  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DELMIGK - DELETE MIGRATING KEY CLASS

CALLED DIRECTLY BY:

-----  
DELAUC - DELETE INHERITED\_ATT\_USE, KEY\_CLASS\_MEMBER,

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELAUCL  
PURPOSE: DELETE A RECORD FROM ATTRIBUTE\_USE\_CL  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELAUCL  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS TAG\_NO := "INPUT PARAMETER 1" ;  
C -- DELAUCL(&TAG\_NO, &STATUS)  
COBOL -- CALL "DELAUCL" USING  
TAG-NO,  
STATUS.  
FORTRAN -- CALL DELAUCL(TAGNO,STATUS)

INPUT:  
INT \*TAG\_NO ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
ATTRIBUTE\_USE\_CL WHERE TAG NO EQUALS TO "INPUT  
PARAMETER 2".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

TAG\_NO = INT \*

PS 620141100  
1 November 1985

STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELAUC - DELETE INHERITED\_ATT\_USE, KEY\_CLASS\_MEMBER,  
DELMIGK - DELETE MIGRATING KEY CLASS  
DLMIGRC - DELETE MIGRATING KEY CLASS

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELCMPR  
PURPOSE: DELETE A RECORD FROM COMPLETE\_RELATION  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELCMPR  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
WHERE KC\_NO := "INPUT PARAMETER 2" ;  
SYNOPSIS  
C -- DELCMPR(&KC\_NO, &STATUS)  
COBOL -- CALL "DELCMPR" USING  
KC-NO,  
STATUS.  
FORTRAN -- CALL DELCMPR(KCNO,STATUS)

INPUT:  
INT \*KC\_NO ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:  
THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
COMPLETE\_RELATION WHERE KC\_NO EQUALS TO "INPUT  
PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 . OTHERWISE -1.

ARGUMENTS

-----  
KC NO - INT \*

PS 620141100  
1 November 1985

STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELMDKC - DELETE ALL KEY CLASSES AND INHERITED KEYS FOR  
AN ENTITY  
DRPKC - CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY  
CLASS".  
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELCPRC  
PURPOSE: DELETE A RECORD FROM COMPLETE\_RELATION  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELCPRC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE RC\_NO := "INPUT PARAMETER 1" ;

SYNOPSIS

C -- DELCPRC(ØRC\_NO, ØSTATUS)  
COBOL -- CALL "DELCPRC" USING  
RC-NO,  
STATUS.  
FORTRAN -- CALL DELCPRC(RCNO, STATUS)

INPUT:

INT \*RC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
COMPLETE\_RELATION WHERE RC NO EQUALS TO "INPUT  
PARAMETER 1".

IF SUCCESSFULLY DELETED THE RECORD . THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 . OTHERWISE -1.

ARGUMENTS

-----

RC NO - INT \*

STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPMIG - CONTROLS THE PROCESSING LOGIC FOR THE "DROP  
MIGRATES" CLAUSE.  
DRPREL - CONTROLS THE PROCESSING LOGIC FOR THE "DROP  
RELATION" COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDAA1  
PURPOSE: DELETE A RECORD FROM DB\_AREA\_ASSIGNMENT  
ENTITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDAA1  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELDAA1(%DB\_ID,%STATUS)  
COBOL -- CALL "DELDAA1" USING  
DB-ID,  
STATUS.  
FORTRAN -- CALL DELDAA1(DBID,STATUS)

INPUT:

INT \*DB\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DB\_AREA\_ASSIGNMENT. IF SUCCESSFULLY DELETED THE RECORD  
THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID - INT \*  
STATUS - INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDAA2  
PURPOSE: DELETE A RECORD FROM DB\_AREA\_ASSIGNMENT  
ENTITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDAA2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELDAA2(#DB\_ID, #RT\_ID, #STATUS)  
COBOL -- CALL "DELDAA2" USING  
DB-ID,  
RT-ID,  
STATUS.  
FORTRAN -- CALL DELDAA2(DBID, RTID, STATUS)

INPUT:

INT \*DB\_ID :  
CHAR \*RT\_ID :

OUTPUT:

INT \*STATUS :

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DB\_AREA\_ASSIGNMENT. IF SUCCESSFULLY DELETED THE RECORD  
, THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*

PS 620141100  
1 November 1985

RT\_ID -            CHAR \*  
STATUS -           INT \*

ROUTINES CALLED:

-----  
COSQL3        - ORACLE ROUTINE  
ERRRPT       - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC       - ORACLE ROUTINE  
COBINDM      - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPREC       - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
              RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN    - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDBA1  
PURPOSE: DELETE A RECORD FROM DATA\_BASE\_AREA ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDBA1  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- DELDBA1(%DB\_ID,%STATUS)  
  
COBOL -- CALL "DELDDBA1" USING  
DB-ID,  
STATUS.  
  
FORTRAN - CALL DELDBA1(DBID,STATUS)

INPUT:

INT \*DB\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DATA\_BASE\_AREA  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

DB\_ID = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDBDF  
PURPOSE: DELETE ALL DATA FIELD ASSOCIATIONS WITH  
THE DATABASE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DELDBDF  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
THIS ROUTINE WAS DESIGNED FOR USE WITH

DRPDB (DROP DATA BASE)

THIS ROUTINE VERIFIES THE EXISTANCE OF A DATA FIELD,  
SELECT A DATA FIELD ONE AT A TIME IN THE NDML LOOP,  
DELETES THE DATA FIELD, DELETES THE TEXTUAL DESCRIPTION,  
AND ADDS THE REUSABLE NUMBER BACK IN THE CDM.

IT HAS ONE INPUT PARAMETER (DB-ID) AND ONE  
OUTPUT PARAMETER FOUND-STATUS (0 GOOD AND -1  
AN ERROR HAS OCCURRED. ALL ERRORS ARE COVERED  
BY ERROR STATEMENTS IN THE ROUTINE.

ARGUMENTS:

-----  
DB-ID = DSPLY [S9(9)]  
FOUND-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

PS 620141100  
1 November 1985

ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
DELDFL3 - DELETE A RECORD FROM DATA\_FIELD ENITY  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENITY  
ERRPRO

CALLED DIRECTLY BY:

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

USED IN MAIN PROGRAM(S):

-----  
WDDL/MAIN - MAIN PROGRAM FOR THE WDDL COMMAND PROCESSOR

**NDML COMMAND PROCESSOR Module Documentation**

**NAME:** DELDBRT  
**PURPOSE:** DELETE ALL RECORD TYPE ASSOCIATIONS WITH  
THE DATABASE  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DELDBRT  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
THIS ROUTINE WAS DESIGNED FOR USE WITH

DRPDB (DROP DATA BASE)

THIS ROUTINE VERIFIES THE EXISTANCE OF A RECORD TYPE.  
SELECT A RECORD TYPE ONE AT A TIME IN THE NDML LOOP.  
DELETES THE RECORD TYPE, DELETES THE TEXTUAL DESCRIPTION,  
AND ADDS THE REUSABLE NUMBER BACK IN THE CDM.

IT HAS ONE INPUT PARAMETER (DB-ID) AND ONE  
OUTPUT PARAMETER FOUND-STATUS (0 GOOD AND -1  
AN ERROR HAS OCCURRED. ALL ERRORS ARE COVERED  
BY ERROR STATEMENTS IN THE ROUTINE.

**ARGUMENTS:**

-----  
DB-ID = DSNLY [S9(9)]  
FOUND-STATUS = DSNLY [S9(9)]

**INCLUDE FILES:**

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQUENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

PS 620141100  
1 November 1985

ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
DELRTY2 - DELETE A RECORD FROM RECORD\_TYPE ENTITY  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENTITY  
ERRPRO

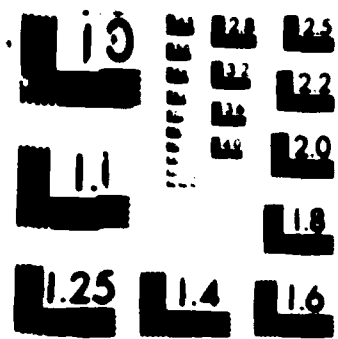
CALLED DIRECTLY BY:

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR







**DDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELDBS1  
**PURPOSE:** DELETE A RECORD FROM DATA\_BASE ENITY  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELDBS1  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----

**SYNOPSIS**

C -- DELDBS1(@DB\_ID,@STATUS)  
  
COBOL -- CALL "DELDBS1" USING  
          DB-ID,  
          STATUS.  
  
FORTRAN -- CALL DELDBS1(DBID,STATUS)

**INPUT:**

INT \*DB\_ID ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DATA\_BASE  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

-----

DB\_ID - INT \*  
STATUS - INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQL3       - ORACLE ROUTINE  
ERRRPT       - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC       - ORACLE ROUTINE  
COBINDN      - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPDB        - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
              DATA BASE.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN   - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDBST  
PURPOSE: DELETE ALL RECORD SETS' ASSOCIATIONS WITH  
THE DATABASE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DELDBST  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
THIS ROUTINE WAS DESIGNED FOR USE WITH

DRPDB (DROP DATA BASE)

THIS ROUTINE VERIFIES THE EXISTANCE OF A SET,  
SELECT A SET ONE AT A TIME IN THE NDML LOOP,  
DELETES THE SET, DELETES THE TEXTUAL DESCRIPTION,  
AND ADDS THE REUSABLE NUMBER BACK IN THE CDM.

IT HAS ONE INPUT PARAMETER (DB-ID) AND ONE  
OUTPUT PARAMETER FOUND-STATUS (0 GOOD AND -1  
AN ERROR HAS OCCURRED. ALL ERRORS ARE COVERED  
BY ERROR STATEMENTS IN THE ROUTINE.

ARGUMENTS:

-----  
DB-ID = DSPLY [S9(9)]  
FOUND-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

PS 620141100  
1 November 1985

**ERRPRO - PROCESS ERROR INCLUDE FILE**

**ROUTINES CALLED:**

-----  
**DELRST2 - DELETE A RECORD FROM RECORD\_SET ENTITY**  
**UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A**  
**ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO**  
**LINKED**  
**DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENITY**  
**ERRPRO**

**CALLED DIRECTLY BY:**

-----  
**DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE**  
**DATA BASE.**

**USED IN MAIN PROGRAM(S):**

-----  
**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELDFL1  
**PURPOSE:** DELETE A RECORD FROM DATA\_FIELD ENITY  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELDFL1  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
**SYNOPSIS**

C -- DELDFL1(%DB\_ID,%STATUS)  
  
COBOL -- CALL "DELDL1" USING  
DB-ID,  
STATUS.  
  
FORTRAN -- CALL DELDFL1(DBID,STATUS)

**INPUT:**  
INT \*DB\_ID ;

**OUTPUT:**  
INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM DATA\_FIELD. IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

-----  
DB\_ID = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDFL2  
PURPOSE: CONTROLS THE DELETING OF DATA FIELDS  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDFL2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: GDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

THAT BELONGED TO THE DELETED RECORD.  
THIS PROCESS WILL LOOP FOR EACH DATA FIELD  
AND DELETE THE TEXT DESCRIPTIONS, THE DATA  
FIELD AND CAPTURE THE DATA FIELD NUMBERS  
FOR REUSABLE NUMBER.

SYNOPSIS

C -- DELDFL2(%DB\_ID, %RT\_ID, %STATUS)  
COBOL -- CALL "DELDFL2" USING  
DB-ID,  
RT-ID,  
STATUS.  
FORTRAN -- CALL DELDFL2(DBID, RTID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

ARGUMENTS:

-----  
DB\_ID = INT \*  
RT\_ID = CHAR \*  
STATUS = INT \*

INCLUDE FILES:

-----  
UNIQUENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COFETCH - ORACLE ROUTINE  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENITY  
DRPDF - DELETE A RECORD FROM DATA\_FIELD ENITY  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE  
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDFL3  
PURPOSE: DELETE A RECORD FROM DATA\_FIELD ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDFL3  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- DELDFL3(&DB\_ID, &RT\_ID, &DF\_ID, &STATUS)

COBOL -- CALL "DELDL3" USING  
DB-ID,  
RT-ID,  
DF-ID,  
STATUS.

FORTRAN -- CALL DELDFL3(DBID, RTID, DFID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;  
CHAR \*DF\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DATA\_FIELD  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*  
RT\_ID = CHAR \*  
DF\_ID = CHAR \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELDBDF - DELETE ALL DATA FIELD ASSOCIATIONS WITH THE  
DATABASE  
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDIV  
PURPOSE: DELETE FROM DATA\_ITEM WHERE  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDIV  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
VIEW\_NO =: "INPUT PARAMETER 1 "  
SYNOPSIS  
C -- DELDIV(@VIEW\_NO, @STATUS)  
COBOL -- CALL "DELDIV" USING  
VIEW-NO,  
STATUS.  
FORTRAN -- CALL DELDIV(VIEWNO,STATUS)

INPUT:  
INT \*VIEW\_NO ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DATA-ITEM  
WHERE VIEW\_NO EQUALS "INPUT PARAMETER 1"  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
VIEW\_NO = INT \*

PS 620141100  
1 November 1985

STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPVIEW - DROP THE VIEW.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDOM  
PURPOSE: DELETE RECORD CLASS FROM DOMAIN\_CLASS  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDOM  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

WHERE DOMAIN\_NO =: "INPUT PARAMETER 1 "

SYNOPSIS

C -- DELDOM(@DOMAIN\_NO, @STATUS)  
COBOL -- CALL "DELDOM" USING  
          DOMAIN-NO,  
          STATUS.  
FORTRAN -- CALL DELDOM(DOMAINNO,STATUS)

INPUT:

INT \*DOMAIN\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DOMAIN\_CLASS  
WHERE DOMAIN\_NO EQUALS TO "INPUT PARAMETER 1"  
TO "INPUT PARAMETER 2".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:  
-----

PS 620141100  
1 November 1985

DOMAIN\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQLS - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDW - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPDOM - PROGRAM NAME

USED IN MAIN PROGRAM(S):

-----  
WDDL/MAIN - MAIN PROGRAM FOR THE WDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELDSL1  
**PURPOSE:** DELETE A RECORD FROM DF\_SET\_LINKAGE ENITY  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELDSL1  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
**SYNOPSIS**

C -- DELDSL1(@DB\_ID,@STATUS)  
  
COBOL -- CALL "DELDL1" USING  
                                  DB-ID,  
                                  STATUS.  
  
FORTRAN -- CALL DELDSL1(DBID,STATUS)

**INPUT:**

INT \*DB\_ID ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DF\_SET\_LINKAGE. IF SUCCESSFULLY DELETED THE RECORD ,  
                                  THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

-----  
DB\_ID - INT \*  
STATUS - INT \*

**ROUTINES CALLED:**

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDSL2  
PURPOSE: DELETE A RECORD FROM DF\_SET\_LINKAGE ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDSL2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELDSL2(@DB\_ID, @RT\_ID, @STATUS)  
COBOL -- CALL "DELDL2" USING  
DB-ID,  
RT-ID,  
STATUS.  
FORTRAN -- CALL DELDSL2(DBID, RTID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DF\_SET\_LINKAGE  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*

PS 620141100  
1 November 1985

RT\_ID - CHAR \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELDSL3  
**PURPOSE:** DELETE A RECORD FROM DF\_SET\_LINKAGE ENITY  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELDSL3  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
**SYNOPSIS**

C -- DELDSL3(%DB\_ID, %RT\_ID, %DF\_ID, %STATUS)  
  
COBOL -- CALL "DELDL3" USING  
DB-ID,  
RT-ID,  
DF-ID,  
STATUS.  
  
FORTRAN -- CALL DELDSL3(DBID, RTID, DFID, STATUS)

**INPUT:**

INT \*DB\_ID ;  
CHAR \*RT\_ID ;  
CHAR \*DF\_ID ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DF\_SET\_LINKAGE  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

PS 620141100  
1 November 1985

-----  
DB\_ID - INT \*  
RT\_ID - CHAR \*  
DF\_ID - CHAR \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDT  
PURPOSE: DELETE RECORD CLASS FROM  
USER\_DEF\_DATA\_TYPE  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDT  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: GDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
WHERE DATA\_TYPE\_NAME =: "INPUT PARAMETER 1 "

SYNOPSIS  
C -- DELDT(DATA\_TYPE\_NAME, @STATUS)  
COBOL -- CALL "DELDT" USING  
DATA-TYPE-NAME,  
STATUS.  
FORTRAN -- CALL DELDT(DATATYPENAME, STATUS)

INPUT:  
CHAR \*DATA\_TYPE\_NAME ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:  
THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
USER\_DEF\_DATA\_TYPE  
WHERE DATA\_TYPE\_NAME EQUALS TO "INPUT PARAMETER 1"  
TO "INPUT PARAMETER 2".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS :

PS 620141100  
1 November 1985

-----  
DATA\_TYPE\_NAME = CHAR \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPDT - DROP A DATA TYPE WITHIN A DOMAIN.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDTD  
PURPOSE: DELETE RECORD CLASS FROM  
USER\_DEF\_DATA-TYPE  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELDTD  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

WHERE DOMAIN\_NO =: "INPUT PARAMETER 1 "

SYNOPSIS

C -- DELDTD(&DOMAIN\_NO, &STATUS)  
COBOL -- CALL "DELDTD" USING  
DOMAIN-NO,  
STATUS.  
FORTRAN -- CALL DELDTD(DOMAINNO,STATUS)

INPUT:

INT \*DOMAIN\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
USER\_DEF\_DATA-TYPE  
WHERE DOMAIN\_NO EQUALS "INPUT PARAMETER 1 "  
TO "INPUT PARAMETER 2".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

PS 620141100  
1 November 1985

-----  
DOMAIN\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPDOM - PROGRAM NAME

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELDTNO  
PURPOSE: DELETE ALL DATA TYPES ASSOCIATED WITH A  
DOMAIN  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DELDTNO  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
-  
THIS ROUTINE WILL RETRIVE ALL USER\_DEF\_DATA\_TYPE  
NUMBERS (USDF\_DT\_NO) ASSOCIATED WITH A GIVEN DOMAIN  
NUMBER AND DELETE ALL TEXTUAL DESCRIPTIONS  
ASSOCIATED WITH THE DATA TYPE AND ADD THE  
USDF\_DT\_NO BACK INTO THE CDM.

ARGUMENTS:

-----  
DOM-NO = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

PS 620141100  
1 November 1985

ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENITY  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

-----  
CALLED DIRECTLY BY:

DRPDOM - PROGRAM NAME

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELEC  
PURPOSE: DELETE THE RECORD FROM ENTITY\_CLASS  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELEC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE EC\_NO =: "INPUT PARAMETER 1 "

SYNOPSIS

C -- DELEC(SEC\_NO, STATUS)  
COBOL -- CALL "DELEC" USING  
                                  EC-NO,  
                                  STATUS.  
FORTRAN -- CALL DELEC(ECNO, STATUS)

INPUT:

INT \*EC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
ENTITY\_CLASS  
WHERE EC\_NO EQUALS TO "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

EC\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING  
ENTITIES.  
FNDECM - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR  
A GIVEN MODEL

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELECAL  
PURPOSE: DELETE RECORD CLASS FROM ENTITY\_NAME  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELECAL  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE EC\_NAME =: "INPUT PARAMETER 1 " AND  
EC\_NO := "INPUT PARAMETER 2" ;

SYNOPSIS

C -- DELECAL(EC\_NAME, @EC\_NO, @STATUS)  
COBOL -- CALL "DELECAL" USING  
EC-NAME,  
EC-NO,  
STATUS.  
FORTRAN -- CALL DELECAL(ECNAME, ECNO, STATUS)

INPUT:

CHAR \*EC\_NAME ;  
INT \*EC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
ENTITY\_NAME  
WHERE EC\_NAME EQUALS TO "INPUT PARAMETER 1" AND EC\_NO  
EQUALS  
TO "INPUT PARAMETER 2".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH

PS 620141100  
1 November 1985

WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
EC\_NAME - CHAR \*  
EC\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
GETECAL - VERIFIES THE EXISTANCE OF AN ALIAS FOR AN ENTITY

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELECKW  
PURPOSE: DELETE A RECORD FROM EC\_KEYWORD  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELECKW  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

WHERE EC\_NO =: "INPUT PARAMETER 1" ;

SYNOPSIS

C -- DELECKW(%EC\_NO,%STATUS)  
COBOL -- CALL "DELECKW" USING  
          EC-NO,  
          STATUS.  
FORTRAN -- CALL DELECKW(ECNO,STATUS)

INPUT:

INT \*EC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
EC\_KEYWORD  
WHERE EC\_NO EQUALS TO "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:  
-----

PS 620141100  
1 November 1985

EC\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
GOEXEC - ORACLE ROUTINE  
GOBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING  
ENTITIES.  
FNDECM - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR  
A GIVEN MODEL

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELECNM  
PURPOSE: DELETE THE RECORD FROM ENTITY\_NAME  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELECNM  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE EC\_NO =: "INPUT PARAMETER 1 "

SYNOPSIS

C -- DELECNM(SEC\_NO, STATUS)  
COBOL -- CALL "DELECNM" USING  
EC-NO,  
STATUS.  
FORTRAN -- CALL DELECNM(ECNO, STATUS)

INPUT:

INT \*EC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
ENTITY\_NAME  
WHERE EC\_NO EQUALS TO "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

PS 620141100  
1 November 1985

EC NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING  
ENTITIES.  
FNDECM - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR  
A GIVEN MODEL

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELIASM  
PURPOSE: DELETE RECORD CLASS FROM AUC\_ST\_MAPPING  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELIASM  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE DB\_ID =: "INPUT PARAMETER 1 " AND  
SET\_ID := INPUT PARAMETER 2 AND  
TAG\_NO := INPUT PARAMETER 3 " ;

SYNOPSIS

C -- DELIASM(DB\_ID, @SET\_ID, @STATUS)  
COBOL -- CALL "DELIASM" USING  
DB-ID,  
SET-ID,  
TAG-NO,  
STATUS.  
FORTRAN -- CALL DELIASM(DBID,SETID,TAGNO,STATUS)

INPUT:

CHAR \*SET\_ID ;  
INT \*DB\_ID ;  
INT \*TAG\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
AUC\_ST\_MAPPING  
WHERE DB\_ID EQUALS TO "INPUT PARAMETER 1" AND SET\_ID  
EQUALS

PS 620141100  
1 November 1985

TO "INPUT PARAMETER 2" AND TAG NO EQUALS INPUT PARAM 3.  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*  
SET\_ID = CHAR \*  
TAG\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPSMAP - DROP A SINGLE MAPPING

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELIAUC  
PURPOSE: DELETE A REORD FROM INHERITED\_ATT\_USE  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELIAUC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

WHERE TAG\_NO = : 1 ;

SYNOPSIS

C -- DELIAUC(&TAG\_NO,&STATUS)  
COBOL -- CALL "DELIAUC" USING  
TAG-NO,  
STATUS.  
FORTRAN -- CALL DELIAUC(TAGNO,STATUS)

INPUT:

INT \*TAG\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
INHERITED\_ATT\_USE  
WHERE TAG\_NO EQUALS TO "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:  
-----

PS 620141100  
1 November 1985

TAG\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELAUC - DELETE INHERITED\_ATT\_USE, KEY\_CLASS\_MEMBER,  
DELMIGK - DELETE MIGRATING KEY CLASS  
DLMIGRC - DELETE MIGRATING KEY CLASS

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELIAUK  
PURPOSE: DELETE A RECORD FROM INHERITED\_ATT\_USE  
ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELIAUK  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELIAUK(ØKC\_NO, ØSTATUS)  
COBOL -- CALL "DELIAUK" USING  
KC-NO,  
STATUS.  
FORTRAN -- CALL DELIAUK(KCNO, STATUS)

INPUT:

INT \*KC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
INHERITED\_ATT\_USE. IF SUCCESSFULLY DELETED THE RECORD ,  
THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
KC\_NO = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

ROUTINES CALLED:

-----  
COSQL3       - ORACLE ROUTINE  
ERRRPT       - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC       - ORACLE ROUTINE  
COBINDN       - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELMDKC       - DELETE ALL KEY CLASSES AND INHERITED KEYS FOR  
                      AN ENTITY

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN     - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELIPDF  
PURPOSE: DELETE RECORD CLASS FROM  
PROJECT\_DATA\_FIELD  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELIPDF  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE AC\_NAME =: "INPUT PARAMETER 1 " AND  
AC\_NO := "INPUT PARAMETER 2" AND  
DF\_ID := "INPUT PARAMETER 3 " AND  
TAG\_NO := "INPUT PARAMETER 4" ;

SYNOPSIS

C -- DELIPDF(ØDB\_ID, RT\_ID, DF\_ID, TAG\_NO  
ØSTATUS)  
COBOL -- CALL "DELIPDF" USING  
DB-ID,  
RT-ID,  
DF-ID,  
TAG-NO,  
STATUS.  
FORTRAN -- CALL DELIPDF(DBID,RTID,DFID,TAGNO,STATUS)

INPUT:

CHAR \*DF\_ID ;  
CHAR \*RT\_ID ;  
INT \*DB\_ID ;  
INT \*TAG\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
PROJECT\_DATA\_FIELD  
WHERE DB\_ID EQUALS TO "INPUT PARAMETER 1" AND RT\_ID  
EQUALS  
TO "INPUT PARAMETER 2" AND DF\_ID EQUALS INPUT PARAMETER  
3  
AND TAG\_NO EQUALS INPUT PARAMETER 4 ".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID - INT \*  
RT\_ID - CHAR \*  
DF\_ID - CHAR \*  
TAG\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPSMAP - DROP A SINGLE MAPPING

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: DELIRCS  
PURPOSE: DELETE RECORD CLASS FROM RC\_BASED\_REC\_SET  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELIRCS  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE DB\_ID =: "INPUT PARAMETER 1 " AND  
SET\_ID := "INPUT PARAMETER 2" AND  
RT\_ID := "INPUT PARAMETER 3 " AND  
RC\_NO := "INPUT PARAMETER 4" ;

SYNOPSIS

C -- DELIRCS(@DB\_ID, SET\_ID, RT\_ID, @RC\_NO  
@STATUS)

COBOL -- CALL "DELIRCS" USING  
DB-ID,  
SET-ID,  
RT-ID,  
RC-NO,  
STATUS.

FOSETRAN -- CALL  
DELIRCS(DBID,SETID,RTID,RCNO,STATUS)

INPUT:

CHAR \*SET\_ID ;  
CHAR \*RT\_ID ;  
INT \*DB\_ID ;  
INT \*RC\_NO ;

OUTPUT

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RC\_BASED\_REC\_SET  
WHERE DB\_ID EQUALS TO "INPUT PARAMETER 1" AND RT\_ID  
EQUALS  
TO "INPUT PARAMETER 3" AND SET\_ID EQUALS INPUT  
PARAMETER 2  
AND RC\_NO EQUALS INPUT PARAMETER 4 ".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*  
SET\_ID = CHAR \*  
RT\_ID = CHAR \*  
RC\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPSMAP - DROP A SINGLE MAPPING

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELISS1  
PURPOSE: DELETE A RECORD FROM IMS\_SEGMENT\_SIZE  
ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELISS1  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- DELISS1(STATUS, DB\_ID)  
COBOL -- CALL "DELISS1" USING  
DB-ID,  
STATUS.  
FORTRAN -- CALL DELISS1(DBID, STATUS)

INPUT:

INT \*DB\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
IMS\_SEGMENT\_SIZE. IF SUCCESSFULLY DELETED THE RECORD ,  
THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

DB\_ID = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELISS2  
PURPOSE: DELETE A RECORD FROM IMS\_SEGMENT\_SIZE  
ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELISS2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELISS2(%DB\_ID, %RT\_ID, %STATUS)  
COBOL -- CALL "DELISS2" USING  
DB-ID,  
RT-ID,  
STATUS.  
FORTRAN -- CALL DELISS2(DBID, RTID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
IMS\_SEGMENT\_SIZE, IF SUCCESSFULLY DELETED THE RECORD ,  
THE STATUS AND  
RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*

PS 620141100  
1 November 1985

RT\_ID - CHAR \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELKC  
PURPOSE: DELETE A RECORD FROM KEY\_CLASS  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELKC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE KC\_NO =: 1 ;

SYNOPSIS

C -- DELKC(%KC\_NO,%STATUS)  
COBOL -- CALL "DELKC" USING  
KC-NO,  
STATUS.  
FORTRAN -- CALL DELKC(KCNO,STATUS)

INPUT:

INT \*KC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
KY CLASS  
WHERE KC\_NO = "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

PS 620141100  
1 November 1985

KC\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELMDKC - DELETE ALL KEY CLASSES AND INHERITED KEYS FOR  
AN ENTITY  
DRPKC - CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY  
CLASS".  
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELKCM  
PURPOSE: DELETE A RECORD FROM KEY\_CLASS\_MEMBER  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELKCM  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE KC\_NO =: 1 ;

SYNOPSIS

C -- DELKCM(%KC\_NO,%STATUS)  
COBOL -- CALL DELKCM" USING  
                                  KC-NO,  
                                  STATUS.  
FORTRAN -- CALL DELKCM(KCNO,STATUS)

INPUT:

INT \*KC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
KEY\_CLASS\_MEMBER WHERE KC\_NO = "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

KC\_NO = INT \*

PS 620141100  
1 November 1985

STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELMDC - DELETE ALL KEY CLASSES AND INHERITED KEYS FOR  
AN ENTITY  
DRPKC - CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY  
CLASS".

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELKCMT  
PURPOSE: DELETE A RECORD FROM KEY\_CLASS\_MEMBER  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELKCMT  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE TAG\_NO =: 1 ;

SYNOPSIS

C -- DELTAGMT(&TAG\_NO,&STATUS)  
COBOL -- CALL "DELKCMT" USING  
TAG-NO,  
STATUS.

FORTRAN -- CALL DELKCMT(TAGNO,STATUS)

INPUT:

INT \*TAG\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
KEY CLASS MEMBER WHERE TAG NO = "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

TAG\_NO = INT \*

PS 620141100  
1 November 1985

STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELAUCK - DELETE ATTRIBUTE USE CLASS KEY MEMBER GIVEN  
DELMIGK - DELETE MIGRATING KEY CLASS  
DLMIGRC - DELETE MIGRATING KEY CLASS

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELKW  
PURPOSE: DELETE A RECORD FROM AC\_KEYWORD  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELKW  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE KW\_NO =: 1 ;

SYNOPSIS

C -- DELKWAC(%KW\_NO,%STATUS)  
COBOL -- CALL "DELKW" USING  
KW\_NO,  
STATUS.  
FORTRAN -- CALL DELKW(KWNO,STATUS)

INPUT:

INT \*KW\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
AC\_KEYWORD  
WHERE KW\_NO = "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

PS 620141100  
1 November 1985

-----  
KW\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPKWC - OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP  
THEIR ASSOCIATIONS.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELKWAC  
**PURPOSE:** DELETE A RECORD FROM AC\_KEYWORD  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELKWAC  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----

WHERE KW\_NO =: 1 ;

**SYNOPSIS**

C -- DELKWAC(%KW\_NO,%STATUS)  
COBOL -- CALL "DELKWAC" USING  
KW-NO,  
STATUS.  
FORTRAN -- CALL DELKWAC(KWNO,STATUS)

**INPUT:**

INT \*KW\_NO ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
AC\_KEYWORD  
WHERE KW\_NO = "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

-----  
KW\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPKW - DROP A KEYWORD ASSOCIATION FROM EITHER AN  
ATTRIBUTE, ENTITY OR REL  
DRPKWC - OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP  
THEIR ASSOCIATIONS.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELKWEC  
PURPOSE: DELETE A RECORD FROM EC\_KEYWORD  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELKWEC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE KW\_NO =: 1 ;

SYNOPSIS

C -- DELKWEC(&KW\_NO,&STATUS)  
COBOL -- CALL "DELKWEC" USING  
KW-NO,  
STATUS.  
FORTRAN -- CALL DELKWEC(KWNO,STATUS)

INPUT:

INT \*KW\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
EC\_KEYWORD  
WHERE KW\_NO = "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

KW\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPKW - DROP A KEYWORD ASSOCIATION FROM EITHER AN  
ATTRIBUTE, ENTITY OR REL  
DRPKWC - OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP  
THEIR ASSOCIATIONS.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**MDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELKWRC  
**PURPOSE:** DELETE A RECORD FROM RC\_KEYWORD  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELKWRC  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----

WHERE KW\_NO =: 1 ;

**SYNOPSIS**

C -- DELKWRC(%KW\_NO,%STATUS)  
COBOL -- CALL "DELKWRC" USING  
KW-NO,  
STATUS.  
FORTRAN -- CALL DELKWRC(KWNO,STATUS)

**INPUT:**

INT \*KW\_NO ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RC\_KEYWORD  
WHERE KW\_NO = "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

-----

PS 620141100  
1 November 1985

KW\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPKW - DROP A KEYWORD ASSOCIATION FROM EITHER AN  
ATTRIBUTE, ENTITY OR REL  
DRPKWC - OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP  
THEIR ASSOCIATIONS.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELMDKC  
PURPOSE: DELETE ALL KEY CLASSES AND INHERITED KEYS  
FOR AN ENTITY  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DELMDKC  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

THIS ROUTINE DELETES THE KEY CLASSES, KEY CLASS MEMBERS,  
COMPLETE RELATIONS AND CROSS REFERENCES FOR A GIVEN ENTITY  
CLASS.

ARGUMENTS:

-----

EC-NO = DSPLY [S9(9)]  
ERROR-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----

DELKCM - DELETE A RECORD FROM KEY CLASS MEMBER  
DELCHPR - DELETE A RECORD FROM COMPLETE RELATION  
DELIAUK - DELETE A RECORD FROM INHERITED\_ATT\_USE ENTITY  
DELKC - DELETE A RECORD FROM KEY CLASS  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO

PS 620141100  
1 November 1985

LINKED

SELECNM - RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

-----  
CALLED DIRECTLY BY:

FNDECM - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR  
A GIVEN MODEL

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELMDRC  
PURPOSE: DELETE ALL RELATION CLASSES FOR AN ENTITY  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DELMDRC  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE DELETES THE RELATION CLASS FOR A GIVEN ENTITY.

ARGUMENTS:  
-----

EC-NO = DSPLY [S9(9)]  
ERROR-STATUS = DSPLY [S9(9)]

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

DELRC - DELETE A RECORD FROM RELATION CLASS  
DELRCWK - DELETE A RECORD FROM RC KEYWORD  
DELTEXT - DELETE A RECORD FROM DESC TEXT ENTITY  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
SELECNM - RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

PS 620141100  
1 November 1985

**CALLLED DIRECTLY BY:**

-----  
FNDECM - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR  
A GIVEN MODEL

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELMIGK  
PURPOSE: DELETE MIGRATING KEY CLASS  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELMIGK  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C	--	DELMIGK(*TAG_NO)	
COBOL	--	CALL "DELMIGK"	USING TAG-NO.
FORTRAN	--	CALL DELMIGK (TAGNO)	

INPUT:

INT \*TAG\_NO

OUTPUT:

DESCRIPTION

THIS ROUTINE USES AN ORACLE SELECT STATEMENT TO  
RETRIEVE ALL  
MIGRATING KEY CLASSES GIVEN A TAG NUMBER. ROUTINES ARE  
CALLED  
WHICH, FOR EACH NEW TAG NUMBER SELECTED, DELETE THE  
INHERITED  
ATTRIBUTE CLASS, ATTRIBUTE USE CLASS, COPY CROSS  
REFERENCE AND  
KEY CLASS MEMBER ENTITIES WHOSE TAG NUMBERS MATCH.

ARGUMENTS:

-----  
TAG\_NO = INT \*  
RTNSTA = INT \*

INCLUDE FILES:

-----  
STDYTP - STANDARD TYPE DEFINITIONS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COFETCH - ORACLE ROUTINE  
DELIAUC - DELETE A REORD FROM INHERITED\_ATT\_USE  
SPRINTF  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELKCMT - DELETE A RECORD FROM KEY\_CLASS\_MEMBER  
DELAUCL - DELETE A RECORD FROM ATTRIBUTE\_USE\_CL  
COEXEC - ORACLE ROUTINE  
CODFINN - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPMGKM - DELETE MIGRATED KEY CLASSES ASSOCIATED WITH A  
KEY CLASS  
DELAUCK - DELETE ATTRIBUTE USE CLASS KEY MEMBER GIVEN

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELMOD  
PURPOSE: DELETE A RECORD FROM MODEL\_CLASS ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELMOD  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELMOD(%MODEL\_NO, %STATUS)  
COBOL -- CALL "DELMOD" USING  
MODEL-NO,  
STATUS.  
FORTRAN -- CALL DELMOD(MODELNO, STATUS)

INPUT:  
INT \*MODEL\_NO ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
MODEL\_CLASS. IF SUCCESSFULLY DELETED THE RECORD , THE  
STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
MODEL\_NO = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQLS - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPHOD - CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**MDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELMTKC  
**PURPOSE:** DELETE EMPTY KEY CLASSES GIVEN THE MODEL  
NUMBER  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELMTKC  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
**SYNOPSIS**

C	--	DELMTCK(#MODEL_NO)	
COBOL	--	CALL "DELMTKC"	USING MODEL-NO.
FORTTRAN	--	CALL DELMTKC (MODELNO)	

**INPUT:**

INT \*MODEL\_NO

**OUTPUT:**

**DESCRIPTION**

THIS ROUTINE USES AN ORACLE SELECT STATEMENT TO OBTAIN ALL KEY CLASS NUMBERS WHICH HAVE NO MEMBERS. FOR EACH KEY CLASS FOUND, THE RECORD IS DELETED FROM THE COMPLETE RELATION CLASS. THE KEY CLASS ITSELF, AND THE KEY CLASS COPY CROSS REFERENCE. THIS ROUTINE WILL ALSO DELETE TEXTUAL DESCRIPTIONS AND ADD THE REUSABLE KEY CLASS NUMBER BACK TO THE CDM

ARGUMENTS :

-----  
MODEL\_NO - INT \*

INCLUDE FILES :

-----  
STDYTP - STANDARD TYPE DEFINITIONS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED :

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COFETCH - ORACLE ROUTINE  
DELCMPR - DELETE A RECORD FROM COMPLETE\_RELATION  
SPRINTF  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELTEXT - DELETE A RECORD FROM DESC TEXT ENTITY  
DELKC - DELETE A RECORD FROM KEY\_CLASS  
COEXEC - ORACLE ROUTINE  
CODFINN - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY :

-----  
ALTENT - CONTROL PROCESSING FOR ALTER ENTITY CLASS  
COMMAND.  
ALTREL - CONTROLS PROCESSING LOGIC FOR THE ALTER  
RELATION COMMAND  
DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED  
ATTRIBUTE CLASSES FROM TH  
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING  
ENTITIES.

USED IN MAIN PROGRAM(S) :

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELOAC  
PURPOSE: DELETE ALL OWNED AND INHERITED ATTRIBUTES  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DELOAC  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
- THIS ROUTINE SEARCHES FOR ALL ATTRIBUTE USE  
CLASSES WHERE THE TAG NUMBER MATCHES THE  
INPUT PARAMETER. EACH OCCURRENCE IS DELETED  
(DELAUC) AND THE OWNED ATTRIBUTE CLASS IS  
DELETED (DELOWAC).

ARGUMENTS:

-----  
AC-NO = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
DELAUC - DELETE INHERITED\_ATT\_USE, KEY\_CLASS\_MEMBER.  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DELOWAC - DELETE A RECORD FROM OWNED\_ATTRIBUTE  
ERRPRO

CALLED DIRECTLY BY:

-----  
DRPAC - DELETE OWNED ATTRIBUTES ASSOCIATED WITH ENTITY

PS 620141100  
1 November 1985

DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED  
ATTRIBUTE CLASSES FROM TH  
FNDOAC - DELETES ALL OWNED ATTRIBUTES FOR A GIVEN ENTITY  
CLASS

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELOACE  
**PURPOSE:** DELETE A RECORD FROM OWNED\_ATTRIBUTE ENTITY  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELOACE  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
**SYNOPSIS**

C -- DELOACE( #EC\_NO , #STATUS )  
  
COBOL -- CALL "DELOACE" USING  
                                  EC-NO.  
                                  STATUS.  
  
FORTRAN -- CALL DELOACE( ECNO , STATUS )

**INPUT:**

INT \*EC\_NO ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM OWNED\_ATTRIBUTE. IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

-----  
EC\_NO - INT \*  
STATUS - INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQL3       - ORACLE ROUTINE  
ERRRPT       - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC       - ORACLE ROUTINE  
COBINDM      - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
FNDECM       - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR  
              A GIVEN MODEL

**USED IN MAIN PROGRAM(S):**

-----  
MDDL/MAIN    - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELOWAC  
PURPOSE: DELETE A RECORD FROM OWNED\_ATTRIBUTE  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELOWAC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE AC\_NO =: "INPUT PARAMETER 1" ;

SYNOPSIS

C -- DELOWAC(%AC\_NO,%STATUS)  
COBOL -- CALL "DELOWAC" USING  
AC-NO.  
STATUS.  
FORTRAN -- CALL DELOWAC(ACNO,STATUS)

INPUT:

INT \*AC\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
OWNED\_ATTRIBUTE  
WHERE AC\_NO =: "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

PS 620141100  
1 November 1985

-----  
AC\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELOAC - DELETE ALL OWNED AND INHERITED ATTRIBUTES

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELPCB  
PURPOSE: DELETE A RECORD FROM PSB\_PCB ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELPCB  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- DELPCB(%DB\_ID,%STATUS)  
COBOL -- CALL "DELPCE" USING  
DB-ID,  
STATUS.  
FORTRAN -- CALL DELPCB(DBID,STATUS)

INPUT:

INT \*DB\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM PSB\_PCB. IF SUCCESSFULLY DELETED THE RECORD, THE STATUS AND RETURN VALUE BOTH WILL BE 0, OTHERWISE -1.

ARGUMENTS:

-----

DB ID - INT \*  
STATUS - INT \*

ROUTINES CALLED

PS 620141100  
1 November 1985

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

-----  
CALLED DIRECTLY BY:

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELPDFT  
PURPOSE: DELETE A RECORD FROM PROJECT DATA FIELD  
ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELPDFT  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELPDFT(@TAG NO, @STATUS)  
COBOL -- CALL "DELPDFT" USING  
TAG-NO  
STATUS  
FORTRAN -- CALL DELPDFT(TAGNO STATUS)

INPUT

INT \*TAG NO  
CHAR \*RT ID

OUTPUT

INT \*STATUS

DESCRIPTION

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
PROJECT DATA FIELD IF SUCCESSFULLY DELETED THE RECORD  
THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 OTHERWISE :

ARGUMENTS

TAG NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
PDFDB - RETRIEVES PROJECT DATA FIELDS FOR A GIVEN DATA  
BASE  
PDFDF - RETRIVES ALL THE PROJ DATA FIELD OR DF BEING  
DROPPED  
PDFREC - RETRIEVES ALL PORJECT DATA FIELDS FOR A GIVEN  
RECORD  
DRPHAP - COMMAND PROCESSOR FOR THE DROP MAP COMMAND

USED IN MAIN PROGRAM(S):

-----  
NDDL MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELPDI  
PURPOSE: DELETE RECORD CLASS FROM PROJECT\_DATA\_ITEM  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELPDI  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
WHERE VIEW\_NO :=INPUT PARAMETER 1" ;  
SYNOPSIS  
C -- DELPDI( @VIEW\_NO, @STATUS)  
COBOL -- CALL "DELPDI" USING  
VIEW-NO,  
STATUS.  
FORTRAN -- CALL DELPDI(VIEWNO,STATUS)

INPUT  
INT \*VIEW\_NO :

OUTPUT  
INT \*STATUS :

DESCRIPTION

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
PROJECT\_DATA\_ITEM  
WHERE VIEW\_NO EQUALS "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 . OTHERWISE -1.

ARGUMENTS

VIEW\_NO INT \*

PS 620141100  
1 November 1985

STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPVIEW - DROP THE VIEW.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

DDDL COMMAND PROCESSOR Module Documentation

NAME DELRBR1  
PURPOSE DELETE A RECORD FROM RC\_BASED\_REC\_SET  
ENTITY  
LANGUAGE C  
MODULE TYPE FUNCTION  
FUNCTION TYPE INT ()  
SOURCE FILE DELRBR1  
SOURCE FILE TYPE C  
HOST  
SUBSYSTEM CDM  
SUBDIRECTORY  
DOCUMENTATION GROUP PS41100

DESCRIPTION

SYNOPSIS

C DELRBR1(DB ID,STATUS)

COBOL CALL "DELRBR1" USING  
DB-ID,  
STATUS

FORTRAN CALL DELRBR1(DBID,STATUS)

INPUT

INT 'DB ID .

OUTPUT

INT 'STATUS .

DESCRIPTION

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RC BASED REC\_SET. IF SUCCESSFULLY DELETED THE RECORD,  
THE STATUS

AND RETURN VALUE BOTH WILL BE 0, OTHERWISE -1.

ARGUMENTS:

DB ID - INT '  
STATUS - INT '

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQLS - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**MDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELRBR2  
**PURPOSE:** DELETE A RECORD FROM RC\_BASED\_REC\_SET  
ENITY  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELRBR2  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
**SYNOPSIS**

C -- DELRBR2(#DB\_ID, #SET\_ID, #STATUS)  
COBOL -- CALL "DELRBR2" USING  
DB-ID,  
SET-ID,  
STATUS.  
FORTRAN -- CALL DELRBR2(DBID, SETID, STATUS)

**INPUT:**

INT \*DB\_ID ;  
CHAR \*SET\_ID ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RC BASED REC SET  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

-----

PS 620141100  
1 November 1985

DB\_ID - INT \*  
SET\_ID - CHAR \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQLS - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPSET - CONTROLS THE PROCESSING LOGIC FOR DELETING A  
SET FROM THE DATABASES  
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET  
BEING DROPPED  
SELSTM - DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE  
BEING DROPPED

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



**NDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELRBR3  
**PURPOSE:** DELETE A RECORD FROM RC\_BASED\_REC\_SET  
ENITY  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELRBR3  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
**SYNOPSIS**

C -- DELRBR3(#DB\_ID, #SET\_ID, #RT\_ID, #STATUS)  
  
COBOL -- CALL "DELRBR3" USING  
DB-ID,  
SET-ID,  
DF-ID,  
STATUS.  
  
FORTRAN -- CALL DELRBR3(DBID, SETID, DFID, STATUS)

**INPUT:**

INT \*DB\_ID ;  
CHAR \*SET\_ID ;  
CHAR \*RT\_ID ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RC BASED\_REC\_SET  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

-----  
DB\_ID - INT \*  
SET\_ID - CHAR \*  
RT\_ID - CHAR \*  
STATUS - INT \*

**ROUTINES CALLED:**

-----  
COSQLS - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.

**USED IN MAIN PROGRAM(S):**

-----  
MDDL/MAIN - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELRC  
**PURPOSE:** DELETE A RECORD FROM RELATION\_CLASS  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELRC  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----

WHERE RC\_NO =: "INPUT PARAMETER 1" ;

**SYNOPSIS**

C -- DELRC( RC\_NO , STATUS )  
COBOL -- CALL "DELRC" USING  
RC\_NO ,  
STATUS .  
FORTRAN -- CALL DELRC( RCNO , STATUS )

**INPUT:**

INT \*RC\_NO ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RELATION CLASS  
WHERE RC NO = "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

PS 620141100  
1 November 1985

-----  
RC\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELMDC - DELETE ALL RELATION CLASSES FOR AN ENTITY  
DRPRCE - DROP A RELATION CLASS FOR AN ENTITY BEING  
DROPPED  
DRPREL - CONTROLS THE PROCESSING LOGIC FOR THE "DROP  
RELATION" COMMAND.

USED IN MAIN PROGRAM(S):

-----  
MDDL/MAIN - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DELRCKW  
**PURPOSE:** DELETE A RECORD FROM RC\_KEYWORD  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DELRCKW  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----

WHERE RC\_NO =: 1 ;

**SYNOPSIS**

C -- DELRCKW(#RC\_NO,#STATUS)  
COBOL -- CALL "DELRCKW" USING  
RC-NO,  
STATUS.  
FORTRAN -- CALL DELRCKW(RCNO,STATUS)

**INPUT:**

INT \*RC\_NO ;

**OUTPUT:**

INT \*STATUS ;

**DESCRIPTION:**

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RC\_KEYWORD  
WHERE RC\_NO = "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

**ARGUMENTS:**

-----

PS 620141100  
1 November 1985

RC\_NO - INT \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQLS - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELMDC - DELETE ALL RELATION CLASSES FOR AN ENTITY  
DRPRCE - DROP A RELATION CLASS FOR AN ENTITY BEING  
          DROPPED  
DRPREL - CONTROLS THE PROCESSING LOGIC FOR THE "DROP  
          RELATION" COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

DDDL COMMAND PROCESSOR Module Documentation

NAME	DELRCST
PURPOSE	DELETE RECORD CLASS FROM RC BASED REC SET
LANGUAGE	C
MODULE TYPE	FUNCTION
FUNCTION TYPE	INT ( )
SOURCE FILE	DELRCST
SOURCE FILE TYPE	C
HOST	
SUBSYSTEM	CDM
SUBDIRECTORY	
DOCUMENTATION GROUP	PS41100

DESCRIPTION

-----  
WHERE RC\_NO = "INPUT PARAMETER 1" ;

SYNOPSIS	
C	-- DELRCST( #AC_NO, #STATUS )
COBOL	-- CALL "DELRCST" USING RC-NO, STATUS.
FORTTRAN	-- CALL DELRCST(RCNO,STATUS)

INPUT  
INT \*RC\_NO ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:  
THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RC BASED REC SET  
WHERE RC\_NO EQUALS "INPUT PARAMETER 1" .  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
RC\_NO = INT \*

PS 620141100  
1 November 1985

STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

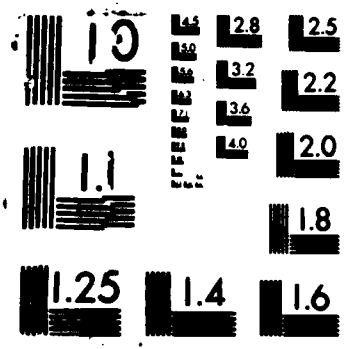
-----  
DRPMAP - COMMAND PROCESSOR FOR THE DROP MAP COMMAND

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR







NDDL COMMAND PROCESSOR Module Documentation

NAME: DELREUS  
PURPOSE:  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELREUS  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELREUS (POOL\_NO, NEXT\_NO, STATUS)  
COBOL -- CALL "DELREUS" USING  
POOL-NO,  
NEXT\_NO,  
STATUS.  
FORTRAN -- CALL DELREUS (POOLNO, NEXT\_NO, STATUS)

INPUT:

INT \*POOL\_NO ;

OUTPUT:

INT \*NEXT\_NO ;  
INT \*STATUS ;

DESCRIPTION

THIS ROUTINE DELETES A UNIQUE NUMBER OF A POOL NUMBER  
FROM  
REUSABLE\_NUMBER RELATION

ARGUMENTS:

-----  
POOL\_NO = INT \*  
NEXT\_NO = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

ROUTINES CALLED:

-----  
COSQL3        - ORACLE ROUTINE  
ERRRPT       - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC       - ORACLE ROUTINE  
COBINDN      - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
NRGET        -

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN    - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELRKM1  
PURPOSE: DELETE A RECORD FROM RECORD\_KEY\_MEMBER  
ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELRKM1  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- DELRKM1(%DB\_ID,%STATUS)  
COBOL -- CALL "DELRKM1" USING  
DB-ID,  
STATUS.  
FORTRAN -- CALL DELRKM1(DBID,STATUS)

INPUT:

INT \*DB\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RECORD\_KEY\_MEMBER. IF SUCCESSFULLY DELETED THE RECORD ,  
THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

DB\_ID = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELRKM2  
PURPOSE: DELETE A RECORD FROM RECORD\_KEY\_MEMBER  
ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELRKM2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELRKM2(%DB\_ID, %RT\_ID, %STATUS)  
COBOL -- CALL "DELRKM2" USING  
DB-ID,  
RT-ID,  
STATUS.  
FORTRAN -- CALL DELRKM2(DBID, RTID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RECORD KEY MEMBER  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

PS 620141100  
1 November 1985

DB\_ID = INT \*  
RT\_ID = CHAR \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELRKM3  
PURPOSE: DELETE A RECORD FROM RECORD\_KEY\_MEMBER  
ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELRKM3  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELRKM3(%DB\_ID, %RT\_ID, %DF\_ID, %STATUS)  
  
COBOL -- CALL "DELRKM3" USING  
DB-ID,  
RT-ID,  
DF-ID,  
STATUS.  
  
FORTRAN -- CALL DELRKM3(DBID, RTID, DFID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;  
CHAR \*DF\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RECORD KEY MEMBER  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

PS 620141100  
1 November 1985

**ARGUMENTS:**

-----  
DB\_ID = INT \*  
RT\_ID = CHAR \*  
DF\_ID = CHAR \*  
STATUS = INT \*

**ROUTINES CALLED:**

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELRKY1  
PURPOSE: DELETE A RECORD FROM RECORD\_KEY ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELRKY1  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELRKY1(%DB\_ID,%STATUS)  
  
COBOL -- CALL "DELRKY1" USING  
                                  DB-ID,  
                                  STATUS.  
  
FORTRAN -- CALL DELRKY1(DBID,STATUS)

INPUT:

INT \*DB\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RECORD\_KEY. IF SUCCESSFULLY DELETED THE RECORD , THE  
STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
GOEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELRKY2  
PURPOSE: DELETE A RECORD FROM RECORD\_KEY ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELRKY2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- DELRKY2(%DB\_ID, %RT\_ID, %STATUS)  
COBOL -- CALL "DELRKY2" USING  
DB-ID,  
RT-ID,  
STATUS.  
FORTRAN -- CALL DELRKY2(DBID, RTID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RECORD\_KEY  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

DB\_ID = INT \*

PS 620141100  
1 November 1985

RT\_ID - CHAR \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.  
DRPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELRST2  
PURPOSE: DELETE A RECORD FROM RECORD\_SET ENTITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELRST2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELRST2(&DB\_ID, SET\_ID, &STATUS)  
  
COBOL -- CALL "DELRST2" USING  
DB-ID,  
SET-ID,  
STATUS.  
  
FRTRAN -- CALL DELRST2(DBID, SETID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*SET\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RECORD\_SET  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*

SET\_ID = CHAR \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELDBST - DELETE ALL RECORD SETS' ASSOCIATIONS WITH THE  
DATABASE  
DRPSET - CONTROLS THE PROCESSING LOGIC FOR DELETING A  
SET FROM THE DATABAS  
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET  
BEING DROPPED  
SELSTM - DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE  
BEING DROPPED

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELRST3  
PURPOSE: DELETE A RECORD FROM RECORD\_SET ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELRST3  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

```
C          --  DELRST3(@DB_ID, @SET_ID,  
              @RT_ID_OF_OWNER, @STATUS)
```

```
COBOL     --  CALL "DELRST3" USING  
              DB-ID,  
              SET-ID,  
              DF-ID,  
              STATUS.
```

```
FORTRAN   --  CALL DELRST3(DBID, SETID, DFID, STATUS)
```

INPUT:

```
INT  *DB_ID ;  
CHAR *SET_ID ;  
CHAR *RT_ID_OF_OWNER ;
```

OUTPUT:

```
INT *STATUS ;
```

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RECORD\_SET  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

PS 620141100  
1 November 1985

**ARGUMENTS:**

-----

DB\_ID = INT \*  
SET\_ID = CHAR \*  
RT\_ID\_OF\_OWNER = CHAR \*  
STATUS = INT \*

**ROUTINES CALLED:**

-----

COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----

DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.

**USED IN MAIN PROGRAM(S):**

-----

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELRTY2  
PURPOSE: DELETE A RECORD FROM RECORD\_TYPE ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELRTY2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS  
C -- DELRTY2(%DB\_ID, %RT\_ID, %STATUS)  
COBOL -- CALL "DELRTY" USING  
DB-ID,  
RT-ID,  
STATUS.  
FORTRAN -- CALL DELRTY(DBID, RTID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
RECORD\_TYPE  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*

RT\_ID =            CHAR \*  
STATUS =            INT \*

ROUTINES CALLED:

-----  
COSQ'3            - ORACLE ROUTINE  
ERRRPT           - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC           - ORACLE ROUTINE  
COBINDN           - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELDERT           - DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE  
                  DATABASE  
DRPREC           - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
                  RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN        - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELSDF1  
PURPOSE: DELETE A RECORD FROM SEGMENT\_DATA\_FIELD  
ENTITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELSDF1  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELSDF1(%DB\_ID,%STATUS)  
COBOL -- CALL "DELSDF1" USING  
DB-ID,  
STATUS.  
FORTRAN -- CALL DELSDF1(DBID,STATUS)

INPUT:  
INT \*DB\_ID ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
SEGMENT\_DATA\_FIELD. IF SUCCESSFULLY DELETED THE RECORD  
, THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELSDF2  
PURPOSE: DELETE A RECORD FROM SEGMENT\_DATA\_FIELD  
ENTITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELSDF2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELSDF2(&DB\_ID, &RT\_ID, &STATUS)  
COBOL -- CALL "DELSDF2" USING  
DB-ID,  
RT-ID,  
STATUS.  
FORTRAN -- CALL DELSDF2(DBID, RTID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
SEGMENT\_DATA\_FIELD  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:  
-----

PS 620141100  
1 November 1985

DB\_ID - INT \*  
RT\_ID - CHAR \*  
STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELSDF3  
PURPOSE: DELETE A RECORD FROM SEGMENT\_DATA\_FIELD  
ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELSDF3  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELSDF3(%DB\_ID, %RT\_ID, %DF\_ID, %STATUS)  
COBOL -- CALL "DELSDF3" USING  
DB-ID,  
RT-ID,  
DF-ID,  
STATUS.  
FORTRAN -- CALL DELSDF3(DBID, RTID, DFID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*RT\_ID ;  
CHAR \*DF\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
SEGMENT\_DATA\_FIELD  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*  
RT\_ID = CHAR \*  
DF\_ID = CHAR \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELSEC  
PURPOSE: DELETE RECORD CLASS FROM SEC  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELSEC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

WHERE VIEW\_NO := "INPUT PARAMETER 1" ;  
SYNOPSIS  
C -- DELSEC( %AC\_NO, %STATUS)  
COBOL -- CALL "DELSEC" USING  
VIEW-NO,  
STATUS.  
FORTRAN -- CALL DELSEC(VIEWNO,STATUS)

INPUT:  
INT \*VIEW\_NO ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:  
THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
SEC  
WHERE VIEW\_NO EQUALS "INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

VIEW\_NO = INT \*

PS 620141100  
1 November 1985

STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPVIEW - DROP THE VIEW.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELSECR  
PURPOSE: DELETE RECORD CLASS FROM SEC\_RC\_COMPONENT  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELSECR  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

WHERE VIEW\_NO =: "INPUT PARAMETER 1" ;  
SYNOPSIS  
C -- DELSECR( @VIEW\_NO, @STATUS)  
COBOL -- CALL "DELSECR" USING  
VIEW-NO,  
STATUS.  
FORTRAN -- CALL DELSECR(VIEWNO,STATUS)

INPUT:  
INT \*VIEW\_NO ;

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:  
THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
SEC\_RC\_COMPONENT  
WHERE VIEW\_NO EQUALS INPUT PARAMETER 1".  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:  
-----

VIEW\_NO = INT \*

PS 620141100  
1 November 1985

STATUS - INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPVIEW - DROP THE VIEW.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELSN1  
PURPOSE: DELETE A RECORD FROM SCHEMA\_NAMES ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELSN1  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DELSN1(@DB\_ID,@STATUS)  
COBOL -- CAL "DELSN1" USING  
DB-ID,  
STATUS.  
FORTRAN -- CALL DELSN1(DBID,STATUS)

INPUT:

INT \*DB\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
SCHEMA\_NAMES  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DELSTM1  
PURPOSE: DELETE A RECORD FROM SET\_TYPE\_MEMBER ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELSTM1  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- DELSTM1(%DB\_ID,%STATUS)  
COBOL -- CALL "DELSTM1" USING  
DB-ID,  
STATUS.  
FORTRAN -- CALL DELSTM1(DBID,STATUS)

INPUT:

INT \*DB\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
SET\_TYPE\_MEMBER. IF SUCCESSFULLY DELETED THE RECORD ,  
THE STATUS  
AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

DB\_ID = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

**ROUTINES CALLED:**

-----  
: COSQLS - ORACLE ROUTINE  
: ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
: COEXEC - ORACLE ROUTINE  
: COBINDM - ORACLE ROUTINE

**CALLED DIRECTLY BY:**

-----  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELSTM2  
PURPOSE: DELETE A RECORD FROM SET\_TYPE\_MEMBER  
ENTITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELSTM2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

SYNOPSIS

C -- DELSTM2(ØDB\_ID, SET\_ID, ØSTATUS)  
COBOL -- CALL "DELSTM2" USING  
DB-ID,  
SET-ID,  
STATUS.  
FRTRAN -- CALL DELSTM2(DBID, SETID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*SET\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
SET\_TYPE\_MEMBER  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:  
-----

DB\_ID = INT \*  
SET\_ID = CHAR \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPSET - CONTROLS THE PROCESSING LOGIC FOR DELETING A  
SET FROM THE DATABAS  
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET  
BEING DROPPED  
SELSTM - DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE  
BEING DROPPED

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELSTM3  
PURPOSE: DELETE A RECORD FROM SET\_TYPE\_MEMBER ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELSTM3  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

```
C          --  DELSTM3(%DB_ID, %SET_ID,  
              %RT_ID_OF_MEMBER, %STATUS)  
  
COBOL     --  CALL "DELSTM3" USING  
              DB-ID,  
              SET-ID,  
              DF-ID,  
              STATUS.  
  
FORTRAN   --  CALL DELSTM3(DBID, SETID, DFID, STATUS)
```

INPUT:

```
INT  *DB_ID ;  
CHAR *SET_ID ;  
CHAR *RT_ID_OF_MEMBER ;
```

OUTPUT:

```
INT *STATUS ;
```

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM SET\_TYPE\_MEMBER IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND RETURN VALUE BOTH WILL BE 0 , OTHERWISE -1.

PS 620141100  
1 November 1985

ARGUMENTS:

-----  
DB\_ID = INT \*  
SET\_ID = CHAR \*  
RT\_ID\_OF\_MEMBER = CHAR \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELTEXT  
PURPOSE: DELETE A RECORD FROM DESC\_TEXT ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELTEXT  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- DELTEXT(%OBJECT\_TYPE, %OBJECT\_NO,  
%STATUS)  
  
COBOL -- CALL "DELTEXT" USING  
OBJECT-TYPE,  
OBJECT-NO,  
STATUS.  
  
FORTRAN -- CALL  
DELTEXT(OBJECT-TYPE, OBJECT-NO, STATUS)

INPUT:

CHAR \*OBJECT\_TYPE ;  
INT \*OBJECT\_NO ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DESC\_TEXT  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
OBJECT\_TYPE = CHAR \*  
OBJECT\_NO = INT \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELDDBF - DELETE ALL DATA FIELD ASSOCIATIONS WITH THE  
DATABASE  
DELDVRT - DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE  
DATABASE  
DELDBST - DELETE ALL RECORD SETS' ASSOCIATIONS WITH THE  
DATABASE  
DELDTNO - DELETE ALL DATA TYPES ASSOCIATED WITH A DOMAIN  
DELMDCR - DELETE ALL RELATION CLASSES FOR AN ENTITY  
DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED  
ATTRIBUTE CLASSES FROM TH  
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE  
DATA BASE.  
DRPDIV - DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED  
WITH VIEW  
DRPDM - PROGRAM NAME  
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING  
ENTITIES.  
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD.  
DRPKWC - OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP  
THEIR ASSOCIATIONS.  
DRPMOD - CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.  
DRPRCE - DROP A RELATION CLASS FOR AN ENTITY BEING  
DROPPED  
DRPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
RECORD COMMAND.  
DRPREL - CONTROLS THE PROCESSING LOGIC FOR THE "DROP  
RELATION" COMMAND.  
DRPSET - CONTROLS THE PROCESSING LOGIC FOR DELETING A  
SET FROM THE DATABAS  
DRPVIEW - DROP THE VIEW.



PS 620141100  
1 November 1985

FNDACH - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL  
FNDECM - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR  
          A GIVEN MODEL  
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET  
          BEING DROPPED  
SELSTM - DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE  
          BEING DROPPED  
DELAUC - DELETE INHERITED ATT\_USE, KEY\_CLASS\_MEMBER,  
DELDFL2 - CONTROLS THE DELETING OF DATA FIELDS  
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DELTXT  
PURPOSE: DELETE DESCRIPTION TEXT GIVEN THE OBJECT  
TYPE,  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DELTXT  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

DESCRIPTION TYPE AND THE OBJECT NUMBER.

SYNOPSIS

C -- DELTXT (OBJ\_TYPE, DESC\_TYPE, %OBJ\_NO,  
%STATUS)

COBOL -- CALL "DELTXT" USING  
OBJ-TYPE  
DESC-TYPE  
OBJ-NO  
STATUS

FORTRAN -- CALL DELTXT  
(OBJTYPE, DESC\_TYP, OBJNO, STATUS)

INPUT:

CHAR OBJ\_TYPE[]  
CHAR DESC\_TYPE[]  
INT \*OBJ\_NO

OUTPUT:

INT \*STATUS

DESCRIPTION

DELTXT DELETES ALL DESCRIPTION TEXT CORRESPONDING TO A

PARTICULAR  
OBJECT TYPE, OBJECT NUMBER AND DESCRIPTION TYPE USING A  
SQL DELETE  
COMMAND. IF THE DELETE WAS SUCCESSFUL, STATUS IS ZERO  
AND THE  
VALUE OF THE FUNCTION IS ZERO. IF UNSUCCESSFUL, STATUS  
AND THE  
FUNCTION VALUE ARE SET TO -1.

ARGUMENTS:

-----  
OBJ\_TYPE = CHAR []  
DESC\_TYPE = CHAR []  
OBJ\_NO = INT \*  
STATUS = INT \*

INCLUDE FILES:

-----  
STD TYP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
FILEINS - INSERT DESCRIPTION TEXT INTO CDM  
STRINS - INSERT DESCRIPTION TEXT INTO THE CDM

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DEPATT  
PURPOSE: SELECT ALL THE ATTRIBUTES IN THE  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DEPATT  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

FROM-MODEL STRUCTURE AND GENERATE  
NDDL COMMANDS TO CREATE THE ATTRIBUTES ,  
THEIR KEYWORDS ,  
ALIAS, THE KEYWORDS AND DESCRIPTIONS IN THE  
TO-MODEL

SYNOPSIS

```
C          --  DEPATT(%EC_NO, %MODEL_NAME,  
              %KEYWORD_FLAG,  
              %ALIAS_FLAG, %DESC_FLAG, %EC_NO) ;  
  
COBOL     --  CALL "DEPATT"  USING  
              MODEL-NO,  
              MODEL-NAME,  
              KEYWORD-FLAG,  
              ALIAS-FLAG,  
              DESC-FLAG,  
              EC-NO.  
  
FORTRAN   --  CALL DEPATT(MODELNO, MODELNAME,  
              KEYWORDFLAG, ALIASFLAG,  
              DESCFLAG, ECNO)
```

INPUT:

```
INT  *EC_NO ;  
CHAR *MODEL_NAME ;
```

INT \*KEYWORD\_FLAG ;  
INT \*ALIAS\_FLAG ;  
INT \*DESC\_FLAG ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO RETRIEVE RECORDS FROM  
ATTRIBUTE\_CLASS, ATTRIBUTE\_NAME, DOMAIN\_CLASS, THEN USE  
THE  
INFORMATION TO GENERATE NDDL COMMAND STATEMENT FOR  
EACH DEPENDANT ENTITY IN THE STRUCTURE.

ARGUMENTS:

-----  
EC\_NO = INT \*  
MODEL\_NAME = CHAR \*  
KEYWORD\_FLAG = CHAR \*  
ALIAS\_FLAG = CHAR \*  
DESC\_FLAG = CHAR \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COFETCH - ORACLE ROUTINE  
STRNCMP  
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN  
OBJECT TYPE AND NO  
GENATT - GENERATE A CREATE ATTRIBUTE.. COMMAND  
STRNCPY  
GENALI - GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT  
TYPE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE  
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
CESTRUC - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES FOR THE STRUCT

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DEPENT  
PURPOSE: SELECT ALL THE DEPENDANT ENTITY CLASSES  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DEPENT  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

IN THE STRUCTURE OF THE FROM-MODEL.

SYNOPSIS

```
C          --  DEPENT(&EC_NO, &MODEL_NAME,  
              &KEYWORD_FLAG,  
              &ALIAS_FLAG, &DESC_FLAG, EC_NO) ;  
  
COBOL     --  CALL "DEPENT"    USING  
              MODEL-NO,  
              MODEL-NAME,  
              KEYWORD-FLAG,  
              ALIAS-FLAG,  
              DESC-FLAG,  
              EC-NO.  
  
FORTRAN   --  CALL DEPENT(MODELNO, MODELNAME,  
              KEYWORDFLAG, ALIASFLAG,  
              DESCFLAG, ECNO)
```

INPUT:

```
INT  *EC_NO ;  
CHAR *MODEL_NAME ;  
INT  *KEYWORD_FLAG ;  
INT  *ALIAS_FLAG ;  
INT  *DESC_FLAG ;  
INT  *EC_NO ;
```

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO RETRIEVE RECORDS FROM  
ATTRIBUTE\_CLASS, ATTRIBUTE\_NAME, DOMAIN\_CLASS, THEN USE  
THE  
INFORMATION TO GENERATE NDDL COMMAND STATEMENT.

ARGUMENTS:

-----  
EC\_NO = INT \*  
MODEL\_NAME = CHAR \*  
KEYWORD\_FLAG = CHAR \*  
ALIAS\_FLAG = CHAR \*  
DESC\_FLAG = CHAR \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COFETCH - ORACLE ROUTINE  
STRNCMP  
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN  
OBJECT TYPE AND NO  
BLECLST - THIS ROUTINE CREATES A ROW IN THE UNBOUNDED  
EC LIST  
GENENT - GENERATE A CREATE ENTITY..OWNED  
ATTRIBUTE..KEYWORD COMMAND  
STRNCPY  
GENALI - GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT  
TYPE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE  
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
CESTRUC - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES FOR THE STRUCT

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DEPFROM  
**PURPOSE:** GENERATE CREATE RELATION, DESCRIBE  
COMMANDS IN THE TO-MODEL  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DEPFROM  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE SELECTS ALL THE RELATIONS IN WHICH THE FROM-ENTITY IS THE DEPENDENT ENTITY. IF THE SAME ENTITY EXISTS IN THE TO-MODEL BUT NOT THE RELATION, THEN A CREATE RELATION AND DESCRIBE COMMANDS ARE GENERATED FOR THE FROM-RELATION.

**ARGUMENTS:**  
-----

FROM-EC-NO = DSPLY [S9(9)]  
TO-EC-NO = DSPLY [S9(9)]  
FROM-EC-NAME = DSPLY [X(30)]  
TO-EC-NAME = DSPLY [X(30)]  
TO-MOD-NO = DSPLY [S9(9)]  
FROM-MOD-NO = DSPLY [S9(9)]  
KW-FLAG = DSPLY [9]  
DESC-FLAG = DSPLY [9]  
COMBINE-TYPE = DSPLY [9]  
RCNO-LIST = RECRD

**INCLUDE FILES:**  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
RELTBL - LIST OF RELATION CLASSES IN A MODEL



ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
SELECNM - RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS  
VERRC - verify the existence of a relation class.  
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A  
MODEL  
VERRCNM - VERIFY EXISTANCE OF A RELATION IN A MODEL  
GENRNME - FORMAT THE CREATE RELATION CLAUSE  
BLKCL1 - STORE ALL KEY CLASS INFO FOR AN ENTITY IN A LIST  
MIGREL - GENERATE A MIGRATES CLAUSE FOR A CREATE  
RELATION COMMAND  
CMBRKW - SELECT AND GENERATE RELATION CLASS KEYWORDS  
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN  
OBJECT TYPE AND NO  
ERRPRO

CALLED DIRECTLY BY:

-----  
CERELS - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES IN A RELATION.  
CMBENT - CONROLS THE PROCESSING LOGIC FOR THE COMBINE  
ENTITY COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DEPREL  
PURPOSE: FOR EACH LEVEL OF RELATIONS IN STRUCTURE  
GENERATE  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DEPREL  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

THE NDDL COMMANDS TO CREATE THE RELATIONSS  
, MIGRATE  
ATTRIBUTE AND ADD THE KEY CLASSES FOR EACH  
DEPENDENT  
ENTITY IN THE LEVEL

SYNOPSIS

```
C          --  DEPREL(&MODEL_NO, &MODEL_NAME,  
              &KEYWORD_FLAG,  
              &DESC_FLAG, EC_NO) ;  
  
COBOL     --  CALL "DEPREL"   USING  
              MODEL-NO,  
              MODEL-NAME,  
              KEYWORD-FLAG,  
              DESC-FLAG,  
              EC-NO.  
  
FORTRAN   --  CALL DEPREL(MODELNO, MODELNAME,  
              KEYWORDFLAG,  
              DESCFLAG, ECNO)
```

INPUT:

```
INT  *MODEL_NO ;  
CHAR *MODEL_NAME ;
```

INT \*KEYWORD\_FLAG ;  
INT \*DESC\_FLAG ;  
INT \*EC\_NO ;

DESCRIPTION:

ARGUMENTS:

-----  
MOD\_NO = INT \*  
EC\_NO = INT \*  
MODEL\_NAME = CHAR \*  
KEYWORD\_FLAG = CHAR \*  
DESC\_FLAG = CHAR \*  
RC\_DEPKC\_LIST = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
BLRCKC1 - FOR EACH LEVEL OF RELATIONS IN STRUCTURE  
COFETCH - ORACLE ROUTINE  
SELECNM - RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS  
GENREL - GENERATE CREATE RELATION ..MIGRATES..KEYWORD  
COMMAND  
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN  
OBJECT TYPE AND NO  
ALLKEY - GENERATES KEY CLASS FOR AN ENTITY  
STRNCPY  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE  
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
CESTRUC - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES FOR THE STRUCT

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DESCRB  
PURPOSE: COMMAND PROCESSOR FOR THE NDDL DESCRIBE  
COMMAND  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DESCRB  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

SYNOPSIS

C	--	DESCRB (#MODEL_NO)	
COBOL	--	CALL "DESCRB"	USING MODEL-NO.
FORTRAN	--	CALL DESCRB (MODELNO)	

INPUT:

INT \*MODEL\_NO

OUTPUT:

DESCRIPTION

DESCRB VERIFIES THAT THE DESCRIPTION TYPE AND THE  
OBJECT TO BE  
DESCRIBED IS VALID AND EXISTS. THE DESCRIPTION INPUT  
TYPE (EDITOR,  
FILE OR ENTERED TEXT) IS DETERMINED. THE DESCRIPTIONS  
ARE THEN  
ENTERED INTO THE CDM.

ARGUMENTS:

-----  
MODEL\_NO = INT \*

INCLUDE FILES:

-----  
STDYTP - STANDARD TYPE DEFINITIONS  
LISTID - PROVIDES LIST OF PARSED OBJECTS

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VEROBJ - VERIFY THAT THE OBJECT EXISTS.  
TXTTYT - DETERMINE SOURCE OF DESCRIPTION TEXT  
RDDESC - STORE DESCRIPTION ON THE CDM  
LOADESC - LOAD DESCRIPTION FROM TEXT EDITOR  
TOUPPER  
VERDSTP - VALIDATE A USER ENTERED DESCRIPTION TYPE  
SPRINTF  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DLDSL2  
PURPOSE: DELETE A RECORD FROM DF\_SET\_LINKAGE ENTITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DLDSL2  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DLDSL2(®DB\_ID, SET\_ID, ®STATUS)  
  
COBOL -- CALL "DLDSL2" USING  
                                  DB-ID,  
                                  SET-ID,  
                                  STATUS.  
  
FRTRAN -- CALL DLDSL2(DBID, SETID, STATUS)

INPUT:

INT \*DB\_ID ;  
CHAR \*SET\_ID ;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DF\_SET\_LINKAGE  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DB\_ID = INT \*

PS 620141100  
1 November 1985

SET\_ID = CHAR \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPSET - CONTROLS THE PROCESSING LOGIC FOR DELETING A  
SET FROM THE DATABAS

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DLMDAUC  
PURPOSE: DELETE ATTRIBUTE USE CLASSES ASSOCIATED  
WITH ENTITY  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DLMDAUC  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE DELETES THE ATTRIBUTE USE CLASSES FOR A GIVEN ENTITY.

ARGUMENTS:  
-----

EC-NO = DSPLY [S9(9)]  
ERROR-STATUS = DSPLY [S9(9)]

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

DELAUC - DELETE INHERITED ATT USE, KEY CLASS MEMBER,  
SELECNM - RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO



PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----

**FNDECM - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR  
A GIVEN MODEL**

**USED IN MAIN PROGRAM(S):**

-----

**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**

NDDL COMMAND PROCESSOR Module Documentation

NAME: DLMIGRC  
PURPOSE: DELETE MIGRATING KEY CLASS  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DLMIGRC  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
ONLY THRU A SINGLE RELATION CLASS

SYNOPSIS

C	--	DLMIGRC(%TAG_NO,%RC_NO)	
COBOL	--	CALL "DLMIGRC"	USING TAG-NO RC_NO.
FORTTRAN	--	CALL DLMIGRC (TAGNO,RCNO)	

INPUT:

INT \*TAG\_NO  
INT \*RC\_NO

OUTPUT:

DESCRIPTION

THIS ROUTINE USES AN ORACLE SELECT STATEMENT TO  
RETRIEVE ALL  
MIGRATING KEY CLASSES GIVEN A TAG NUMBER. ROUTINES  
ARE CALLED  
WHICH, FOR EACH NEW TAG NUMBER SELECTED, DELETE THE

INHERITED  
ATTRIBUTE CLASS, ATTRIBUTE USE CLASS, COPY CROSS  
REFERENCE AND  
KEY CLASS MEMBER ENTITIES WHOSE TAG NUMBERS MATCH.  
THE QUERY IS PHRASED SUCH THAT ONLY THE ATTRIBUTES  
INHERITED THRU A SINGLE RELATION CLASS ARE DELETED,  
NOT EVERYWHERE THAT ATTRIBUTE IS MIGRATED TO.

ARGUMENTS:

-----  
TAG\_NO = INT \*  
RC\_NO = INT \*  
RTNSTA = INT \*

INCLUDE FILES:

-----  
STD TYP - STANDARD TYPE DEFINITIONS  
UNI QENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COFETCH - ORACLE ROUTINE  
DELIAUC - DELETE A REORD FROM INHERITED\_ATT\_USE  
SPRINTF  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELKCMT - DELETE A RECORD FROM KEY\_CLASS\_MEMBER  
DELAUCL - DELETE A RECORD FROM ATTRIBUTE\_USE\_CL  
COEXEC - ORACLE ROUTINE  
CODFINN - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DRPMGRC - DELETE MIGRATED KEY CLASS MEMBERS ASSOCIATED  
WITH A KEY CLASS

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DOMUSAG  
PURPOSE: DETERMINE IF DOMAIN ASSOCIATED WITH  
VIEWS, DATAFIELDS, ATTRIB  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DOMUSAG  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
DETERMINE IF A DOMAIN BEING DROPPED IS CURRENTLY  
BEING ASSOCIATED WITH ANY DATA FIELDS  
VIEWS OR ATTRIBUTE CLASSES.

SELECT THE DATA TYPE NAME, QUALIFYING ON  
THE DOMAIN NUMBER WHICH IS THE UNIQUE  
NUMBER ASSIGNED TO THE DOMAIN CLASS BEING  
DROPPED.

ARGUMENTS:

-----  
DOM-NO = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
VERDFDT - DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH  
A PROJECT DF  
VERDIDT - DETERMINE IF A DATA TYPE IS ASSOCIATED WITH ANY  
DATA ITEMS  
ERRPRO

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----  
DRPDOM - PROGRAM NAME

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DPKCLST  
PURPOSE: CREATE AN KEY\_CLASS\_LIST TABLE CONTAINING ALL THE ENTITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DPKCLST  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
CLASS MEMBERS IN THE STRUCTURE, THIER  
KEY CLASS NOS AND NAMES AND KEY CLASS TAG  
NUMBERS AND NAMES.

SYNOPSIS

C -- DPKCLST(&MODEL\_NO, &EC\_NO) ;  
COBOL -- CALL "DPKCLST" USING  
MODEL-NO  
EC-NO.  
FORTRAN -- CALL DPKCLST(MODELNO, ECNO)

INPUT:

INT \*MODEL\_NO ;  
INT \*FROM\_EC\_NO ;

DESCRIPTION:

ARGUMENTS:

-----  
MODEL\_NO = INT \*  
FROM\_EC\_NO = INT \*

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
COFETCH - ORACLE ROUTINE  
AKCROW - THIS ROUTINE ADDS A ROW TO THE UNBOUNDED  
          KEY\_CLASS\_LIST  
  
STRNCPY  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE  
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
CESTRUC - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
          ENTITIES FOR THE STRUCT

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPAC  
PURPOSE: DELETE OWNED ATTRIBUTES ASSOCIATED WITH  
ENTITY  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DRPAC  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
DRPAC CONTROLS THE PROCESSING LOGIC FOR THE DROP  
OWNED ATTRIBUTE CLAUSE IN THE ALTER ENTITY  
COMMAND. THE CLAUSE IS OPTIONAL. WHEN  
SPECIFIED, MAY BE A LIST OF ATTRIBUTE NAMES.

PROCESSING CONTINUES, REGARDLESS OF  
ERRORS, UNTIL THE LIST OF ATTRIBUTE  
NAMES IS EXHAUSTED.

ARGUMENTS:

-----  
MODEL-NO = DSPLY [S9(9)]  
EC-NO = DSPLY [S9(9)]

INCLUDE FILES:

-----  
LISTNOS - VALID LIST NUMBERS  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A  
MODEL



PS 620141100  
1 November 1985

UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DELOAC - DELETE ALL OWNED AND INHERITED ATTRIBUTES  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
ERRPRO

-----  
CALLED DIRECTLY BY:

-----  
ALTENT - CONTROL PROCESSING FOR ALTER ENTITY CLASS  
COMMAND.

-----  
USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPALI  
PURPOSE: DROP THE ALIAS FOR AN ENTITY OR ATTRIBUTE.  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DRPALI  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
DROP ALIAS WILL DROP THE ALIAS FOR AN ENTITY  
OR ATTRIBUTE.

OBTAIN OBJECT ID, OBJECT TYPE AND ALIAS  
ID, VERIFY IF THE ALIAS TO BE DROPPED IS  
AN ENTITY OR ATTRIBUTE, THEN CALL  
ROUTINES TO DELETE THE ALIASES. CONTINUE  
PROCESSING TILL NO MORE RECORDS TO BE  
DELETED

ARGUMENTS:

-----  
MODEL-NO = DSPLY [S9(9)]

INCLUDE FILES:

-----  
LISTNOS - VALID LIST NUMBERS

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
LOWUPP - CONVERT A STRING TO UPPER CASE CHARACTERS  
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A  
MODEL  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A  
MODEL

PS 620141100  
1 November 1985

GETEGAL - VERIFIES THE EXISTANCE OF AN ALIAS FOR AN ENTITY  
GETACAL - VERIFY THE EXISTANCE OF AN ALIAS FOR AN  
ATTRIBUTE

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPATT  
PURPOSE: CONTROLS THE DROPPING OF USER SPECIFIED  
ATTRIBUTE CLASSES F  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DRPATT  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
- THE DROP ATTRIBUTE CLASS COMMAND PROCESSOR  
CONTROLS THE DROPPING OF USER SPECIFIED  
ATTRIBUTE CLASSES FROM THE CDM.

ARGUMENTS:

-----  
MODEL-NO = DSPLY [S9(9)]

INCLUDE FILES:

-----  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER  
VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A  
MODEL  
DELOAC - DELETE ALL OWNED AND INHERITED ATTRIBUTES

PS 620141100  
1 November 1985

DELAC - DELETE A RECORD FROM ATTRIBUTE CLASS  
DELACNM - DELETE RECORD CLASS FROM ATTRIBUTE\_NAME WHERE  
AC\_NO  
DELACKW - DELETE A RECORD FROM AC KEYWORD  
DELTEXT - DELETE A RECORD FROM DESC TEXT ENITY  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
ERRPRO

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPDB  
PURPOSE: CONTROLS THE PROCESSING LOGIC FOR  
DELETING THE DATA BASE.  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: DRPDB  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR DELETING THE DATA BASE FROM THE SYSTEM AND ALL ENTITIES ASSOCIATED WITH THE DATA BASE. DB FROM SCHEMA-NAMES, DB FROM RECORD-TYPES, DB FROM DB-AREA-ASSIGNMENT, DB FROM DATA-FIELD, DB FROM RECORD-SET, DB FROM RECORD SET-TYPE-MEMBERS, DB FROM RECORD-KEY, DB FROM RECORD-KEY-MEMBER, DB FROM AUC-ST-MAPPING, DB FROM RC-BASED-REC-SET, DB FROM IMS-SEGMENT-SIZE, DB FROM DATA-BASE-AREA, DB FROM DF-SET-LINKAGE, AND DB FROM SEGMENT-DATA-FIELD. ADDITIONALLY ALL TEXTUAL DESCRIPTIONS ARE DELETED FOR THE DATA BASE AND ASSOCIATED OBJECTS: SET, RECORDS, AND DATA FIELDS. EACH OBJECT BEING DELETED WILL HAVE THEIR UNIQUE OBJECT NUMBERS RETURNED TO THE CDM FOR REUSE.

AFTER EXTRACTING THE DBMS NAME, DATA BASE NAME AND HOST NAME FROM THE PARSER LISTS, VERIFY THAT THE DATA BASE INFORMATION IS CORRECT, AND THEN DELETE THE DATA BASE FROM THE ASSOCIATED ENTITIES.

ASSOCIATED WITH THE SETS, AND  
ADD THE SET-NO BACK TO THE CDM FOR REUSE.  
DELDBST IS A PRC (NDML) ROUTINE.

INCLUDE FILES:

-----  
LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
LOWUPP - CONVERT A STRING TO UPPER CASE CHARACTERS  
VERDBAS - VERIFY A DATA BASE FOR A GIVEN NAME AND HOST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DELDBS1 - DELETE A RECORD FROM DATA BASE ENITY  
DELSN1 - DELETE A RECORD FROM SCHEMA NAMES ENITY  
DELDVRT - DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE  
DATABASE  
DELDA1 - DELETE A RECORD FROM DB AREA ASSIGNMENT ENITY  
DELDDBF - DELETE ALL DATA FIELD ASSOCIATIONS WITH THE  
DATABASE  
DELDBST - DELETE ALL RECORD SETS' ASSOCIATIONS WITH THE  
DATABASE  
DELSTM1 - DELETE A RECORD FROM SET TYPE MEMBER ENITY  
DELRKY1 - DELETE A RECORD FROM RECORD KEY ENITY  
DELRKM1 - DELETE A RECORD FROM RECORD KEY MEMBER ENITY  
DELASM1 - DELETE A RECORD FROM AUC ST MAPPING ENITY  
DELRBR1 - DELETE A RECORD FROM RC BASED REC SET ENITY  
DELISS1 - DELETE A RECORD FROM IMS SEGMENT SIZE ENITY  
DELDBA1 - DELETE A RECORD FROM DATA BASE AREA ENITY  
DELDL1 - DELETE A RECORD FROM DF SET LINKAGE ENITY  
DELSDF1 - DELETE A RECORD FROM SEGMENT DATA FIELD ENITY  
PDFDB - RETRIEVES PROJECT DATA FIELDS FOR A GIVEN DATA  
BASE  
DELPCB - DELETE A RECORD FROM PSB PCB ENITY  
DELTEXT - DELETE A RECORD FROM DESC TEXT ENITY  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
ERRPRO

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

PS 620141100  
1 November 1985

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPDF  
PURPOSE: DELETE A RECORD FROM DATA\_FIELD ENITY  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DRPDF  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- DRPDF(DF\_NO, @STATUS)  
  
COBOL -- CALL "DRPDF" USING  
          DF-NO,  
          STATUS.  
  
FORTRAN -- CALL DRPDF(DFNO, STATUS)

INPUT:

INT \*DF\_NO;

OUTPUT:

INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO DELETE A RECORD FROM  
DATA\_FIELD  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----  
DF NO - INT \*  
STATUS - INT \*

PS 620141100  
1 November 1985

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
DELDFL2 - CONTROLS THE DELETING OF DATA FIELDS

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPDIV  
PURPOSE: DELETE DATA ITEM DESCRIPTION TEXTS  
ASSOCIATED WITH VIEW  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DRPDIV  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

EXTS ASSOCIATED WITH VIEW

DESCRIPTION :-

This routine will given a view number select all data items associated with the view, delete all textula descriptions of the data items and send the cdm the deleted data item numbers for reuse.

ARGUMENTS:

-----

VIEW-NO = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----

PS 620141100  
1 November 1985

DELTEXT - DELETE A RECORD FROM DESC TEXT ENITY  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
ERRPRO

CALLLED DIRECTLY BY:

-----  
DRPVIEW - DROP THE VIEW.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DRPDOM  
**PURPOSE:** PROGRAM NAME  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** PROGRAM  
**SOURCE FILE:** DRPDOM  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
THE DROP DOMAIN COMMAND OBTAINS THE DOMAINS TO BE DROPPED, AND DELETES THE DOMAIN CLASS AND ALL ASSOCIATED DATA TYPES.

OBTAIN DOMAIN NAMES FROM PARSER AND  
VERIFY THA THE DOMAIN TO BE DELETED DOES  
EXIST. THEN CHECK THE DOMAIN USAGE, IF  
NOT BEING USED BY OTHERS, DELETE THE DATA  
TYPES AND THE DOMAIN. CONTINUE  
PROCESSING TILL NO MORE DOMAINS REMAIN TO  
BE DELETED.

**INCLUDE FILES:**

-----  
LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

**ROUTINES CALLED:**

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERDOM - VERIFY THE EXISTENCE OF A DOMAIN CLASS IN THE  
CDM  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERACDT - DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH  
ATTRIBUTE CLASSES  
DOMUSAG - DETERMINE IF DOMAIN ASSOCIATED WITH  
VIEWS.DATAFIELDS.ATTRIBUTES

PS 620141100  
1 November 1985

DELDTNO - DELETE ALL DATA TYPES ASSOCIATED WITH A DOMAIN  
DELDTD - DELETE RECORD CLASS FROM USER\_DEF\_DATA-TYPE  
DELDOM - DELETE RECORD CLASS FROM DOMAIN\_CLASS  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENITY  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.

-----  
CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPDT  
PURPOSE: DROP A DATA TYPE WITHIN A DOMAIN.  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DRPDT  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
THIS CLAUSE WILL DROP A DATA TYPE WITHIN A  
DOMAIN AFTER VERIFYING THAT A DATA TYPE  
DOES EXIST

OBTAIN DATA-TYPE-NAMES FROM THE PARSER  
LISTS, THEN VERIFY THAT THE DATA-TYPE-  
NAME DOES EXIST IF IT DOES, MAKE SURE IT  
IS NOT USED BY OTHERS, AND THEN DELETE IT

ARGUMENTS:

-----  
DOM-NO, DATA-TYPE-NAME =

ROUTINES CALLED:

-----  
VERDTD - VERIFY THE EXISTENCE OF A DATA TYPE IN A GIVEN  
DOMAIN  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DTUSAGE - DETERMINE ASSOCIATIONS OF A DATA TYPE  
DELDT - DELETE RECORD CLASS FROM USER\_DEF\_DATA\_TYPE

CALLED DIRECTLY BY:

-----  
PROCDT - PROGRAM NAME

USED IN MAIN PROGRAM(S):

PS 620141100  
1 November 1985

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPENT  
PURPOSE: CONTROL THE PROCESSING LOGIC FOR DELETING ENTITIES.  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DRPENT  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
THE DROP ENTITY COMMAND PROCESSOR WILL CONTROL THE THE PROCESSING LOGIC FOR DELETING ENTITIES AND THEIR ASSOCIATED NAMES, KEYWORDS, DESCRIPTIONS, OWNED ATTRIBUTE, ATTRIBUTE USE CLASSES AND RELATION CLASSES

ARGUMENTS:

-----  
MODEL-NO = DSPLY [S9(9)]

INCLUDE FILES:

-----  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
LISTNOS - VALID LIST NUMBERS

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A MODEL  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
FNDOAC - DELETES ALL OWNED ATTRIBUTES FOR A GIVEN ENTITY CLASS  
FNDAUC - DELETE ALL ATTRIBUTE USE CLASSES FOR A GIVEN ENTITY CLASS

PS 620141100  
1 November 1985

DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER  
DRPRCE - DROP A RELATION CLASS FOR AN ENTITY BEING  
          DROPPED  
DELECNM - DELETE THE RECORD FROM ENTITY\_NAME  
DELECKW - DELETE A RECORD FROM EC\_KEYWORD  
DELEC   - DELETE THE RECORD FROM ENTITY\_CLASS  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
          LINKED  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENITY  
CPFNXT  - ACCESS THE NEXT TOKEN IN A PARSER LIST.

-----  
CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPFLD  
PURPOSE: CONTROLS THE PROCESSING LOGIC FOR  
DROPPING A DATA FIELD.  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: DRPFLD  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR DROPPING A  
DATA FIELD FROM THE DATA BASE.

FETCH THE DATA BASE ID, RECORD TYPE AND DATA FIELD. AFTER  
VERIFYING THAT THE DATA FIELD DOES EXIST, DELETE  
THE FIELD AND ITS ASSOCIATIONS. REPEAT TILL NO MORE  
DATA FIELDS REMAIN.

INCLUDE FILES:  
-----

UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERDBAS - VERIFY A DATA BASE FOR A GIVEN NAME AND HOST  
VERDFLD - VERIFIES THE EXISTENCE OF A DATA FIELD FOR A  
GIVEN DATA BASE  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO

LINKED

DELTEXT - DELETE A RECORD FROM DESC TEXT ENITY  
VERDSL3 - RETRIEVES SET ID FOR TOTAL DATA BASE  
DELDSL3 - DELETE A RECORD FROM DF SET LINKAGE ENITY  
DELSTM3 - DELETE A RECORD FROM SET TYPE MEMBER ENITY  
DELRST3 - DELETE A RECORD FROM RECORD SET ENITY  
DELASM2 - DELETE A RECORD FROM AUC ST MAPPING ENITY  
DELRBR3 - DELETE A RECORD FROM RC BASED REC SET ENITY  
DELSDF3 - DELETE A RECORD FROM SEGMENT DATA FIELD ENITY  
DELDL3 - DELETE A RECORD FROM DATA FIELD ENITY  
DELRKM3 - DELETE A RECORD FROM RECORD KEY MEMBER ENITY  
DELRKY2 - DELETE A RECORD FROM RECORD KEY ENITY  
PDFDF - RETRIVES ALL THE PROJ DATA FIELD OR DF BEING  
DROPPED  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
ERRPRO

-----  
CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

-----  
USED IN MAIN PROGRAM(S):

NDDL MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPKC  
PURPOSE: CONTROLS THE PROCESSING LOGIC FOR THE  
"DROP KEY CLASS".  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DRPKC  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY  
CLASS" CLAUSE IN THE ALTER ENTITY COMMAND. THE  
CLAUSE CONSISTS OF A LIST OF ONE OR MORE KEY  
CLASS NAMES.  
-

ARGUMENTS:

-----  
EC-NO = DSPLY [S9(9)]

INCLUDE FILES:

-----  
LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERKC - VERIFY THE EXISTENCE OF A KEY CLASS IN A MODEL  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERCRC - DETERMINE IF A KEY CLASS IS PART OF A COMPLETE  
RELATION  
DRPMGKM - DELETE MIGRATED KEY CLASSES ASSOCIATED WITH A  
KEY CLASS

PS 620141100  
1 November 1985

DELKCN - DELETE A RECORD FROM KEY CLASS MEMBER  
DELCHPR - DELETE A RECORD FROM COMPLETE RELATION  
DELKC - DELETE A RECORD FROM KEY CLASS  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
CPPNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST  
ERRPRO

-----  
CALLED DIRECTLY BY

ALTENT - CONTROL PROCESSING FOR ALTER ENTITY CLASS  
COMMAND

USED IN MAIN PROGRAM(S)

DDDL MAIN - MAIN PROGRAM FOR THE DDOL COMMAND PROCESSOR

**DDDL COMMAND PROCESSOR Module Documentation**

<b>NAME</b>	<b>DRPKW</b>
<b>PURPOSE</b>	<b>DROP A KEYWORD ASSOCIATION FROM EITHER AN ATTRIBUTE, ENTITY</b>
<b>LANGUAGE</b>	<b>VAX-11 COBOL</b>
<b>MODULE TYPE</b>	<b>SUBROUTINE</b>
<b>SOURCE FILE</b>	<b>DRPKW</b>
<b>SOURCE FILE TYPE</b>	<b>COB</b>
<b>HOST</b>	
<b>SUBSYSTEM</b>	<b>CDM</b>
<b>SUBDIRECTORY</b>	
<b>DOCUMENTATION GROUP</b>	<b>PS41100</b>

**DESCRIPTION**

**DROP KEYWORD CLAUSE WILL DROP A KEYWORD ASSOCIATION FROM EITHER AN ATTRIBUTE, ENTITY OR RELATION CLASS THIS CLAUSE IS CALLED FROM THE ALTER ENTITY, ATTRIBUTE AND RELATION COMMANDS**

**ARGUMENTS**

**OBJ TYPE - RECRD**  
**OBJ ID - DSPLY (S9(9))**  
**RTH STATUS - DSPLY (S9(9))**

**INCLUDE FILES**

**LISTNOS - VALID LIST NUMBERS**

**ROUTINES CALLED**

<b>CPFONE</b>	<b>- EXTRACT THE FIRST TOKEN FROM THE NAMED LIST</b>
<b>LOWUPP</b>	<b>- CONVERT A STRING TO UPPER CASE CHARACTERS</b>
<b>VERKW</b>	<b>- verify the existence of a keyword.</b>
<b>UERROR</b>	<b>- ISSUE A MESSAGE TO THE USER, CONSIDERED A</b>
<b>DELKWAC</b>	<b>- DELETE A RECORD FROM AC_KEYWORD</b>
<b>DELKVEC</b>	<b>- DELETE A RECORD FROM EC_KEYWORD</b>
<b>DELKWRC</b>	<b>- DELETE A RECORD FROM RC_KEYWORD</b>
<b>CPFNXT</b>	<b>- ACCESS THE NEXT TOKEN IN A PARSER LIST.</b>

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----  
**ALTENT** - CONTROL PROCESSING FOR ALTER ENTITY CLASS  
COMMAND.  
**ALTREL** - CONTROLS PROCESSING LOGIC FOR THE ALTER  
RELATION COMMAND  
**ALTATT** - THE ALTER ATTRIBUTE COMMAND PROCESSOR ALTERS

**USED IN MAIN PROGRAM(S):**

-----  
**NDDL/MAIN** - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DRPKWC  
**PURPOSE:** OBTAIN THE USED IDENTIFIED KEYWORD, THEN  
DROP THEIR ASSOCIA  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** PROGRAM  
**SOURCE FILE:** DRPKWC  
**SOURCE FILE TYPE:** .COB  
**HOT:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
OBTAIN THE USED IDENTIFIED KEYWORD, THEN  
DROP THEIR ASSOCIATIONS WITH ATTRIBUTE CLASS,  
ENTITIES AND RELATIONS AND THEN DROP THE  
KEYWORD.

FOR EACH KEYWORD OBTAINED, VERIFY  
THAT THE KEYWORD DOES EXIST, THEN  
DELETE THE ITS ASSOCIATION WITH  
ATTRIBUTES, ENTITIES, RELATION CLASS,  
AND FINALLY THE KEYWORD ITSELF. THE  
ADD THE KEYWORD NUMBER BACK TO THE  
REUSABLE POOL OF KEYWORD NUMBERS.

**INCLUDE FILES:**

-----  
LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

**ROUTINES CALLED:**

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERKW - verify the existence of a keyword.  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DELKWAC - DELETE A RECORD FROM AC\_KEYWORD  
DELKWEC - DELETE A RECORD FROM EC\_KEYWORD  
DELKWRC - DELETE A RECORD FROM RC\_KEYWORD

PS 620141100  
1 November 1985

DELKW - DELETE A RECORD FROM AC\_KEYWORD  
DELTEXT - DELETE A RECORD FROM DESC TEXT ENTITY  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
CPPNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.

**CALLED DIRECTLY BY:**

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMA' PROCESSOR

**MDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DRPHAP  
**PURPOSE:** COMMAND PROCESSOR FOR THE DROP MAP COMMAND  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** DRPHAP  
**SOURCE FILE TYPE:** C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION**

**SYNOPSIS**

C	DRPHAP
COBOL	CALL "DRPHAP"
FORTRAN	CALL DRPHAP

**INPUT**

**OUTPUT**

**DESCRIPTION**

DRPHAP IS THE COMMAND PROCESSOR FOR THE DROP MAP  
COMMAND. AFTER  
VERIFYING THE MAP NAME, A ROUTINE IS CALLED WHICH  
REMOVES THE MAP  
FROM THE DATABASE.

**INCLUDE FILES:**

-----  
STDTYP - STANDARD TYPE DEFINITIONS  
LISTID - PROVIDES LIST OF PARSED OBJECTS

**ROUTINES CALLED:**

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A  
MODEL  
  
SPRINTF  
UERROR - ISSUE A MESSAGE TO THE USER. CONSIDERED A  
DELRCS - DELETE RECORD CLASS FROM RC BASED REC SET  
VERRCS - REPORT WHETHER A RC TO SET MAP EXISTS.  
DELPDFT - DELETE A RECORD FROM PROJECT DATA FIELD ENTITY  
DELASH - DELETE FROM ATTRIBUTE USE CLASS MAPPING  
PDFSRCH - DETERMINES IF A TAG HAS A PRIMARY MAP TO DATA  
FIELD  
VERASH - DETERMINE IF ANY AUC/SET MAPPINGS EXIST FOR AN  
AUC  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
VERREL - VERIFY THE DEPENDENT AND INDEPENDENT ENTITIES  
IN THE RELATION  
VERAUC - VERIFY THE EXISTENCE OF AN ATTRIBUTE USE CLASS

**CALLED DIRECTLY BY:**

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

**USED IN MAIN PROGRAM(S):**

-----  
NDDL MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDML COMMAND PROCESSOR Module Documentation**

**NAME:** DRPHGKM  
**PURPOSE:** DELETE MIGRATED KEY CLASSES ASSOCIATED  
WITH A KEY CLASS  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DRPHGKM  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
GIVEN A KEY CLASS, DELETE ALL ITS MIGRATING KEY  
CLASS MEMBERS.

EXECUTE AN NDML SELECT STATEMENT TO  
RETRIEVE ALL THE KEY CLASS MEMBERS FOR  
THIS KEY CLASS BY TAG NO. FOR EACH KEY  
CLASS MEMBER RETRIEVED, DELETE THE KEY  
CLASS MEMBER AND ITS MIGRATIONS; I.E.  
ATTRIBUTE USE CLASS AND INHERITED  
ATTRIBUTE USE CLASS.

**ARGUMENTS:**

-----  
KC-NO - DSPLY [S9(9)]  
RETURN-STATUS - DSPLY [S9(9)]

**INCLUDE FILES:**

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**

-----  
DELMIGK - DELETE MIGRATING KEY CLASS  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----

**DRPKC - CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY  
CLASS".**

**USED IN MAIN PROGRAM(S):**

-----

**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DRPMGRC  
**PURPOSE:** DELETE MIGRATED KEY CLASS MEMBERS  
ASSOCIATED WITH A KEY CLA  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DRPMGRC  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
GIVEN A KEY CLASS AND A RELATION CLASS, DELETE  
ALL OF THE MIGRATING KEY CLASS MEMBERS THRU THIS  
RELATION CLASS.

EXECUTE AN NDML SELECT STATEMENT TO  
RETRIEVE ALL THE KEY CLASS MEMBERS FOR  
THIS KEY CLASS BY TAG NO. FOR EACH KEY  
CLASS MEMBER RETRIEVED, DELETE THE KEY  
CLASS MEMBER AND ITS MIGRATIONS; I.E.  
ATTRIBUTE USE CLASS AND INHERITED  
ATTRIBUTE USE CLASS.

**ARGUMENTS:**

-----  
KC-NO = DSPLY [S9(9)]  
RC-NO = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

**INCLUDE FILES:**

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**

-----  
DLMIGRC - DELETE MIGRATING KEY CLASS

PS 620141100  
1 November 1985

UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

-----  
CALLED DIRECTLY BY:

DRPMIG - CONTROLS THE PROCESSING LOGIC FOR THE "DROP  
MIGRATES" CLAUSE.  
DRPREL - CONTROLS THE PROCESSING LOGIC FOR THE "DROP  
RELATION" COMMAND.

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DRPHIG  
**PURPOSE:** CONTROLS THE PROCESSING LOGIC FOR THE  
"DROP MIGRATES" CLAUS  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DRPHIG  
**SOURCE FILE TYPE:** COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION**

-----  
CONTROLS THE PROCESSING LOGIC FOR THE "DROP  
MIGRATES" CLAUSE. THE CLASUE IS OPTIONAL.  
WHEN SPECIFIED, CONSISTS OF A KEY-CLAUSE-NAME

**ARGUMENTS**

-----  
MODEL-NO - DSPLY [S9(9)]  
RC-NO - DSPLY [S9(9)]  
IND-EC-NO - DSPLY [S9(9)]

**INCLUDE FILES**

-----  
LISTNOS - VALID LIST NUMBERS

**ROUTINES CALLED**

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERKC - VERIFY THE EXISTENCE OF A KEY CLASS IN A MODEL  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERRCC - verify if the relation class is complete.  
DRPMGR - DELETE MIGRATED KEY CLASS MEMBERS ASSOCIATED  
WITH A KEY CLASS  
DELCPRC - DELETE A RECORD FROM COMPLETE RELATION

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----  
**ALTREL - CONTROLS PROCESSING LOGIC FOR THE ALTER  
RELATION COMMAND**

**USED IN MAIN PROGRAM(S):**

-----  
**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**





**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DRPMOD  
**PURPOSE:** CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DRPMOD  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE CONTROLS THE PROCESSING LOGIC TO DROP A MODEL  
- IE ENTITIES, ATTRIBUTES, KEYWORDS, RELATIONS, VIEWS, ETC.

**ARGUMENTS:**  
-----

MODEL-NO = DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERMOD - verify the existence of a model and return the  
unique number.  
DELMOD - DELETE A RECORD FROM MODEL CLASS ENTITY  
DELTEXT - DELETE A RECORD FROM DESC TEXT ENTITY  
FNDECM - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR

PS 620141100  
1 November 1985

A GIVEN MODEL

FNDACH - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
          LINKED  
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

**CALLED DIRECTLY BY:**

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DRPRCE  
**PURPOSE:** DROP A RELATION CLASS FOR AN ENTITY BEING DROPPED  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DRPRCE  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

DRPRCE WILL DROP A RELATION CLASS BASED ON AN ENTITY CLASS NUMBER THAT IS TO BE DELETED.

ADDITIONALLY THIS ROUTINE WILL DELETE ALL TEXTUAL DESCRIPTIONS ASSOCIATED WITH THE RELATION AND ADD THE RELATION-CLASS NUMBER BACK INTO THE CDM FOR REUSE.

**ARGUMENTS:**  
-----

EC-NO = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

DELRC - DELETE A RECORD FROM RELATION CLASS  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO LINKED

PS 620141100  
1 November 1985

DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENITY  
DELRCWK - DELETE A RECORD FROM RC\_KEYWORD  
ERRPRO

CALLED DIRECTLY BY:

-----  
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING  
ENTITIES.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DRPREC  
**PURPOSE:** CONTROLS THE PROCESSING LOGIC FOR THE  
DROP RECORD COMMAND.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** PROGRAM  
**SOURCE FILE:** DRPREC  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
CONTROLS THE PROCESSING LOGIC FOR THE  
DROP RECORD COMMAND.

\*\*\*\*\* SPECIAL NOTICE \*\*\*\*\*

SPECIAL NOTE: THIS ROUTINE IS SET UP TO HANDLE  
RECORD, DATAFIELD, AND SET NUMBERS IN THE CDM

IT WILL ALSO DELETE TEXTUAL DESCRIPTIONS IN  
THE DESC TEXT TABLE AND ADD REUSABLE OBJECT  
NUMBERS TO THE CDM REUSABLE NUMBER TABLE.

THIS IS RELEASED UNDER VERSION 1.6

**INCLUDE FILES:**

-----  
LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

**ROUTINES CALLED:**

-----  
GETDRT - FETCH DATA BASE, RECORD TYPE INFO FROM PARSER  
LISTS  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DELRTY2 - DELETE A RECORD FROM RECORD\_TYPE ENITY  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO

LINKED

DELTEXT - DELETE A RECORD FROM DESC TEXT ENITY  
DELDA2 - DELETE A RECORD FROM DB AREA ASSIGNMENT ENITY  
DELDFL2 - CONTROLS THE DELETING OF DATA FIELDS  
PDFREC - RETRIEVES ALL PORJECT DATA FIELDS FOR A GIVEN  
RECORD  
DELRY2 - DELETE A RECORD FROM RECORD KEY ENITY  
DELRYM2 - DELETE A RECORD FROM RECORD KEY MEMBER ENITY  
DELISS2 - DELETE A RECORD FROM IMS SEGMENT SIZE ENITY  
DELSDF2 - DELETE A RECORD FROM SEGMENT DATA FIELD ENITY  
DELDSL2 - DELETE A RECORD FROM DF SET LINKAGE ENITY  
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET  
BEING DROPPED  
SELSTM - DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE  
BEING DROPPED

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** DRPREL  
**PURPOSE:** CONTROLS THE PROCESSING LOGIC FOR THE  
"DROP RELATION" COMMA  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** DRPREL  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
DRPREL CONTROLS THE PROCESSING LOGIC FOR THE  
"DROP RELATION" COMMAND.

**ARGUMENTS:**

-----  
MODEL-NO = DSPLY [S9(9)]

**INCLUDE FILES:**

-----  
LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

**ROUTINES CALLED:**

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
VERREL - VERIFY THE DEPENDENT AND INDEPENDENT ENTITIES  
IN THE RELATION  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERRCC - verify if the relation class is complete.  
DRPMGRC - DELETE MIGRATED KEY CLASS MEMBERS ASSOCIATED  
WITH A KEY CLASS  
DELCPRC - DELETE A RECORD FROM COMPLETE\_RELATION

PS 620141100  
1 November 1985

DELRC - DELETE A RECORD FROM RELATION\_CLASS  
DELRCWK - DELETE A RECORD FROM RC\_KEYWORD  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENTITY

CALLLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPSET  
PURPOSE: CONTROLS THE PROCESSING LOGIC FOR  
DELETING A SET FROM THE D  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: DRPSET  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR DELETING A SET FROM THE DATABASE.

THIS ROUTINE FETCHES INFORMATION REGARDING THE DATA BASE AND THE SET TO BE DROPPED. AFTER VERIFYING THAT THE SET DOES EXIST, IT IS DELETED FROM THE DATA BASE.

INCLUDE FILES:  
-----

UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERDBAS - VERIFY A DATA BASE FOR A GIVEN NAME AND HOST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERRSET - VERIFIES THE EXISTANCE OF A SET.  
DELRST2 - DELETE A RECORD FROM RECORD\_SET ENTITY  
DELSTM2 - DELETE A RECORD FROM SET\_TYPE\_MEMBER ENTITY  
DELASM2 - DELETE A RECORD FROM AUC\_ST\_MAPPING ENITY

PS 620141100  
1 November 1985

DELRR2 - DELETE A RECORD FROM RC\_BASED\_REC\_SET ENTITY  
DLDSL2 - DELETE A RECORD FROM DF\_SET\_LINKAGE ENTITY  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
LINKED  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENTITY  
ERRPRO

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPSMAP  
PURPOSE: DROP A SINGLE MAPPING  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: DRPSMAP  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

```
C          -- DRPSMAP (MAP_TYPE, &TAG_NO, &RC_NO)

COBOL     -- CALL  "DRPSMAP"      USING
                                     MAP_TYPE
                                     TAG_NO
                                     RC_NO.

FORTRAN   -- CALL  DRPSMAP (MAPTYPE, TAGNO, RCNO)
```

INPUT:

```
CHAR MAP_TYPE[]
INT  *TAG_NO
INT  *RC_NO
```

OUTPUT:

DESCRIPTION

DETERMINE WHETHER MAP TO BE DROPPED IS PRIMARY OR  
SECONDARY.  
IF PRIMARY, ENSURE THAT THERE ARE NO OTHER MAPPINGS.  
DEPENDING ON THE MAP TYPE, CALL THE APPROPRIATE DELETE  
ROUTINE.

ARGUMENTS:

-----  
MAP\_TYPE = CHAR []  
TAG\_NO = INT \*  
RC\_NO = INT \*

INCLUDE FILES:

-----  
STDYTP - STANDARD TYPE DEFINITIONS  
LISTID - PROVIDES LIST OF PARSED OBJECTS

ROUTINES CALLED:

-----  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
CPFCOR - ACCESS A TOKEN IN A CORRESPONDING NAMED LIST  
FNDRCH - RETRIEVES ALL RC/RT MAPPINGS FOR A NAMED RECORD  
VERPDF - verify the existence of a project\_data\_field  
occurrence.  
VERDB - VERIFY THE EXISTENCE OF A DATA BASE IN THE CDM  
FNDASH - DETERMINES IF AN AUC TO SET TYPE MAPPING EXISTS  
TOWER  
GETMAPC - SELECT ALL PROJECT DATA FIELDS MATCHING A GIVEN  
TAG NO  
STRNCMP  
SPRINTF  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DELIRCS - DELETE RECORD CLASS FROM RC BASED\_REC\_SET  
DELIASM - DELETE RECORD CLASS FROM AUC\_ST\_MAPPING  
DELIPDF - DELETE RECORD CLASS FROM PROJECT\_DATA\_FIELD

CALLED DIRECTLY BY:

-----  
ALTMAP - ALTER MAP COMMAND PROCESSOR

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPVIEW  
PURPOSE: DROP THE VIEW.  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: DRPVIEW  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
EXTRACT THE VIEW NAME, AND DROP THE CORRESPONDING  
DATA ITEM, PROJECT DATA ITEM, SEC ANDSEC-RC-  
COMPONENT.

SELECT THE FIRST VIEW-NAME AND VERIFY \*  
ASSOCIATED DATA ITEM, PROJECT DATA ITEM,  
SEC AND SEC-RC-COMPONENT. NOW ADD THE  
UNIQUE VIEW NUMBER. REPEAT THE PROCESS  
FOR THE OTHER VIEWS TO BE DROPPED.

INCLUDE FILES:

-----  
LISTNOS - VALID LIST NUMBERS  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERVIEW - verify the existence of a view in the system.  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DELPDI - DELETE RECORD CLASS FROM PROJECT DATA ITEM  
DELSECR - DELETE RECORD CLASS FROM SEC\_RC COMPONENT  
DRPDIV - DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED  
WITH VIEW  
DELDIV - DELETE FROM DATA ITEM WHERE  
DELSEC - DELETE RECORD CLASS FROM SEC  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO

PS 620141100  
1 November 1985

LINKED

DELTEXT - DELETE A RECORD FROM DESC TEXT ENITY  
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: DTUSAGE  
PURPOSE: DETERMINE ASSOCIATIONS OF A DATA TYPE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: DTUSAGE  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
DETERMINE IF THE DATA TYPE BEING DROPPED ARE  
ASSOCIATED WITH ANY DATA FIELD, DATA ITEMS OR  
ATTRIBUTE CLASS.

VERIFY IF DATA FIELD, DATA ITEM AND  
ATTRIBUTE CLASS IS ASSOCIATED WITH THE  
DATA TYPE.

ARGUMENTS:

-----  
DOM-NO = DSPLY [S9(9)]  
DATA-TYPE-NAME = DSPLY [X(30)]  
RETURN-STATUS = DSPLY [S9(9)]

ROUTINES CALLED:

-----  
VERDFDT - DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH  
A PROJECT DF  
VERDIDT - DETERMINE IF A DATA TYPE IS ASSOCIATED WITH ANY  
DATA ITEMS  
VERACDT - DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH  
ATTRIBUTE CLASSES

CALLED DIRECTLY BY:

-----  
DRPDT - DROP A DATA TYPE WITHIN A DOMAIN.

PS 620141100  
1 November 1985

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** ENTKW  
**PURPOSE:** SEARCH FOR ENTITY KEYWORD MATCHES WITHIN  
TWO MODELS  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** ENTKW  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** GDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE CONTROLS THE PROCESSING LOGIC TO POPULATE A  
KEYWORD TABLE WITH ENTITY CLASS KEYWORDS OF ONE MODEL, THEN  
SEARCHES THIS TABLE FOR MATCHES FROM THE SECOND MODEL.

**ARGUMENTS:**  
-----

MOD-NO1 = DSPLY [S9(9)]  
MOD-NO2 = DSPLY [S9(9)]  
MOD-NAME1 = DSPLY [X(30)]  
MOD-NAME2 = DSPLY [X(30)]

**INCLUDE FILES:**  
-----

KWDTBL - KEYWORD TABLE  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

RETECKW - POPULATES KEYWORD TABLE FOR ENTITY CLASS  
KEYWORDS  
RECKW2 - COMPARES ENTITY CLASS KEYWORDS BETWEEN MODELS  
ERRPRO

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**  
-----

**CMPMOD - CONTROLS THE PROCESSING LOGIC TO COMPARE TWO  
MODELS.**

**USED IN MAIN PROGRAM(S):**  
-----

**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**

NDDL COMMAND PROCESSOR Module Documentation

NAME: ERRRPT  
PURPOSE: HANDLE ANY ERROR CODE FROM ORACLE,  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: ERRORS  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- ERRRPT( RETCODE, MMSG, MOD\_NAME)

COBOL --  
CALL "ERRRPT" USING

RET-CODE,

MMSG-DESC,

MODULE-NAME.

FORTRAN -- CALL ERRRPT(RETCOD, MMSG, MODNAM)

INPUT:

RETCODE -- THE ORACLE ERROR CODE.

MMSG A CHARACTER STRING CONTAINING THE ERROR  
MESSAGE FOR THE ERROR LOG.

MOD\_NAME -- THE NAME OF THE MODULE EXPERIENCING

THE ORACLE ERROR.

OUTPUT:

DESCRIPTION

COMBINE THE RETURN CODE, THE MESSAGE, AND THE  
MODULE NAME INTO A SINGLE MESSAGE AND  
USE "ERRPRO" TO PUT IT INTO THE ERROR LOG.  
THEN CALL THE ORACLE ROUTINE OERMSG TO  
TO GET THE EXPANDED ERROR MESSAGE.  
AND PLACE THE ORACLE ERROR MESSAGE ON THE LOG.

ARGUMENTS:

-----  
RETCODE - INT  
MESG - CHAR \*  
MOD\_NAME - CHAR \*

INCLUDE FILES:

-----  
NDDL - \*\*\*\* PURPOSE NOT FOUND BY STRIPPER \*\*\*\*  
CMDS - COMMAND NO. FOR EACH NDDL/NDML COMMAND

ROUTINES CALLED:

-----  
SPRINTF  
ERRPRO  
COERMSG - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
ALLATT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND  
GENERATE  
ALLENT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND  
GENERATE  
ALLREL - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL  
GENERATE  
ALTSMAP - ALTER A SINGLE MAP  
BLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN UP THE  
HIERARCHY  
BLRCKC - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL FIND



BLRCKC1	- FOR EACH LEVEL OF RELATIONS IN STRUCTURE
CHKLOOP	- CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS
COMMIT	- STORE THE REUSEABLE NUMBER TO THE DATA BASE.
DEL1PDF	- DELETE A RECORD FROM PROJECT_DATA_FIELD ENTITY
DELAC	- DELETE A RECORD FROM ATTRIBUTE_CLASS
DELACAL	- DELETE RECORD CLASS FROM ATTRIBUTE_NAME
DELACKW	- DELETE A RECORD FROM AC_KEYWORD
DELACNM	- DELETE RECORD CLASS FROM ATTRIBUTE_NAME WHERE AC_NO
DELASM	- DELETE FROM ATTRIBUTE USE CLASS MAPPING
DELASM1	- DELETE A RECORD FROM AUC_ST_MAPPING ENTITY
DELASM2	- DELETE A RECORD FROM AUC_ST_MAPPING ENTITY
DELAUCL	- DELETE A RECORD FROM ATTRIBUTE_USE_CL
DELCMPR	- DELETE A RECORD FROM COMPLETE_RELATION
DELCPRC	- DELETE A RECORD FROM COMPLETE_RELATION
DELDA1	- DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDA2	- DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDBA1	- DELETE A RECORD FROM DATA_BASE_AREA ENTITY
DELDBS1	- DELETE A RECORD FROM DATA_BASE ENTITY
DELDFL1	- DELETE A RECORD FROM DATA_FIELD ENTITY
DELDFL2	- CONTROLS THE DELETING OF DATA FIELDS
DELDFL3	- DELETE A RECORD FROM DATA_FIELD ENTITY
DELDIV	- DELETE FROM DATA_ITEM WHERE
DELDOM	- DELETE RECORD CLASS FROM DOMAIN CLASS
DELDSL1	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELDSL2	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELDSL3	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELDT	- DELETE RECORD CLASS FROM USER_DEF_DATA_TYPE
DELDTD	- DELETE RECORD CLASS FROM USER_DEF_DATA-TYPE
DELEC	- DELETE THE RECORD FROM ENTITY_CLASS
DELECAL	- DELETE RECORD CLASS FROM ENTITY_NAME
DELECKW	- DELETE A RECORD FROM EC_KEYWORD
DELECNM	- DELETE THE RECORD FROM ENTITY_NAME
DELIASM	- DELETE RECORD CLASS FROM AUC_ST_MAPPING
DELIAUC	- DELETE A REORD FROM INHERITED_ATT_USE
DELIAUK	- DELETE A RECORD FROM INHERITED_ATT_USE ENTITY
DELIPDF	- DELETE RECORD CLASS FROM PROJECT_DATA_FIELD
DELIRCS	- DELETE RECORD CLASS FROM RC_BASED_REC_SET
DELISS1	- DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
DELISS2	- DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
DELKC	- DELETE A RECORD FROM KEY_CLASS
DELKCM	- DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKCMT	- DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKW	- DELETE A RECORD FROM AC_KEYWORD
DELKWAC	- DELETE A RECORD FROM AC_KEYWORD
DELKWEC	- DELETE A RECORD FROM EC_KEYWORD
DELKWRC	- DELETE A RECORD FROM RC_KEYWORD

DELMIGK - DELETE MIGRATING KEY CLASS  
DELMOD - DELETE A RECORD FROM MODEL\_CLASS ENTITY  
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER  
DELOACE - DELETE A RECORD FROM OWNED\_ATTRIBUTE ENTITY  
DELOWAC - DELETE A RECORD FROM OWNED\_ATTRIBUTE  
DELPGB - DELETE A RECORD FROM PSB\_PCB ENTITY  
DELPDFT - DELETE A RECORD FROM PROJECT\_DATA\_FIELD ENTITY  
DELPDI - DELETE RECORD CLASS FROM PROJECT\_DATA\_ITEM  
DELRBR1 - DELETE A RECORD FROM RC\_BASED\_REC\_SET ENTITY  
DELRBR2 - DELETE A RECORD FROM RC\_BASED\_REC\_SET ENTITY  
DELRBR3 - DELETE A RECORD FROM RC\_BASED\_REC\_SET ENTITY  
DELRC - DELETE A RECORD FROM RELATION\_CLASS  
DELRCWK - DELETE A RECORD FROM RC\_KEYWORD  
DELRCST - DELETE RECORD CLASS FROM RC\_BASED\_REC\_SET  
DELREUS -  
DELRKM1 - DELETE A RECORD FROM RECORD\_KEY\_MEMBER ENTITY  
DELRKM2 - DELETE A RECORD FROM RECORD\_KEY\_MEMBER ENTITY  
DELRKM3 - DELETE A RECORD FROM RECORD\_KEY\_MEMBER ENTITY  
DELRKY1 - DELETE A RECORD FROM RECORD\_KEY ENTITY  
DELRKY2 - DELETE A RECORD FROM RECORD\_KEY ENTITY  
DELRST2 - DELETE A RECORD FROM RECORD\_SET ENTITY  
DELRST3 - DELETE A RECORD FROM RECORD\_SET ENTITY  
DELRTY2 - DELETE A RECORD FROM RECORD\_TYPE ENTITY  
DELSDF1 - DELETE A RECORD FROM SEGMENT\_DATA\_FIELD ENTITY  
DELSDF2 - DELETE A RECORD FROM SEGMENT\_DATA\_FIELD ENTITY  
DELSDF3 - DELETE A RECORD FROM SEGMENT\_DATA\_FIELD ENTITY  
DELSEC - DELETE RECORD CLASS FROM SEC  
DELSECR - DELETE RECORD CLASS FROM SEC\_RC COMPONENT  
DELSN1 - DELETE A RECORD FROM SCHEMA\_NAMES ENTITY  
DELSTM1 - DELETE A RECORD FROM SET\_TYPE\_MEMBER ENTITY  
DELSTM2 - DELETE A RECORD FROM SET\_TYPE\_MEMBER ENTITY  
DELSTM3 - DELETE A RECORD FROM SET\_TYPE\_MEMBER ENTITY  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENTITY  
DELTXT - DELETE DESCRIPTION TEXT GIVEN THE OBJECT TYPE,  
DEPATT - SELECT ALL THE ATTRIBUTES IN THE  
DEPENT - SELECT ALL THE DEPENDANT ENTITY CLASSES  
DEPREL - FOR EACH LEVEL OF RELATIONS IN STRUCTURE  
GENERATE  
DLDSL2 - DELETE A RECORD FROM DF\_SET\_LINKAGE ENTITY  
DLMIGRC - DELETE MIGRATING KEY CLASS  
DPKCLST - CREATE AN KEY\_CLASS\_LIST TABLE CONTAINING ALL  
THE ENTITY  
DRPDF - DELETE A RECORD FROM DATA\_FIELD ENTITY  
GETNXNO -  
INITSES - PERFORM ANY SESSION INITIALIZATION NECESSARY  
INSAC - INSERT A RECORD INTO ATTRIBUTE\_CLASS  
INSACNM - INSERT A RECORD INTO ATTRIBUTE\_NAME

INSAREA - INSERT A RECORD INTO THE DATA\_BASE AREA ENTITY.  
IF

INSAUC - INSERT A RECORD INTO ATTRIBUTE\_USE\_CL  
INSAUCS - INSERT A RECORD INTO AUC\_ST\_MAPPING  
INSCRC - INSERT A RECORD INTO COMPLETE\_RELATION  
INSDAA - INSERT A RECORD INTO THE DB\_AREA\_ASSIGNMENT  
ENTITY. IF

INSDB - INSERT A RECORD INTO THE DATA\_BASE ENTITY. IF  
INSDFLD - INSERT A RECORD INTO THE DATA\_FIELD ENTITY.  
INSDI - INSERT A RECORD INTO DATA\_ITEM  
INSDOM - INSERT A RECORD INTO DOMAIN\_CLASS  
INSDSL - INSERT A RECORD INTO THE DF\_SET\_LINKAGE ENTITY.  
IF

INSDT - INSERT A RECORD INTO USER\_DEF\_DATA\_TYPE  
INSEC - INSERT A RECORD INTO ENTITY\_CLASS  
INSECNM - INSERT A RECORD INTO ENTIYT\_NAME  
INSIAUC - INSERT A RECORD INTO INHERITED\_ATT\_USE  
INSISS - INSERT A RECORD INTO THE IMS\_SEGMENT\_SIZE  
ENTITY. IF

INSKC - INSERT A RECORD INTO KEY\_CLASS  
INSKCM - INSERT A RECORD INTO KEY\_CLASS\_MEMBER  
INSKW - INSERT A RECORD INTO KEYWORD  
INSKWAC - INSERT A RECORD INTO AC\_KEYWORD  
INSKWEC - INSERT A RECORD INTO EC\_KEYWORD  
INSKWRC - INSERT A RECORD INTO RC\_KEYWORD  
INSMOD - INSERT A RECORD INTO MODEL\_CLASS  
INSOAC - INSERT A RECORD INTO OWNED\_ATTRIBUTE  
INSPCB - INSERT A RECORD INTO THE PSB\_PCB ENTITY. IF  
INSPDF - INSERT A RECORD INTO PROJECT\_DATA\_FIELD  
INSPDI - INSERT A RECORD INTO PROJECT\_DATA\_ITEM  
INSPSB - INSERT A RECORD INTO THE PSB ENTITY. IF  
INSPWRD - INSERT A RECORD INTO THE DB\_PASSWORD. IF  
SUCCESSFUL.

INSRC - INSERT A RECORD INTO RELATION\_CLASS  
INSRCRS - INSERT A RECORD INTO RC\_BASED\_REC\_SET  
INSREUS - INSERT A RECORD INTO REUSABLE\_NUMBER  
INSRKEY - INSERT A RECORD INTO THE RECORD\_KEY ENTITY. IF  
INSRKM - INSERT A RECORD INTO THE RECORD\_KEY\_MEMBER  
ENTITY. IF

INSRSET - INSERT A RECORD INTO THE RECORD\_SET ENTITY. IF  
INSRTYP - INSERT A RECORD INTO THE RECORD\_TYPE ENTITY. IF  
INSSCH - INSERT A RECORD INTO THE SCHEMA\_NAMES ENTITY.  
IF

INSSDFL - INSERT A RECORD INTO THE SEGMENT\_DATA\_FIELD  
ENTITY. IF

INSSEC - INSERT A RECORD INTO SEC  
INSSECR - INSERT A RECORD INTO SEC\_RC\_COMPONENT

INSSTM - INSERT A RECORD INTO THE SET\_TYPE\_MEMBER  
ENTITY. IF  
MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING  
MODEL\_2  
MRGNODE - SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT  
ENTITY'S  
NRGET -  
RDDESC - STORE DESCRIPTION ON THE CDM  
ROLBACK - ROLBACK THE TRANSACTIONS.  
SELIAUC - SELECTS ALL THE INHERITED TAG NAMES  
TERMSES - ROUTINE TO TERMINATE AN NDDL  
TLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN DOWN THE  
HIERARCHY  
TOPNODE - SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT  
ENTITY  
UPDAC - UPDATE ATTRIBUTE\_CLASS SET DOMAIN\_NO = : 1  
UPDACAL - UPDATE MODEL\_CLASS SET AC\_NAME\_TYPE = : 1  
UPDACNM - UPDATE ATTRIBUTE\_NAME SET AC\_NAME = : 1  
UPDECAL - UPDATE MODEL\_CLASS SET EC\_NAME\_TYPE = : 1  
UPDECNM - UPDATE MODEL\_CLASS SET ENTITY\_NAME = : 1  
UPDIND - UPDATE USER\_DEF\_DATA\_TYPE SET DATA\_TYPE\_IND = :  
1  
UPDMNAM - UPDATE MODEL\_CLASS SET MODEL\_NAME = : 1  
UPDMOD - UPDATE MODEL\_CLASS SET MODEL\_STATUS = : 1  
UPDNXNO -  
UPDRCNM - UPDATE RELATION\_CLASS SET RC\_NAME = : 1  
UPDTDOM - UPDATE DOMAIN SET DOMAIN\_NAME = : 1  
UPDTDI - UPDATE USER\_DEF\_DATA\_TYPE SET TYPE\_ID = : 1  
UPDTKW - UPDATE KEYWORD SET KEYWORD = : 1  
UPDTRC - UPDATE RELATION\_CLASS SET NO\_IND\_ENT = : 1.  
UPDVIEW - UPDATE VIEW SET SEC\_ID = : 1  
WRTDESC - SELECT A RECORD FROM DESC\_TEXT ENITY  
WRTDSC4 - SELECT A RECORD FROM DESC\_TEXT ENITY

USED IN MAIN PROGRAM(S):

-----  
DELDFL1 - DELETE A RECORD FROM DATA\_FIELD ENITY  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** EXCFLAG  
**PURPOSE:** DETERMINE IF KEYWORDS, ALIASES, DESCR ARE  
TO BE EXCLUDED  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** EXCFLAG  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

GETS THE ITEMS TO BE EXCLUDED (IF ANY) FROM THE PARSER LIST  
AND SETS THE KEYWORD, ALIAS AND DESC FLAP ACCORDINGLY. A  
FLAG VALUE OF ZERO INDICATES KEYWORDS, ALIASES, AND/OR  
DESCRIPTIONS ARE TO BE INCLUDED IN GENERATED NDDL COMMANDS.  
A VALUE OF 1 INDICATE THEY ARE TO BE EXCLUDED FROM THE  
GENERATED NDDL.

. EACH KEYWORD IS PLACED ON A SEPERATE  
LIST BY THE PARSER.

**ARGUMENTS:**  
-----

KEYWRD-FLAG = DSPLY [9]  
ALIAS-FLAG = DSPLY [9]  
DESC-FLAG = DSPLY [9]

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS

**ROUTINES CALLED:**  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST

-----  
CALLED DIRECTLY BY:

CMBENT - CONROLS THE PROCESSING LOGIC FOR THE COMBINE  
ENTITY COMMAND.  
COPATT - CONTROLS THE PROCESSING LOGIC FOR COPYING AN  
ATTRIBUTE.  
COPENT - CONTROLS THE PROCESSING LOGIC FOR THE COPY  
ENTITY COMMAND.  
CPYMOD - CONTROLS THE PROCESSING LOGIC FOR THE COPY  
MODEL COMMAND.  
MRGMOD - MERGE TWO IDEF MODELS INTO ONE

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: EXPRCLT  
PURPOSE: EXPAND RELATION LIST TO INCLUDE DEP AND  
IND ENTITIES  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: EXPRCLT  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
THIS ROUTINE CONTAINS THE PROCESSING LOGIC TO  
EXPAND THE RELATION CLASS LIST.

For each entry in the  
relation class list, populate the dependent  
or independent entity class name.

ARGUMENTS:

-----  
VIEW-RELATION-CLASS-LIST = RECRD  
VIEW-FROM-LIST = RECRD

INCLUDE FILES:

-----  
VWRC - LIST OF RELATION CLASSES INHERENT TO A VIEW  
VWFROM - LIST OF ENTITIES SPECIFIED IN A VIEW

CALLED DIRECTLY BY:

-----  
REVIEW - POPULATE VIEW DEFINITION AND ES-CS-MAPPING

USED IN MAIN PROGRAM(S):

PS 620141100  
1 November 1985

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



PS 620141100  
1 November 1985

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** EXPRTLT  
**PURPOSE:** EXPAND THE RETRIEVE LIST (SELECT..FROM..)  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** EXPRTLT  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** GDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

**ARGUMENTS:**  
-----

VIEW-RETRIEVE-LIST = RECRD  
VIEW-FROM-LIST = RECRD  
RETURN-STATUS = DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

VWRETR - LIST OF ENTITIES AND TAGS SPECIFIED IN A VIEW  
VWFROM - LIST OF ENTITIES SPECIFIED IN A VIEW

**ROUTINES CALLED:**  
-----

UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

**CALLED DIRECTLY BY:**  
-----

PMFROM - CREATE A VIEW FROM MULTIPLE ENTITY CLASSES

**USED IN MAIN PROGRAM(S):**  
-----

PS 620141100  
1 November 1985

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FCOPATT  
PURPOSE: GENERATE NDDL COMMANDS FROM A COPY  
ATTRIBUTE COMMAND  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FCOPATT  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE IS CALLED BY COPY ATTRIBUTE TO GENERATE NDDL  
COMMANDS ON A FILE.

ARGUMENTS:  
-----

CUR-MODEL-NO = DSPLY [S9(9)]  
FILE-NAME = DSPLY [X(30)]  
MOD-NAME = DSPLY [X(30)]  
AC-NAME1 = DSPLY [X(30)]  
AC-NAME2 = DSPLY [X(30)]  
DOM-NAME = DSPLY [X(30)]  
AC-NO1 = DSPLY [S9(9)]  
KEYWORD-FLAG = DSPLY [9]  
ALIAS-FLAG = DSPLY [9]  
DESC-FLAG = DSPLY [9]

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A  
MODEL  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
OPNFIL  
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
GENATT - GENERATE A CREATE ATTRIBUTE.. COMMAND  
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN  
OBJECT TYPE AND NO  
GENALI - GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT  
TYPE  
CLSFIL - THIS ROUTINE CLOSES AN OUTPUT FILE. THE FILE  
WILL  
ERRPRO

-----  
CALLED DIRECTLY BY:

COPATT - CONTROLS THE PROCESSING LOGIC FOR COPYING AN  
ATTRIBUTE.

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100  
1 November 1985

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** FCOPENT  
**PURPOSE:** DETERMINE IF COPY ENTITY WITH STRUCTURE  
OR RELATION  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** FCOPENT  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR THE COPY  
ENTITY COMMAND WHERE THE NDDL STATEMENTS ARE GENERATED ON A  
FILE.

**ARGUMENTS:**  
-----

WITH-TYPE = DSPLY [X(9)]  
FROM-MODEL-NO = DSPLY [S9(9)]  
FROM-MODEL-NAME = DSPLY [X(30)]  
FROM-EC-NO = DSPLY [S9(9)]  
FROM-EC-NAME = DSPLY [X(30)]  
TO-EC-NAME = DSPLY [X(30)]  
KW-FLAG = DSPLY [9]  
ALIAS-FLAG = DSPLY [9]  
DESC-FLAG = DSPLY [9]  
FILE-NAME = DSPLY [X(30)]  
CUR-MODEL-NO = DSPLY [S9(9)]  
CUR-MODEL-NAME = DSPLY [X(30)]

**INCLUDE FILES:**  
-----

ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**

-----  
CESTRUC - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES FOR THE STRUCT  
CERELS - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES IN A RELATION.  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

**CALLED DIRECTLY BY:**

-----  
COPENT - CONTROLS THE PROCESSING LOGIC FOR THE COPY  
ENTITY COMMAND.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FILEINS  
PURPOSE: INSERT DESCRIPTION TEXT INTO CDM  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FILEINS  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
EXT INTO CDM

DESCRIPTION : FILEINS READS DESCRIPTION TEXT FROM A USER-SPECIFIED FILE AND INSERTS THE TEXT INTO THE CDM. PREVIOUS DESCRIPTION TEXT, IF ANY, IS DELETED. IF THE FILE DOESN'T EXIST OR IF IT IS EMPTY, PRE-EXISTING TEXT IS NOT DELETED AND AN ERROR MESSAGE IS GENERATED.

FILEINS READS TEXT FROM A FILE, BINDS IN A TEXT LINE NUMBER AND THE TEXT ITSELF AND INSERTS THE OBJECT TYPE, OBJECT NUMBER, DESCRIPTION TYPE, TEXT LINE NUMBER AND TEXT INTO THE DATABASE.

F IT EXISTS.

EXT.

ARGUMENTS:

-----  
LDA = RECRD  
TEXTCUR = RECRD  
FILE-NAME = DSPLY [X(30)]  
OBJ-TYPE = DSPLY [X(30)]  
DESC-TYPE = DSPLY [X(30)]  
OBJ-NO = DSPLY [S9(9)]

PS 620141100  
1 November 1985

LINE-COUNT - DSPLY [S9(9)]

INCLUDE FILES:

-----  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
DELTXT - DELETE DESCRIPTION TEXT GIVEN THE OBJECT TYPE,  
OBINDN  
OEXEC  
ERRPRO

CALLED DIRECTLY BY:

-----  
RDDESC - STORE DESCRIPTION ON THE CDM

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: FINDDOM  
PURPOSE: RETRIEVE A DOMAIN NUMBER FOR A GIVEN  
DOMAIN NAME  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FINDDOM  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
-  
Given an attribute class number, this routine will  
return its domain number.

ARGUMENTS:

-----  
AC-NO = DSPLY [S9(9)]  
DOM-NO = DSPLY [S9(9)]

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
ERRPRO

CALLED DIRECTLY BY:

PS 620141100  
1 November 1985

-----  
ALTSMAP - ALTER A SINGLE MAP  
MAPADF - MAP ON AUC TO A DATA FIELD

USED IN MAIN PROGRAM(S):  
-----

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FMTIAUC  
PURPOSE: FORMATS A LINE FOR THE ..SET.. CLAUSE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FMTIAUC  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

formats a line for the SET clause

ARGUMENTS:

-----

IAUC-TAG-NAME = DSPLY [X(30)]  
OWNER-TAG-NAME = DSPLY [X(30)]

ROUTINES CALLED:

-----

ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER

CALLED DIRECTLY BY:

-----

SELIAUC - SELECTS ALL THE INHERITED TAG NAMES

USED IN MAIN PROGRAM(S):

-----

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FND1MEM  
PURPOSE: RETRIEVE RECORD TYPE OF MEMBER ASSOCIATED  
WITH A SET  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FND1MEM  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
FND1MEM VERIFIES THAT THE DATABASE NAME, SET-NAME AND MEMBER  
RECORD NAME IS VALID. IN THE CASE THAT THE OPTIONAL MEMBER  
RECORD NAME HAS NOT BEEN ENTERED, THIS ROUTINE VERIFIES THAT  
THIS SET HAS ONLY 1 MEMBER RECORD NAME WHICH IT RETURNS.

AN NDML SELECT STATEMENT RETRIEVES ALL OCCURRENCE OF THE  
RT ID OF MEMBER WHOSE DB ID AND SET ID MATCH THE INPUT  
ARGUMENTS. IF THE USER ENTERED A MEMBER RECORD NAME, IT IS  
VALIDATED. IF THE USER DID NOT ENTER A MEMBER RECORD NAME,  
A CHECK IS MADE FOR A SINGLE SET TYPE MEMBER, AND, IF SO,  
THE SINGLE SET TYPE MEMBER NAME IS RETURNED.

ARGUMENTS:

-----  
DB-'O = DSPLY [S9(9)]  
SET-NAME = DSPLY [X(30)]  
MEMBER-RECORD-NAME = RECRD

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

PS 620141100  
1 November 1985

-----  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

-----  
CALLED DIRECTLY BY:

-----  
MAPRC - MAP A RELATION CLASS TO A SET

-----  
USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FNDACH  
PURPOSE: DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN  
MODEL  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FNDACH  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE DELETES THE ATTRIBUTE CLASS FROM THE MODEL.

DELETE ATTRIBUTE, KEYWORDS, VIEWS, ETC, FROM  
MODEL, GIVEN THE AC-NO. THEN RETURN THIS NUMBER TO THE  
REUSABLE POOL.

ARGUMENTS:  
-----

MODEL-NO = DSPLY [S9(9)]  
ERROR-STATUS = DSPLY [S9(9)]

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

DELAC - DELETE A RECORD FROM ATTRIBUTE\_CLASS  
DELACKW - DELETE A RECORD FROM AC\_KEYWORD  
DELTEXT - DELETE A RECORD FROM DESC\_TEXT ENTITY

PS 620141100  
1 November 1985

DELACNM - DELETE RECORD CLASS FROM ATTRIBUTE\_NAME WHERE  
          AC\_NO  
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO  
          LINKED  
SELACNM - RETRIEVE THE PRIMARY NAME FOR ATTRIBUTE CLASS  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

-----  
CALLED DIRECTLY BY:

DRPHOD - CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FNDASA  
PURPOSE: VERIFY WHETHER A SET HAS BEEN MAPPED TO  
AN AUC  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FNDASA  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----  
-

This routine will verify that a set has or has not  
been mapped to an attribute use class before.

ARGUMENTS:  
-----

DB-ID = DSPLY [S9(9)]  
SET-ID = DSPLY [X(30)]  
FOUND = DSPLY [S9(9)]

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

ERRPRO



PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----

MAPASET - MAP AN AUC TO A SET  
MAPRC - MAP A RELATION CLASS TO A SET

**USED IN MAIN PROGRAM(S):**

-----

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FNDASM  
PURPOSE: DETERMINES IF AN AUC TO SET TYPE MAPPING  
EXISTS  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FNDASM  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
- THIS ROUTINE PERFORMS AN NDML SEARCH OF THE  
AUC ST MAPPING TABLE TO DETERMINE WHETHER A PARTICULAR  
MAPPING EXISTS. IF SO, FOUND = 1; OTHERWISE ZERO.

ARGUMENTS:

-----  
DB-NO = DSPLY [S9(9)]  
SET-NAME = DSPLY [X(30)]  
TAG-NO = DSPLY [S9(9)]  
FOUND = DSPLY [S9(9)]

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
ERRPRO

CALLED DIRECTLY BY:

-----  
ALTSMAP - ALTER A SINGLE MAP  
DRPSMAP - DROP A SINGLE MAPPING

PS 620141100  
1 November 1985

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FNDauc  
PURPOSE: DELETE ALL ATTRIBUTE USE CLASSES FOR A  
GIVEN ENTITY CLASS  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FNDauc  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
FNDauc WILL RETRIEVE ALL ATTRIBUTE USED CLASSES  
FOR AN ENTITY AND DELETE EACH ONE.

ARGUMENTS:

-----  
EC-NO = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
DELAUC - DELETE INHERITED ATT USE, KEY CLASS MEMBER,  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

CALLED DIRECTLY BY:

PS 620141100  
1 November 1985

-----  
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING  
ENTITIES.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** FNDECM  
**PURPOSE:** DELETES ALL ENTITIES AND ASSOCIATED  
OBJECTS FOR A GIVEN MOD  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** FNDECM  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE DELETES THE ENTITY CLASS FROM THE MODEL TO BE  
DROPPED, ALONG WITH THE ATTRIBUTE USE CLASSES, KEY CLASSES  
AND RELATION CLASSES.

**ARGUMENTS:**  
-----

MODEL-NO = DSPLY [S9(9)]  
ERROR-STATUS = DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

DELOACE - DELETE A RECORD FROM OWNED ATTRIBUTE ENTITY  
DELEC - DELETE THE RECORD FROM ENTITY CLASS  
DELECKW - DELETE A RECORD FROM EC KEYWORD  
DELTEXT - DELETE A RECORD FROM DESC TEXT ENTITY  
DLMDAUC - DELETE ATTRIBUTE USE CLASSES ASSOCIATED WITH

PS 620141100  
1 November 1985

ENTITY

DELMDKC	- DELETE ALL KEY CLASSES AND INHERITED KEYS FOR AN ENTITY
DELMDRC	- DELETE ALL RELATION CLASSES FOR AN ENTITY
DELECNM	- DELETE THE RECORD FROM ENTITY NAME
ADDRNUM	- ADD A AVAILABLE NO OF A POOL NO BACK TO NO LINKED
SELECNM	- RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS
UERROR	- ISSUE A MESSAGE TO THE USER, CONSIDERED A
ERRPRO	

-----  
CALLED DIRECTLY BY:

DRPMOD - CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FND0AC  
PURPOSE: DELETES ALL OWNED ATTRIBUTES FOR A GIVEN  
ENTITY CLASS  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FND0AC  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
FND0AC WILL RETRIEVE ALL OWNED ATTRIBUTES  
FOR AN ENTITY CLASS AND DELETE THEM AS OWNED  
ATTRIBUTES OF THE ENTITY CLASS.

ARGUMENTS:

-----  
EC-NO = DSPLY [S9(9)]  
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
DELOAC - DELETE ALL OWNED AND INHERITED ATTRIBUTES  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

CALLED DIRECTLY BY:



PS 620141100  
1 November 1985

DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING  
ENTITIES.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: FNDRCM  
PURPOSE: RETRIEVES ALL RC/RT MAPPINGS FOR A NAMED RECORD  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: FNDRCM  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
- FNDRCM RETURNS ALL ROWS OF RC\_BASED\_REC\_SET WITH MATCHING DB NUMBERS, SET NAMES AND RELATION CLASS NUMBERS.  
IF THE USER ENTERED A RECORD NAME IN THE COMMAND, A CHECK IS MADE TO DETERMINE IF IT IS VALID. IF NO RECORD NAME WAS ENTERED, AND IF MORE THAN 1 ROW IS RETURNED, AN ERROR CONDITION EXISTS. ALSO, AN ERROR CONDITION EXISTS IF NO ROWS ARE RETURNED.

ARGUMENTS:

-----  
DB-NO = DSPLY [S9(9)]  
SET-NAME = DSPLY [X(30)]  
RC-NO = DSPLY [S9(9)]  
REC-NAME = RECRD  
ERROR-FLAG = DSPLY [S9(9)]

INCLUDE FILES:

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
ERRPRO

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----  
DRPSMAP - DROP A SINGLE MAPPING

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** FRTOREL  
**PURPOSE:** DETERMINE IF RELATION EXISTS BETWEEN  
INTRA-MOD ENTITIES  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** FRTOREL  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE LOOKS FOR RELATIONS BETWEEN THE FROM-ENTITY  
AND TO-ENTITY ON AN INTRA MODEL COMBINE. IF FOUND, GENERATE  
A "DROP RELATION".

**ARGUMENTS:**  
-----

FROM-EC-NO = DSPLY [S9(9)]  
FROM-EC-NAME = DSPLY [X(30)]  
TO-EC-NO = DSPLY [S9(9)]  
TO-EC-NAME = DSPLY [X(30)]  
RCNO-LIST = RECRD

**INCLUDE FILES:**  
-----

LISTREL - LIST OF RELATIONS  
RELTBL - LIST OF RELATION CLASSES IN A MODEL  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

SELRCNM - RETRIEVE ALL RELATION CLASSES FOR WHICH AN  
ENTITY IS PART OF

PS 620141100  
1 November 1985

UWARN           - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ADDPARM       - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
                  CHECKS/DELIMITER  
ERRPRO

**CALLED DIRECTLY BY:**

-----  
CMBENT       - CONROLS THE PROCESSING LOGIC FOR THE COMBINE  
                  ENTITY COMMAND.

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN   - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GENAKW  
PURPOSE: RETRIEVE ALL KEYWORDS FOR AN ATTRIBUTE  
CLASS  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GENAKW  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

SELECTS ALL THE KEYWORDS FOR A GIVEN ATTRIBUTE AND GENERATES  
THE KEYWORD PHRASE FOR THE CREATE ATTRIBUTE COMMAND.

ARGUMENTS:  
-----

AC-NO = DSPLY [S9(9)]

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
ERRPRO

CALLED DIRECTLY BY:  
-----

GENATT - GENERATE A CREATE ATTRIBUTE.. COMMAND

PS 620141100  
1 November 1985

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GENALI  
PURPOSE: GENERATE A CREATE ALIAS COMMAND FOR AN  
OBJECT TYPE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GENALI  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

GENERATES A CREATE ALIAS GIVEN THE OBJECT TYPE ("ENTITY" OR  
"ATTRIBUTE") AND THE OBJECT'S PRIMARY NAME AND ALIAS NAME.

ARGUMENTS:

-----

OBJ-TYPE = DSPLY [X(30)]  
OBJ-NAME = DSPLY [X(30)]  
ALIAS-NAME = DSPLY [X(30)]

ROUTINES CALLED:

-----

ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER

CALLED DIRECTLY BY:

-----

CMBACAL - GENERATE CREATE ALIAS ATTRIBUTE.. AND ALIAS  
DESC TEXT COMMANDS  
CMBALI - GENERATE CREATE ALIAS ENTITY..COMMAND  
FCOPATT - GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE  
COMMAND  
ALLATT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND



PS 620141100  
1 November 1985

                  GENERATE  
ALLENT          - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND  
                  GENERATE  
DEPATT          - SELECT ALL THE ATTRIBUTES IN THE  
DEPENT          - SELECT ALL THE DEPENDANT ENTITY CLASSES

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN      - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GENALT1  
**PURPOSE:** GENERATE AN ALTER ENTITY..ADD KEY CLASS  
COMMAND  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GENALT1  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

Generates an ALTER ENTITY command to add a key  
class to an entity, given the entity's ec-no and ec-name.

**ARGUMENTS:**  
-----

EC-NO = DSPLY [S9(9)]  
EC-NAME = DSPLY [X(30)]  
KC-LIST = RECRD

**INCLUDE FILES:**  
-----

KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

**ROUTINES CALLED:**  
-----

NEXTKC - THIS ROUTINE RETURNS A KC\_NAME FOR A GIVEN  
EC\_NO FROM  
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
NEXTKCM - THIS ROUTINE RETURNS A KG\_NO,KG\_NAME FOR A  
GIVEN EC\_NO

**CALLED DIRECTLY BY:**

PS 620141100  
1 November 1985

-----  
ALLKEY - GENERATES KEY CLASS FOR AN ENTITY

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GENALTE  
PURPOSE: GENERATE AN ALTER ENTITY..ADD KEY CLASS  
COMMAND  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GENALTE  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
Generates an ALTER ENTITY command to add a key  
class to an entity, given the entity's ec-no and ec-name.

ARGUMENTS:

-----  
EC-NO = DSPLY [S9(9)]  
EC-NAME = DSPLY [X(30)]

ROUTINES CALLED:

-----  
NEXTKC - THIS ROUTINE RETURNS A KC\_NAME FOR A GIVEN  
EC\_NO FROM  
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
NEXTKCM - THIS ROUTINE RETURNS A KG\_NO,KG\_NAME FOR A  
GIVEN EC\_NO

CALLED DIRECTLY BY:

-----  
CERELS - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES IN A RELATION.  
CESTRUC - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES FOR THE STRUCT  
CMBENT - CONROLS THE PROCESSING LOGIC FOR THE COMBINE

PS 620141100  
1 November 1985

ENTITY COMMAND.  
MRGNODE - SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT  
ENTITY'S  
TOPNODE - SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT  
ENTITY

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GENATT  
PURPOSE: GENERATE A CREATE ATTRIBUTE.. COMMAND  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GENATT  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

GENERATES A CREATE ATTRIBUTE COMMAND.

ARGUMENTS:  
-----

AC-NO = DSPLY [S9(9)]  
AC-NAME = DSPLY [X(30)]  
DOM-NAME = DSPLY [X(30)]  
KEYWRD-FLAG = DSPLY [9]

ROUTINES CALLED:  
-----

ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
GENAKW - RETRIEVE ALL KEYWORDS FOR AN ATTRIBUTE CLASS

CALLED DIRECTLY BY:  
-----

CMBOA - GENERATE COMMANDS FOR ATTRIBUTES, ITS  
KEYWORDS, ALIAS, DESC  
FCOPATT - GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE  
COMMAND  
ALLATT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND  
GENERATE

PS 620141100  
1 November 1985

DEPATT - SELECT ALL THE ATTRIBUTES IN THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100  
1 November 1985



**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GENDESC  
**PURPOSE:** GENERATED NDDL DESCRIBE COMMANDS FOR A  
GIVEN OBJECT TYPE AND  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GENDESC  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

GENERATES A DESCRIBE COMMAND WITH EMBEDDED TEXT DESCRIPTION  
FOR EACH DESC-TYPE IN THE CDM FOR A GIVEN OBJECT-TYPE AND  
OBJECT-NO. OBJECT-TYPES USED BY THIS ROUTINE ARE 'ENTITY',  
'ATTRIBUTE' AND 'RELATION'.

**ARGUMENTS:**  
-----

OBJ-TYPE = DSPLY [X(30)]  
OBJ-NO = DSPLY [S9(9)]  
OBJ-NAME1 = DSPLY [X(30)]  
OBJ-NAME2 = DSPLY [X(30)]  
OBJ-NAME3 = DSPLY [X(30)]

**INCLUDE FILES:**  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

ADDTXT - THIS ROUTINE WRITES 80 CHARACTERS OF  
DESCRIPTIVE TEXT

ADDPARN - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
ERRPRO

-----  
CALLED DIRECTLY BY:

CMBOA - GENERATE COMMANDS FOR ATTRIBUTES, ITS  
KEYWORDS, ALIAS, DESC  
DEPFROM - GENERATE CREATE RELATION, DESCRIBE COMMANDS IN  
THE TO-MODEL  
FCOPATT - GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE  
COMMAND  
GENENT1 - GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND  
DESCRS.  
INDFROM - RETRIEVES RELATIONS, DETERMINES IND EC AND  
GENERATES NDDL  
ALLATT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND  
GENERATE  
ALLENT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND  
GENERATE  
ALLREL - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL  
GENERATE  
DEPATT - SELECT ALL THE ATTRIBUTES IN THE  
DEPENT - SELECT ALL THE DEPENDANT ENTITY CLASSES  
DEPREL - FOR EACH LEVEL OF RELATIONS IN STRUCTURE  
GENERATE  
NRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING  
MODEL\_2

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GENEKW  
**PURPOSE:** SELECT KEYWORD FOR ENTITY AND CREATE  
KEYWORD PHRASE FOR CRT  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GENEKW  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

SELECTS ALL THE KEYWORDS FOR A GIVEN ENTITY AND GENERATES  
THE KEYWORD PHRASE FOR THE CREATE ENTITY COMMAND.

**ARGUMENTS:**  
-----

EC-NO - DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
ERRPRO

**CALLED DIRECTLY BY:**  
-----

GENENT - GENERATE A CREATE ENTITY OWNED

PS 620141100  
1 November 1985

ATTRIBUTE..KEYWORD COMMAND

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GENENT  
PURPOSE: GENERATE A CREATE ENTITY..OWNED  
ATTRIBUTE..KEYWORD COMMAND  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GENENT  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

GENERATES A CREATE ENTITY COMMAND GIVEN THE ENTITY CLASS  
NO. AND NAME.

ARGUMENTS:

-----

EC-NO = DSPLY [S9(9)]  
EC-NAME = DSPLY [X(30)]  
KEYWRD-FLAG = DSPLY [9]

ROUTINES CALLED:

-----

ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
GENOA - SELECT OWNED ATT FOR ENTITY AND CREATES OWNED  
ATT FOR CRT ENT  
GENEKW - SELECT KEYWORD FOR ENTITY AND CREATE KEYWORD  
PHRASE FOR CRT ENT

CALLED DIRECTLY BY:

-----

ALLENT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND  
GENERATE

PS 620141100  
1 November 1985

DEPENT - SELECT ALL THE DEPENDANT ENTITY CLASSES

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GENENT1  
**PURPOSE:** GENERATE CREATE/ALTER ENTITY  
COMMAND, ALIAS, AND DESCRS.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GENENT1  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE GENERATES A CREATE ATTRIBUTE COMMAND FOR ALL OWNED ATTRIBUTES OF THE FROM-ENTITY ALONG WITH ITS KEYWORDS ALIASES AND DESCRIPTIONS FOR THE TO-ENTITY, IF THE ATTRIBUTE DOES NOT EXIST IN THE TO-MODEL.

**ARGUMENTS:**  
-----

FROM-MOD-NO = DSPLY [S9(9)]  
TO-MOD-NO = DSPLY [S9(9)]  
FROM-EC-NO = DSPLY [S9(9)]  
TO-EC-NO = DSPLY [9(9)]  
TO-EC-NAME = DSPLY [X(30)]  
KW-FLAG = DSPLY [9]  
ALIAS-FLAG = DSPLY [9]  
DESC-FLAG = DSPLY [9]  
COMBINE-TYPE = DSPLY [9]  
CREATE-FLAG = DSPLY [9]  
RETURN-STATUS = DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
ADPARM1 - CREATES 80 CAHR NDDL COMMANDS WITH PARAMETERS  
AND DELIMITERS  
CMBOA - GENERATE COMMANDS FOR ATTRIBUTES, ITS  
KEYWORDS, ALIAS, DESC  
CMBEKW - GENERATE ADD KEYWORD CLAUSE FOR ENTITY KEYWORDS  
CMBALI - GENERATE CREATE ALIAS ENTITY..COMMAND  
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN  
OBJECT TYPE AND NO  
ERRPRO

CALLED DIRECTLY BY:

-----  
CERELS - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES IN A RELATION.  
CESTRUC - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES FOR THE STRUCT  
CMBENT - CONROLS THE PROCESSING LOGIC FOR THE COMBINE  
ENTITY COMMAND.  
MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING  
MODEL 2  
MRGNODE - SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT  
ENTITY'S

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: GENOA  
PURPOSE: SELECT OWNED ATT FOR ENTITY AND CREATES  
OWNED ATT FOR CRT E  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GENOA  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

SELECTS ALL THE OWNED ATTRIBUTES FOR A GIVEN ENTITY AND  
GENERATES THE OWNED ATTRIBUTE CLAUSE FOR A CREATE ENTITY  
COMMAND.

ARGUMENTS:  
-----

EC-NO = DSPLY [S9(9)]

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
ERRPRO

CALLED DIRECTLY BY:  
-----

PS 620141100  
1 November 1985

GENENT - GENERATE A CREATE ENTITY..OWNED  
ATTRIBUTE..KEYWORD COMMAND

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GENREL  
PURPOSE: GENERATE CREATE RELATION  
          ..MIGRATES..KEYWORD COMMAND  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GENREL  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
GENERATES A CREATE RELATION COMMAND GIVEN THE RELATION CLASS  
JOIN. IF THE RELATION IN THE FROM-MODEL IS COMPLETE,  
GENERATES THE MIGRATES CLAUSE. IF KEYWORDS WERE NOT EXCLUDED  
BY THE USER, GENERATES A KEYWORD CLAUSE.

ARGUMENTS:

-----  
MODEL-NO = DSPLY [S9(9)]  
IND-EC-NO = DSPLY [S9(9)]  
IND-EC-NAME = DSPLY [X(30)]  
DEP-EC-NO = DSPLY [S9(9)]  
DEP-EC-NAME = DSPLY [X(30)]  
RC-NO = DSPLY [S9(9)]  
RC-NAME = DSPLY [X(30)]  
IND-ENT = DSPLY [S9(9)]  
MIN-DEP-ENT = DSPLY [S9(9)]  
MAX-DEP-ENT = DSPLY [S9(9)]  
KEYWRD-FLAG = DSPLY [9]

ROUTINES CALLED:

-----  
GENRME - FORMAT THE CREATE RELATION CLAUSE  
MIGREL - GENERATE A MIGRATES CLAUSE FOR A CREATE

PS 620141100  
1 November 1985

RELATION COMMAND  
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
GENRKW - SELECT KEYWORDS FOR RELATION CREATES KW PHRASE  
FOR CREATE RC

-----  
CALLED DIRECTLY BY:

ALLREL - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL  
GENERATE  
DEPREL - FOR EACH LEVEL OF RELATIONS IN STRUCTURE  
GENERATE

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GENRKW  
**PURPOSE:** SELECT KEYWORDS FOR RELATION CREATES KW  
PHRASE FOR CREATE R  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GENRKW  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

SELECTS ALL THE KEYWORDS FOR A GIVEN RELATION AND GENERATES  
THE KEYWORD PHRASE FOR THE CREATE RELATION COMMAND.

**ARGUMENTS:**  
-----

RC-NO = DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

ADDPARM WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS DELIMITER  
ERRPRO

**CALLED DIRECTLY BY:**  
-----

GENREL GENERATE CREATE RELATION MIGRATES KEYWORD

PS 620141100  
1 November 1985

**COMMAND**

**USED IN MAIN PROGRAM(S):**

-----  
**NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR**

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GENRNME  
**PURPOSE:** FORMAT THE CREATE RELATION CLAUSE  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GENRNME  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
Formats the CREATE RELATION clause.

**ARGUMENTS:**

-----  
IND-ENT = DSPLY [S9(9)]  
IND-EC-NAME = DSPLY [X(30)]  
RC-NAME = DSPLY [X(30)]  
MIN-DEP-ENT = DSPLY [S9(9)]  
MAX-DEP-ENT = DSPLY [S9(9)]  
DEP-EC-NAME = DSPLY [X(30)]

**ROUTINES CALLED:**

-----  
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER

**CALLED DIRECTLY BY:**

-----  
DEPFROM - GENERATE CREATE RELATION, DESCRIBE COMMANDS IN  
THE TO-MODEL  
GENREL - GENERATE CREATE RELATION ..MIGRATES..KEYWORD  
COMMAND  
INDFROM - RETRIEVES RELATIONS, DETERMINES IND EC AND  
GENERATES NDDL  
MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING

PS 620141100  
1 November 1965

MODEL 2

USED IN MAIN PROGRAM(S)

-----  
DDOL/MAIN - MAIN PROGRAM FOR THE DDOL COMMAND PROCESSOR



**DDDL COMMAND PROCESSOR Module Documentation**

<b>NAME</b>	<b>GETACAL</b>
<b>PURPOSE</b>	<b>VERIFY THE EXISTANCE OF AN ALIAS FOR AN ATTRIBUTE</b>
<b>LANGUAGE</b>	<b>VAX 11 COBOL</b>
<b>MODULE TYPE</b>	<b>SUBROUTINE</b>
<b>SOURCE FILE</b>	<b>GETACAL</b>
<b>SOURCE FILE TYPE</b>	<b>PRC</b>
<b>HOST</b>	
<b>SUBSYSTEM</b>	<b>CDM</b>
<b>SUBDIRECTORY</b>	
<b>DOCUMENTATION GROUP</b>	<b>PS41100</b>

**DESCRIPTION**

**VERIFY THE EXISTANCE OF AN ALIAS NAME FOR AN ATTRIBUTE CLASS**

**SELECT THE ATTRIBUTE QUALIFYING ON THE UNIQUE ATTRIBUTE NUMBER. ALIAS NAME TO BE**

**VERIFIED AND NAME TYPE BEING "ALIAS"**

**ARGUMENTS**

**AC NO - DPLY (89(9))**  
**ALIAS ID - DPLY (X(30))**

**INCLUDE FILES**

**SRVRET AS THE RETURN GIVEN A TABLE-FULL ERROR**  
**ERRPRO PROCESS ERROR INCLUDE FILE**

**ROUTINES CALLED**

**DELACAL DELETE RECORD CLASS FROM ATTRIBUTE NAME**  
**UERROR ISSUE A MESSAGE TO THE USER. CONSIDERED A**  
**ERRPRO**

**CALLED DIRECTLY BY**

PS 620141100  
1 November 1985

-----  
DRPALI - DROP THE ALIAS FOR AN ENTITY OR ATTRIBUTE.

USED IN MAIN PROGRAM(S):

-----  
MDDL/MAIN - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR

PS 620141100  
1 November 1985

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GETDBST  
**PURPOSE:** RETURN INFORMATION ABOUT THE CURRENT  
SESSIONS' DATABASE  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GETDBST  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE RETURNS WITH INFORMATION ABOUT THE DATA BASE,  
WHICH IS OBTAINED EITHER FROM THE PARSER LISTS OR FROM  
INFORMATION WHICH IS GLOBALLY AVAILABLE.

EXTRACT THE DATA BASE NAME AND HOST NAME FROM THE PARSER  
LIST. THEN, AFTER VERIFYING THAT THE DB DOES EXIST, FETCH  
THE DBMS NAME. IF NOTHING WAS RETRIEVED FROM THE PARSER  
LIST, THE CURRENT DATA BASE IS ASSUMED. NEXT, EXTRACT THE  
SET ID OR MEMBER/OWNER ID DEPENDING ON WHETHER IT IS A SET  
OR PATH.

**ARGUMENTS:**  
-----

DB-ID = DSPLY [S9(9)]  
SET-ID = DSPLY [X(30)]  
DB-NAME = DSPLY [X(30)]  
HOST-NAME = DSPLY [X(30)]  
DBMS-NAME = DSPLY [X(30)]  
RETURN-STATUS = DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**

-----  
CPPONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERDEAS - VERIFY A DATA BASE FOR A GIVEN NAME AND HOST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
CHGLOBL - CHANGES THE GLOBAL DB PARAMETERS  
GETGLOB - WILL PROVIDE GLOBAL VARIABLES  
CPPNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.  
ERRPRO

**CALLED DIRECTLY BY:**

-----  
DEFSET - CONTROLS THE PROCESSING LOGIC FOR THE DEFINE  
SET COMMAND.

**USED IN MAIN PROGRAM(S):**

-----  
MDDL/MAIN - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR

**WDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GETDOM  
**PURPOSE:** RETRIEVES DOMAIN NUMBER BASED ON TAG NAME  
FOR AUC  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GETDOM  
**SOURCE FILE TYPE:** PRC  
**HOST:**  
**SUBSYSTEM:** CDN  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION**

-----  
This routine will retrieve the domain number for a  
tag name assigned to an attribute use class

**ARGUMENTS**

-----  
EC-NO - DSPLY [S9(9)]  
TAG-NAME - DSPLY [X(30)]  
DOM-NO - DSPLY [S9(9)]  
TAG-NO - DSPLY [S9(9)]

**INCLUDE FILES:**

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**

-----  
ERRPRO

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----  
**CHKDONS** - VERIFY THAT THE DATA ITEM AND ATTRIBUTE USE  
CLASS.  
**ENVIEW** - CREATE ES-CS-MAPPING FOR USER SPECIFIED RETRIEVE  
LIST

**USED IN MAIN PROGRAM(S):**

-----  
**WDDL/MAIN** - MAIN PROGRAM FOR THE WDDL COMMAND PROCESSOR

PS 620141100  
1 November 1985

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GETDRT  
**PURPOSE:** FETCH DATA BASE, RECORD TYPE INFO FROM  
PARSER LISTS  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GETDRT  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

**ARGUMENTS:**  
-----

DB-ID - DSPLY [S9(9)]  
DBMS-NAME - DSPLY [X(30)]  
RT-ID - DSPLY [X(30)]  
RT-NO - DSPLY [S9(9)]  
RET-STATUS - DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS

**ROUTINES CALLED:**  
-----

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
VERDBAS - VERIFY A DATA BASE FOR A GIVEN NAME AND HOST  
VERRT - VERIFIES THE EXISTANCE OF A RECORD TYPE.

**CALLED DIRECTLY BY:**

PS 620141100  
1 November 1985

-----  
DRPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP  
RECORD COMMAND.

USED IN MAIN PROGRAM(S):

-----  
WDDL/MAIN - MAIN PROGRAM FOR THE WDDL COMMAND PROCESSOR



**MDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GETECAL  
**PURPOSE:** VERIFIES THE EXISTANCE OF AN ALIAS FOR AN ENTITY  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GETECAL  
**SOURCE FILE TYPE:** .PRC  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
VERIFY THE EXISTANCE OF ALIAS NAME FOR AN ENTITY

SELECT THE ENTITY CLASS NAME, QUALIFYING  
ON THE ENTITY NUMBER AND ALIAS NAME AS  
NAME TYPE "ALIAS".

**ARGUMENTS:**

-----  
EC-NO = DSPLY [S9(9)]  
ALIAS-ID = DSPLY [X(30)]

**INCLUDE FILES:**

-----  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**

-----  
DELECAL - DELETE RECORD CLASS FROM ENTITY\_NAME  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
ERRPRO

**CALLED DIRECTLY BY:**

-----  
DRPALI - DROP THE ALIAS FOR AN ENTITY OR ATTRIBUTE.

PS 620141100  
1 November 1985

USED IN MAIN PROGRAM(S):

-----  
WDDL/MAIN - MAIN PROGRAM FOR THE WDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME	GETECNM
PURPOSE	THIS ROUTINE SEARCHES THE UNBOUNDED EC LIST DATA
LANGUAGE	C
MODULE TYPE	FUNCTION
FUNCTION TYPE	INT ( )
SOURCE FILE	GETECNM
SOURCE FILE TYPE	C
HOST	
SUBSYSTEM	CDM
SUBDIRECTORY	
DOCUMENTATION GROUP	PS41100

DESCRIPTION

STRUCTURE FOR A MATCHING ROW GIVEN EC\_NO  
AND RETURNS  
EC\_NAME

SYNOPSIS

C	--	GETECNM(#EC_NO, #EC_NAME, #STATUS)
COBOL	-	CALL "GETECNM" USING EC-NO, EC-NAME, STATUS
FORTRAN	--	CALL GETECNM(ECNO, ECNAME, STATUS)

INPUT:

INT	*EC_NO :
CHAR	*EC_NAME :
INT	*STATUS :

OUTPUT:

DESCRIPTION:

ARGUMENTS:

-----  
EC\_NO = INT \*

PS 620141100  
1 November 1985

EC NAME -            CHAR \*  
STATUS -            INT \*

INCLUDE FILES:

-----  
KEYLIST        - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

ROUTINES CALLED:

-----  
STRMCPY

CALLED DIRECTLY BY:

-----  
ALLREL        - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL  
                    GENERATE  
TOPNODE       - SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT  
                    ENTITY

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN     - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

AD-A181 674

INTEGRATED INFORMATION SUPPORT SYSTEM (IIS) VOLUME 5

5/5

COMMON DATA MODEL 5 (U) GENERAL ELECTRIC CO

SCHENECTADY NY PRODUCTION RESOURCES CONSU

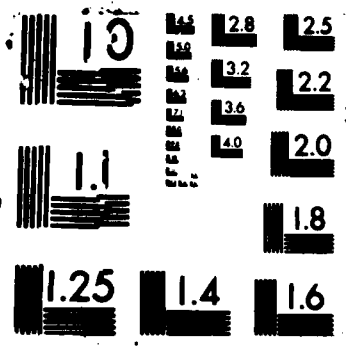
S SINGH ET AL 01 NOV 85 PS-628141100

F/G 9/2

NL

UNCLASSIFIED





NDDL COMMAND PROCESSOR Module Documentation

NAME: GETECS  
PURPOSE: USING ENTITY CLASS VERIFIES CHECK MODEL  
RULES  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GETECS  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

THIS ROUTINE USES THE ENTITY CLASSES TO HELP VERIFYING  
CERTAIN RULES BEFORE A MODEL CAN BE MARKED AS "CHECKED".

ARGUMENTS:

-----

ORACLE-LDA = RECRD  
MODEL-NO = DSPLY [S9(9)]  
NDDL-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ORCLEDA - WS DEFINITION FOR THE ORACLE LOGIN AREA  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----

CHKREL - CHECK IF RELATIONS HAVE BEEN CREATED ACCORDING  
TO STANDARDS  
CHKKEYS - CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.  
CHKATT - CHECK IF ATTRIBUTES HAVE BEEN CREATED ACCORDING  
TO STANDARDS

PS 620141100  
1 November 1985

**UWARN** - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
**ERRPRO**

**CALLED DIRECTLY BY:**  
-----

**CHKMOD** - DETERMINES WHETHER CERTAIN RULES ARE FULFILLED,

**USED IN MAIN PROGRAM(S):**  
-----

**NDDL/MAIN** - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: GETGLOB  
PURPOSE: WILL PROVIDE GLOBAL VARIABLES  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: GETGLOB  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----  
SYNOPSIS

C -- GETGLOB(DB\_ID, DB\_NAME, HOST\_NAME, DBMS\_NAME,  
STATUS)

COBOL -- CALL "GETGLOB" USING

DB\_ID,

DB\_NAME,

HOST\_NAME,

DBMS\_NAME,

STATUS.

FORTRAN -- CALL

GETGLOB(DB\_ID, DB\_NAME, HOST\_NAME, DBMS\_NAME, STATUS)

INPUT:

OUTPUT:

DB\_ID,  
DB\_NAME,  
HOST\_NAME  
DBMS\_NAME,  
STATUS.

DESCRIPTION

THIS ROUTINE WILL GET THE FOLLOWING 4 GLOBAL VARIABLE

1. CUR\_DBID (CURRENT DATABASE ID)
2. CUR\_DBNAME (CURRENT DATABASE NAME)
3. CUR\_HOST (CURRENT HOST NAME)
4. CUR\_DBMS (CURRENT DATABASE MANAGEMENT)

IF ALL 4 ABOVE GLOBAL VARIABLE ARE DEFINED, THEN  
THE RETURN STATUS IS 0, OTHERWISE -1

ARGUMENTS:

-----  
DB\_ID = INT \*  
DB\_NAME = CHAR \*  
HOST\_NAME = CHAR \*  
DBMS\_NAME = CHAR \*  
STATUS = INT \*

ROUTINES CALLED:

-----  
STRNCPY

CALLED DIRECTLY BY:

-----  
GETDBST - RETURN INFORMATION ABOUT THE CURRENT SESSIONS'  
DATABASE  
GETRDH - RETURN WITH CURRENT SESSIONS' DATA BASE INFO

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GETMAPC  
PURPOSE: SELECT ALL PROJECT DATA FIELDS MATCHING A  
GIVEN TAG NO  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GETMAPC  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

GETMAPC SELECTS ALL ROWS FROM PROJECT DATA FIELD WHOSE TAG NUMBERS MATCH THE INPUT ARGUMENT. THE NUMBER OF ROWS SELECTED IS RETURNED AS THE OUTPUT ARGUMENT.

AN NDML SELECT STATEMENT IS USED TO RETRIEVE ALL OCCURRENCES OF PROJECT DATA FIELD WHOSE TAG NO MATCHES THE INPUT ARGUMENT. THE NUMBER OF OCCURRENCES RETRIEVED IS RETURNED AS THE OUTPUT ARGUMENT.

ARGUMENTS:  
-----

TAG-NO = DSPLY [S9(9)]  
REC-COUNT = DSPLY [S9(9)]

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

ERRPRO

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----  
DRPSMAP - DROP A SINGLE MAPPING

**USED IN MAIN PROGRAM(S):**

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GETNCHR  
**PURPOSE:** GET THE NEXT CHARACTER FROM EITHER  
STANDARD INPUT  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** GETNCHR  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

OR FORMS COMMAND BUFFER DEPENDING ON MODE

**SYNOPSIS**

C	--	GETNCHR(%CHR,%FLAG)	
COBOL	--	CALL "GETNCHR"	USING CHR FLAG.
FORTRAN	--	CALL GETNCHR(CHR,FLAG)	

**INPUT:**

**OUTPUT:**

INT \*CHR  
INT \*FLAG

**DESCRIPTION**

GETNCHR GETS THE NEXT CHARACTER FROM THE INPUT SOURCE  
DEPENDING  
ON WHETHER IN BATCH MODE (STANDARD INPUT) OR INTERACTIVE  
OR UIMS  
MODES (FORMS COMMAND BUFFER). IF OK, FLAG = 0, IF

PS 620141100  
1 November 1985

NEWLINE  
CHARACTER, FLAG = 1, IF TAB CHARACTER, FLAG = 2 AND IF  
EOF,  
FLAG = 3. ALSO, IF EOF ENCOUNTERED, THE GLOBAL  
SKIP\_FLAG IS SET  
TO EOF.

ARGUMENTS:

-----  
CHR = INT \*  
FLAG = INT \*

INCLUDE FILES:

-----  
STDYTP - STANDARD TYPE DEFINITIONS  
STDIO - \*\*\*\* PURPOSE NOT FOUND BY STRIPPER \*\*\*\*  
NDDL - \*\*\*\* PURPOSE NOT FOUND BY STRIPPER \*\*\*\*

ROUTINES CALLED:

-----  
GETCHAR

CALLED DIRECTLY BY:

-----  
STRINS - INSERT DESCRIPTION TEXT INTO THE CDM

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GETNNUM  
**PURPOSE:**  
**LANGUAGE:** C  
**MODULE TYPE:** FUNCTION  
**FUNCTION TYPE:** INT ()  
**SOURCE FILE:** GETNNUM  
**SOURCE FILE TYPE:** .C  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

-----  
**SYNOPSIS**

C -- GETNNUM (POOL\_NO, NEXT\_NO)  
  
COBOL -- CALL "GETNNUM" USING  
          POOL-NO,  
          NEXT\_NO.  
  
FORTRAN -- CALL GETNNUM (POOLNO, NEXT\_NO)

**INPUT:**

INT \*POOL\_NO ;

**OUTPUT:**

INT \*NEXT\_NO ;  
NEXT\_NO WILL BE A ZERO IF ANY TROUBLES WERE FOUND

**DESCRIPTION**

THIS ROUTINE RETURNS A UNIQUE NUMBER FOR A POOL NUMBER

**ARGUMENTS:**

-----  
POOL\_NO = INT \*  
NEXT\_NO = INT \*

**ROUTINES CALLED:**

-----  
NRGET -

ERRPRO  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

-----  
CALLED DIRECTLY BY:

-----  
ADDKC - CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR  
CREATE/ALTER ENTITY  
ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD  
KEYWORD" (OPTIONAL)  
ADDMIG - PROCESS THE ADD MIGRATES...SET..CLAUSE  
ADDOAC - ADD ATTRIBUTE AS AN OWNED ATTRIBUTE AND AS ATT  
USE CLASS  
BLDATT - CREATE ATT CLASS AND ATT NAME FOR A  
MODEL(CREATE/COPY ATT)  
BLDATT1 - CREATES ATT CLASS AND ATT NAME FOR A  
MODEL(CREATE/COPY ATT)  
CRTDOM - PROGRAM NAME CRTDOM  
CRTENT - CONTROL THE PROCESSING LOGIC FOR CREATING A NEW  
ENTITY CLASS.  
CRTMOD - CONTROL THE PROCESSING LOGIC IF CREATING A  
MODEL WITHIN THE SYSTE  
CRTREL - CONTROLS THE LOGIC FOR VALIDATING AND CREATING  
A NEW RELATION CLA  
CRTVIEW - CONTROLS THE PROCESSING LOGIC FOR THE CREATE  
VIEW COMMAND.  
DEFDB - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
DATABASE TO THE SYST  
DEFSET - CONTROLS THE PROCESSING LOGIC FOR THE DEFINE  
SET COMMAND.  
ICOPENT - INTERACTIVE COPY ENTITY WITH  
ATTRIBUTES,KEYWORDS,ALIAS,DESC  
KEYLOOK - RETRIVES KEYCLASS NAME AND NUMBER BASED ON TAG  
NO  
INSDFLD - INSERT A RECORD INTO THE DATA\_FIELD ENTITY.  
INSDI - INSERT A RECORD INTO DATA\_ITEM  
INSDT - INSERT A RECORD INTO USER\_DEF\_DATA\_TYPE  
INSRTYP - INSERT A RECORD INTO THE RECORD\_TYPE ENTITY. IF

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: GETNXNO  
PURPOSE:  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: GETNXNO  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- GETNXNO (POOL\_NO, NEXT\_NO, STATUS)  
  
COBOL -- CALL "GETNXNO" USING  
          POOL-NO,  
          NEXT\_NO,  
          STATUS.  
  
FORTRAN -- CALL GETNXNO (POOLNO, NEXT\_NO, STATUS)

INPUT:

INT \*POOL\_NO ;

OUTPUT:

INT \*NEXT\_NO ;  
INT \*STATUS ;

DESCRIPTION

THIS ROUTINE GET A UNIQUE NUMBER OF A POOL NUMBER FROM  
NEXT\_NUMBER

RELATION

ARGUMENTS:

-----

POOL\_NO = INT \*  
NEXT\_NO = INT \*  
STATUS = INT \*

PS 620141100  
1 November 1985

ROUTINES CALLED:

-----  
COSQL3 - ORACLE ROUTINE  
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,  
UPDNXNO -  
COFETCH - ORACLE ROUTINE  
COEXEC - ORACLE ROUTINE  
COBINDN - ORACLE ROUTINE  
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

-----  
NRGET -

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GETRCID  
PURPOSE: QUERIES CDM FOR INFORMATION ABOUT A  
RELATION CLASS  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GETRCID  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE DOES AN NDML SELECT TO RETRIEVE INFORMATION  
ABOUT THE RELATION CLASS.

ARGUMENTS:  
-----

RC-NO = DSPLY [S9(9)]  
IND-EC-NO = DSPLY [S9(9)]  
RC-NAME = DSPLY [X(30)]  
DEP-EC-NO = DSPLY [S9(9)]

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

ERRPRO

CALLED DIRECTLY BY:  
-----

PS 620141100  
1 November 1985

GETRCNM - RETRIEVE RELTION CLASS NAME, IND AND DEP ENTITY  
NAMES

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GETRCNM  
PURPOSE: RETRIEVE RELTION CLASS NAME, IND AND DEP  
ENTITY NAMES  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: GETRCNM  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE RETRIEVES THE INDEPENDENT ENTITY CLASS NAME  
RELATION NAME AND DEPENDENT ENTITY CLASS NAME -

ARGUMENTS:  
-----

RC-NO = DSPLY [S9(9)]  
IND-EC-NAME = DSPLY [X(30)]  
RC-NAME = DSPLY [X(30)]  
DEP-EC-NAME = DSPLY [X(30)]

INCLUDE FILES:  
-----

ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:  
-----

GETRCID - QUERIES CDM FOR INFORMATION ABOUT A RELATION  
CLASS  
RETRECP - RETIEVE AN ENTITY CLASS NAME FOR A GIVEN NO.  
AND NAME TYPE  
ERRPRO

PS 620141100  
1 November 1985

**CALLED DIRECTLY BY:**

-----

RRCKW2 - COMPARES RELATION CLASS KEY WORDS FOR TWO MODELS

**USED IN MAIN PROGRAM(S):**

-----

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** GETRDH  
**PURPOSE:** RETURN WITH CURRENT SESSIONS' DATA BASE  
INFO  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** GETRDH  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE RETURNS WITH INFORMATION ABOUT THE DATA BASE WHICH IS OBTAINED EITHER FROM THE PARSER LISTS OR FROM INFORMATION WHICH IS GLOBALLY AVAILABLE.

EXTRACT THE DATA BASE NAME AND HOST NAME FROM THE PARSER LIST. IF NO INFORMATION IS RETRIEVED, THE CURRENT DATA BASE IS ASSUMED. ALSO, EXTRACT THE RECORD TYPE FROM THE PARSER LIST.

**ARGUMENTS:**  
-----

DB-ID = DSPLY [S9(9)]  
RT-ID = DSPLY [X(30)]  
DB-NAME = DSPLY [X(30)]  
HOST-NAME = DSPLY [X(30)]  
DBMS-NAME = DSPLY [X(30)]  
RETURN-STATUS = DSPLY [S9(9)]

**INCLUDE FILES:**  
-----

LISTNOS - VALID LIST NUMBERS  
ERRPRO - PROCESS ERROR INCLUDE FILE

PS 620141100  
1 November 1985

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
VERDBAS - VERIFY A DATA BASE FOR A GIVEN NAME AND HOST  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
CHGLOBL - CHANGES THE GLOBAL DB PARAMETERS  
GETGLOB - WILL PROVIDE GLOBAL VARIABLES  
ERRPRO

CALLED DIRECTLY BY:

-----  
DEFREC - CONTROLS THE PROCESSING LOGIC FOR DEFINING A  
RECORD FOR THE SYSTE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: HALT  
PURPOSE: HALT WITH 'COMMIT' OR 'ROLLBACK'.  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: HALT  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C -- HALT(&STATUS) ;  
COBOL -- CALL "HALT" USING  
STATUS.  
FORTRAN -- CALL HALT (STATUS)

INPUT:  
NONE

OUTPUT:  
INT \*STATUS ;

DESCRIPTION:

THIS ROUTINE PROCESSES THE 'HALT' COMMAND WITH 'COMMIT'  
OR 'ROLLBACK'  
IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND  
RETURN VALUE BOTH  
WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

-----

STATUS = INT \*

INCLUDE FILES:

PS 620141100  
1 November 1985

-----  
LISTID - PROVIDES LIST OF PARSED OBJECTS

ROUTINES CALLED:

-----  
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST  
STRNCMP  
ROLBACK - ROLBACK THE TRANSACTIONS.  
COMMIT - STORE THE REUSEABLE NUMBER TO THE DATA BASE.

CALLED DIRECTLY BY:

-----  
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** ICOPATT  
**PURPOSE:** INTERACTIVE COPY ATTRIBUTE, WITH  
KEYWORDS, ALIAS, DESCRS.  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** ICOPATT  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

THIS ROUTINE IS CALLED BY COPY ATTRIBUTE. IT COPIES AN  
ATTRIBUTE EITHER WITHIN THE MODEL OR TO ANOTHER MODEL. IT  
ALSO COPIES THE KEYWORDS, DESCRIPTIONS AND ALIASES IF THE  
COMMAND SPECIFIES THIS REQUIREMENT.

**ARGUMENTS:**  
-----

MOD-NO = DSPLY [S9(9)]  
AC-NAME1 = DSPLY [X(30)]  
AC-NAME2 = DSPLY [X(30)]  
DOM-NO = DSPLY [S9(9)]  
OLD-AC-NO = DSPLY [S9(9)]  
KEYWORD-FLAG = DSPLY [9]  
ALIAS-FLAG = DSPLY [9]  
DESC-FLAG = DSPLY [9]

**INCLUDE FILES:**  
-----

ERRPRO - PROCESS ERROR INCLUDE FILE

**ROUTINES CALLED:**  
-----

PS 620141100  
1 November 1985

BLDATT - CREATE ATT CLASS AND ATT NAME FOR A  
MODEL(CREATE/COPY ATT)  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
INSACNM - INSERT A RECORD INTO ATTRIBUTE\_NAME  
WRTALI - RETRIEVE ALL ALIAS NAMES OF AN ATTRIBUTE CLASS.  
WRTACKW - RETREIVE AND WRITE ALL THE KEYWORDS ASSOCIATD  
WITH AN ATTRIBUTE C  
WRTDESC - SELECT A RECORD FROM DESC\_TEXT ENITY  
ERRPRO

-----  
CALLED DIRECTLY BY:

COPATT - CONTROLS THE PROCESSING LOGIC FOR COPYING AN  
ATTRIBUTE.

-----  
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ICOPENT  
PURPOSE: INTERACTIVE COPY ENTITY WITH  
ATTRIBUTES, KEYWORDS, ALIAS, DESC  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: ICOPENT  
SOURCE FILE TYPE: .COB  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE COPIES AN ENTITY EITHER WITHIN A MODEL OR TO ANOTHER MODEL.

NOTE: 0 : INTRA MODEL COPY  
1 : INTER MODEL COPY  
0 : KEYWORDS, ALIAS, DESCRIPTIONS WANTED COPIED  
1 : ABOVE, EXCEPTED.

ARGUMENTS:  
-----

MODEL-NO = DSPLY [S9(9)]  
TO-EC-NAME = DSPLY [X(30)]  
INTER-INTRA = DSPLY [9]  
KW-FLAG = DSPLY [9]  
ALIAS-FLAG = DSPLY [9]  
DESC-FLAG = DSPLY [9]  
FROM-EC-NO = DSPLY [S9(9)]

INCLUDE FILES:  
-----

UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS  
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
GETNUM -  
INSEC - INSERT A RECORD INTO ENTITY CLASS  
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A  
INSECNM - INSERT A RECORD INTO ENTIYT NAME  
WRTECKW - RETREIVE AND COPY ALL THE KEYWORDS ASSOCIATED  
WITH AN ENTITY CLAS  
WRDESC - SELECT A RECORD FROM DESC TEXT ENITY  
WRTENAM - RETRIEVE AND COPY ALL THE ALIAS NAMES OF AN  
ENTITY CLASS.  
COPYAC - CREATE AN ATTRIBUTE, ASSOCIATE WITH ENTITY, ADD  
KEY CLASSES  
ERRPRO

CALLED DIRECTLY BY:

-----  
COPENT - CONTROLS THE PROCESSING LOGIC FOR THE COPY  
ENTITY COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: INDFROM  
PURPOSE: RETRIEVES RELATIONS, DETERMINES IND EC AND  
GENERATES NDDL  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: SUBROUTINE  
SOURCE FILE: INDFROM  
SOURCE FILE TYPE: .PRC  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:  
-----

THIS ROUTINE SELECTS ALL THE RELATIONS IN WHICH THE  
FROM-ENTITY IS AN INDEPENDENT ENTITY. IF THE SAME ENTITIES  
EXIST IN THE TO-MODEL, BUT NOT THE RELATION, THEN CREATE  
RELATION AND DESCRIBE COMMANDS ARE GENERATED FOR THE  
FROM-RELATION.

ARGUMENTS:  
-----

FROM-EC-NO = DSPLY [S9(9)]  
TO-EC-NO = DSPLY [S9(9)]  
FROM-EC-NAME = DSPLY [X(30)]  
TO-EC-NAME = DSPLY [X(30)]  
FROM-MODEL-NO = DSPLY [S9(9)]  
TO-MOD-NO = DSPLY [S9(9)]  
KW-FLAG = DSPLY [9]  
DESC-FLAG = DSPLY [9]  
COMBINE-TYPE = DSPLY [9]  
RCNO-LIST = RECRD

INCLUDE FILES:  
-----

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR  
RELTBL - LIST OF RELATION CLASSES IN A MODEL

ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

-----  
SELECNM - RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS  
VERRC - verify the existence of a relation class.  
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A  
MODEL  
VERRCNM - VERIFY EXISTANCE OF A RELATION IN A MODEL  
GENRNME - FORMAT THE CREATE RELATION CLAUSE  
MIGREL - GENERATE A MIGRATES CLAUSE FOR A CREATE  
RELATION COMMAND  
CMBRKW - SELECT AND GENERATE RELATION CLASS KEYWORDS  
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS  
CHECKS/DELIMITER  
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN  
OBJECT TYPE AND NO  
ERRPRO

CALLED DIRECTLY BY:

-----  
CERELS - GENERATES NDDL COMMANDS ON A FILE FOR ALL  
ENTITIES IN A RELATION.  
CMBENT - CONROLS THE PROCESSING LOGIC FOR THE COMBINE  
ENTITY COMMAND.

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



NDDL COMMAND PROCESSOR Module Documentation

NAME: INITCMD  
PURPOSE: INITIALIZATION FOR EACH COMMAND TO BE PROCESSED  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: INT ()  
SOURCE FILE: CPF  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: CDM  
SUBDIRECTORY:  
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-----

SYNOPSIS

C --

INITCMD()

COBOL -- CALL "INITCMD"

FORTRAN

CALL INITCMD

INPUT:

OUTPUT:

DESCRIPTION

THIS WILL PERFORM A LOOP INITIALIZING THE DATA STRUCTURES  
IN WHICH THE PARSER WILL STORE COMMAND TOKENS.

INCLUDE FILES:

-----

LISTS - PROVIDES THE DIMENSIONS OF THE NDDL LISTS  
NDDL - \*\*\*\* PURPOSE NOT FOUND BY STRIPPER \*\*\*\*

CALLED DIRECTLY BY:

PS 620141100  
1 November 1985

-----  
PRCCMD - THIS ROUTINE IS CALLED TO HANDLE ALL

USED IN MAIN PROGRAM(S):

-----  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

**NDDL COMMAND PROCESSOR Module Documentation**

**NAME:** INITRDL  
**PURPOSE:** INITIALISE THE TABLE WHICH STORES A  
MODELS RELATIONS  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** SUBROUTINE  
**SOURCE FILE:** INITRDL  
**SOURCE FILE TYPE:** .COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:**  
**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**  
-----

INITRDL initializes the RC-DEPKC table.  
The table contains information for every relation in a model  
in sequence by level, rc-no, and dep-ec-no.

**ARGUMENTS:**  
-----

RC-DEPKC-LIST = RECRD

**INCLUDE FILES:**  
-----

RCDEPKC - LIST OF KEYS MIGRATED VIA A RELATION

**CALLED DIRECTLY BY:**  
-----

BLRCKC - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL FIND  
BLRCKC1 - FOR EACH LEVEL OF RELATIONS IN STRUCTURE

**USED IN MAIN PROGRAM(S):**  
-----

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

END

7-87

DTIC