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

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

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
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

DATE

5 Aug 1976

FOR THE COMMANDER.

Gerald C. Shumaker, Branch Chief
AFWAL/MLTC
Wright Patterson AFB OH 45433

DATE

7 Aug 86

"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
This document is the product specification establishing the design implementation of the IESS Configuration Item Neutral Data Definition Language (NDL) which is the primary tool used for maintaining the Common Data Model (CDM) data base.

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

A SD 86 1482
17 Jul 1986
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:

<table>
<thead>
<tr>
<th>TASK 4.2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subcontractors</strong></td>
</tr>
<tr>
<td>Boeing Military Aircraft Company (BMAC)</td>
</tr>
<tr>
<td>D. Appleton Company (DACOM)</td>
</tr>
<tr>
<td>General Dynamics/ Ft. Worth</td>
</tr>
</tbody>
</table>
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 IDEF0 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.
## Subcontractors

<table>
<thead>
<tr>
<th>Subcontractors</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Equipment Corporation (DEC)</td>
<td>Consulting and support of the performance testing and on DEC software and computer systems operation.</td>
</tr>
<tr>
<td>McDonnell Douglas Automation Company (McAuto)</td>
<td>Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.</td>
</tr>
<tr>
<td>On-Line Software International (OSI)</td>
<td>Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.</td>
</tr>
<tr>
<td>Rath and Strong Systems Products (RSSP) (In 1985 became McCormack &amp; Dodge)</td>
<td>Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.</td>
</tr>
<tr>
<td>SofTech, Inc.</td>
<td>Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.</td>
</tr>
<tr>
<td>Software Performance Engineering (SPE)</td>
<td>Responsible for directing the work on performance evaluation and analysis.</td>
</tr>
<tr>
<td>Structural Dynamics Research Corporation (SDRC)</td>
<td>Responsible for the User Interface and Virtual Terminal Interface Subsystems.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Contractors</th>
<th>ICAM Project</th>
<th>Contributing Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boeing Military Aircraft Company (BMAC)</td>
<td>1701, 2201, 2202</td>
<td>Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC)</td>
</tr>
<tr>
<td>Contractors</td>
<td>ICAM Project</td>
<td>Contributing Activities</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Control Data Corporation (CDC)</td>
<td>1502, 1701</td>
<td>IISS enhancements to Common Data Model Processor (CDMP)</td>
</tr>
<tr>
<td>D. Appleton Company (DACOM)</td>
<td>1502</td>
<td>IISS enhancements to Integration Methodology</td>
</tr>
<tr>
<td>General Electric</td>
<td>1502</td>
<td>Operation of the Test Bed and communications equipment.</td>
</tr>
<tr>
<td>Hughes Aircraft Company (HAC)</td>
<td>1701</td>
<td>Test Bed enhancements</td>
</tr>
<tr>
<td>Structural Dynamics Research Corps</td>
<td>1502, 1701, 1703</td>
<td>IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI)</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

## BOUND SEGMENT 1

<table>
<thead>
<tr>
<th>SECTION 1.0</th>
<th>SCOPE</th>
<th>1-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Identification</td>
<td>1-1</td>
</tr>
<tr>
<td>1.2</td>
<td>Functional Summary</td>
<td>1-1</td>
</tr>
</tbody>
</table>

## BOUND SEGMENT 2

| 3.10.8 | Module Documentation - CPFONE | 3-515 |

## BOUND SEGMENT 3

<p>| 3.10.8 | Module Documentation - INITSES | 3-717 |</p>
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS (Continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BOUND SEGMENT 4</strong></td>
</tr>
<tr>
<td>3.10.9 Include File Description 3-1084</td>
</tr>
<tr>
<td>3.10.10 Hierarchy Chart 3-1116</td>
</tr>
<tr>
<td>3.11 Program Listings Comments 3-1518</td>
</tr>
<tr>
<td><strong>SECTION 4.0</strong></td>
</tr>
<tr>
<td>QUALITY ASSURANCE PROVISIONS</td>
</tr>
<tr>
<td>4.1 Introduction and Definitions 4-1</td>
</tr>
<tr>
<td>4.2 Computer Programming and Test Evaluation 4-1</td>
</tr>
</tbody>
</table>
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:

3-315
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


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 CREATE ENTITY COMMAND
ADDKC - CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR CREATE/ALTER ENTITY
ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD

3-316
ADDMIG  - PROCESS THE ADD MIGRATES...SET..CLAUSE
ALTALI  - CONTROLS ALTER ALIAS PROCESSING (ALIAS TO PRIM
         OR VICE VERSA)
ALTCARD - PROCESS CARDINALITY FOR USER SPECIFIED RELATION
ALTDOM  - PROGRAM NAME
ALTENT  - CONTROL PROCESSING FOR ALTER ENTITY CLASS
         COMMAND.
ALTMOD  - CONTROLS PROCESSING FOR ALTER MODEL COMMANDS
ALTREL  - CONTROLS PROCESSING LOGIC FOR THE ALTER
         RELATION COMMAND
BLUWLST - 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  - CONTROLS 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
DEFCODL - 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 DATABASE
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
        LIST
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
PROCRT - 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
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
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

DESCRIPTION:

SYNOPSIS
C -- CPFVAL(LIST1, LIST2, VALUE)

COBOL -- CALL "CPFVAL" USING

LIST1,
LIST2,
VALUE.

FORTRAN
CALL CPFVAL(LIST1, LIST2, VALUE)

INPUT:
LIST1 -- THE LIST CONTAINING THE COUNTER.
LIST2 -- THE LIST WHOSE CURRENT ROW WILL BE USED AS THE INDEX INTO LIST1.

OUTPUT:

3-320
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
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:

3-322
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 ENTITY

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: CPMOD
PURPOSE: CONTROLS THE PROCESSING LOGIC FOR THE COPY MODEL COMMAND.
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CPMOD
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

3-324
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.
- CPPNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.
- OPMIFIL -
- EXCFLAG - DETERMINE IF KEYWORDS, ALIASES, DESCR ARE TO BE EXCLUDED
- ADDPARN - 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
- CLSFLIL - THIS ROUTINE CLOSES 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

3-325
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
DOCUMENTATION GROUP: PS41100

DESCRIPTION:
------------------
CREATES ALIASES FOR AN ENTITY OR ATTRIBUTE

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

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
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 INSECNM
INSECNM     - INSERT A RECORD INTO ENTIYT_NAME

3-326
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
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 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 CPFONE - 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

3-328
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
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 -
INSNOM - INSERT A RECORD INTO DOMAIN_CLASS
PROCRT - PROGRAM NAME

CALLED DIRECTLY BY:
------------------------
3-330
BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NAME: CRTENT
PURPOSE: CONTROL THE PROCESSING LOGIC FOR CREATING A NEW ENTITY CLASS
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
GETNNUM -
INSEC - INSERT A RECORD INTO ENTITY_CLASS
INSECNM - INSERT A RECORD INTO ENTITY_NAME

3-332
ADDATT - ASSOCIATES EXISTING ATT WITH ENTITY IN CREATE 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
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:

CPFONE  - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
VERENT  - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A MODEL
SPRINTF - ISSUE A MESSAGE TO THE USER, CONSIDERED A
UERROR  - ADD A CS-15 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 SYSTEM
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
GETNUM -
INSMOD - INSERT A RECORD INTO MODEL CLASS
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ERRPRO

3-336
CALLED DIRECTLY BY:

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

USED IN MAIN PROGRAM(S):

--------------------------
  NDDL/MAIN      - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NAME: CRTREL
PURPOSE: 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
GETNNUM -

3-338
CHKCARD - OBTAINS THE USER-SPECIFIED CARDINALITY (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
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

3-340
BLVWLST - CREATE BUILD VIEW LISTS FOR THE CREATE VIEW COMMAND
RENVIEW - 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
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

3-342
CALLED DIRECTLY BY:

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

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
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

3-344
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 DATABASE 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
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
GETNUM
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
NAME: DEFFLD
PURPOSE: PROCESSES THE FIELD INFORMATION FOR THE DEFINE RECORD COMMAND.

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
CPFCCOR - ACCESS A TOKEN IN A CORRESPONDING NAMED LIST
LOWUPP - CONVERT A STRING TO UPPER CASE CHARACTERS
INSDLD - INSERT A RECORD INTO THE DATA_FIELD ENTITY.
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
INSSSDFL - INSERT A RECORD INTO THE SEGMENT_DATA_FIELD

3-348
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 SYSTEM

USED IN MAIN PROGRAM(S):
------------------------
N-DDL/MAIN - MAIN PROGRAM FOR THE N-DDL 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 = DSPLY [S9(9)]
HOST-ID = DSPLY [X30]
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
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.

3-350
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):

------------------
NDDDL/MAIN - MAIN PROGRAM FOR THE NDSDL COMMAND PROCESSOR
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
CALLED DIRECTLY BY:
---------------------
DEFREC      - CONTROLS THE PROCESSING LOGIC FOR DEFINING A
             RECORD FOR THE SYSTEM

USED IN MAIN PROGRAM(S):
------------------------
NDDL/MAIN    - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NDCL 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
CPFCOR - 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
INSRKEY - INSERT A RECORD INTO THE RECORD_KEY ENTITY. IF
CPFVAL - RETURN THE COUNTER OF LIST1 BASED ON ROW IN
LIST2

3-354
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

3-355
NDLL 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

3-356
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
NAME: DEFREC
PURPOSE: CONTROLS THE PROCESSING LOGIC FOR DEFINING A RECORD FOR THE SYSTEM.

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 VERIFICATION OF THE EXISTENCE OF A RECORD TYPE.
INSTRYP - 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.
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
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

<table>
<thead>
<tr>
<th>DBMS</th>
<th>RULES</th>
<th>ORACLE TOTAL</th>
<th>IMS</th>
<th>VAX-11</th>
<th>IDMS</th>
<th>IDS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>* CODASYL DBMS'S</td>
<td>*</td>
<td>* CODASYL DBMS'S</td>
<td>*</td>
<td>* CODASYL DBMS'S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RULE</td>
<td>*</td>
<td>RULE</td>
<td>*</td>
<td>RULE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SETS</th>
<th>NO</th>
<th>YES</th>
<th>YES</th>
<th>YES</th>
<th>YES</th>
<th>YES</th>
<th>YES</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>EQUATED/</th>
<th>PTIONAL</th>
<th>NO</th>
<th>NO</th>
<th>NO</th>
<th>YES</th>
<th>YES</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR MEMBERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INCLUSION/</th>
<th>AT FIELD</th>
<th>NO</th>
<th>YES</th>
<th>NO</th>
<th>NO</th>
<th>NO</th>
<th>NO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SINGE</th>
<th>EMBER</th>
<th>NO</th>
<th>YES</th>
<th>YES</th>
<th>YES</th>
<th>YES</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOWED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MULTIPLE</th>
<th>EMBERS</th>
<th>NO</th>
<th>NO</th>
<th>NO</th>
<th>YES</th>
<th>YES</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOWED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3-360
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
GETNMNM -
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
CPFCCOR - 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):

NDIIL/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 name 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.
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
NAME: DEL1PDF
PURPOSE: DELETE A RECORD FROM PROJECT_DATA_FIELD
ENTITY
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 DELPFDFT(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 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:

```
  PDFDB - RETRIEVES PROJECT DATA FIELDS FOR A GIVEN DATA BASE
  PDFDF - RETRIEVES 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
```
NDDDL 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
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:
----------
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
- FNDACM - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL

USED IN MAIN PROGRAM(S):

- NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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, &AC_NO, &STATUS)
COBOL -- CALL "DELACAL" USING
         AC_NAME,
         AC_NO,
         STATUS.
FORTRAN -- CALL DELACAL(AC_NAME, AC_NO, 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
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
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(AC_NO, 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:
----------

3-370
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 THE
FNDACM      - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL

USED IN MAIN PROGRAM(S):
----------------------
NDDDL/MAIN   - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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

DESCRIPTION:

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:

3-372
AC_NO = INT *
STATUS = INT *

ROUTINES CALLED:

-----------------------------------------------
COSQL5   - 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 THE
FNDACM   - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL

USED IN MAIN PROGRAM(S):

-------------------------------
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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: CDM
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
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_STMT 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 *
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
MDDL COMMAND PROCESSOR Module Documentation

NAME: DELASM1
PURPOSE: DELETE A RECORD FROM AUC_ST_MAPPING ENTITY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELASM1
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
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 = 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: DELASM2
PURPOSE: DELETE A RECORD FROM AUC_ST_MAPPING ENTITY
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(DB_ID, SET_ID, STATUS)

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 TODELETE 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 *
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 DATABASE
SELRSRT  - 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
NDLL 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

3-380
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:

DELAUCL - DELETE ATTRIBUTE USE CLASS KEY MEMBER GIVEN
DELAUCL - DELETE ATTRIBUTE USE CLASS ASSOCIATED WITH ENTITY
DELAUCL - DELETE ALL ATTRIBUTE USE CLASSES FOR A GIVEN ENTITY CLASS

CALLED DIRECTLY BY:

DELOAC - DELETE ALL OWNED AND INHERITED ATTRIBUTES

USED IN MAIN PROGRAM(S):

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

3-382
ANY MIGRATION OF THOSE ATTRIBUTE USE CLASSES.

ARGUMENTS:

----------
TAG NO  =  INT *
STATUS  =  INT *

INCLUDE FILES:

----------
STDTYP   -  STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

----------
DELKCMT   -  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):

----------
NDDDL/MAIN   -  MAIN PROGRAM FOR THE NDDDL COMMAND PROCESSOR
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
DOCUMENTATION GROUP: PS41100

DESCRIPTION:
---------
TAG_NO := "INPUT PARAMETER 1" ;

SYNOPSIS
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 *
STATUS - INT

ROUTINES CALLED:

- COSQL3 - ORACLE ROUTINE
- ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE
- COEXEC - ORACLE ROUTINE
- COBINDM - 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
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 *
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
DOCUMENTATION GROUP: PS41100

DESCRIPTION:
----------------------------------
WHERE RC_NO := "INPUT PARAMETER 1" ;

SYNOPSIS
C       --  DELCPRC(RC_NO, STATUS)
COBOL    --  CALL "DELCPRC" USING
            RC_NO,
            STATUS.
FORTRAN --  CALL DELCPRC(RC_NO, 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
NAME: DELDA1
PURPOSE: DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELDA1
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C

-- DELDA1(&DB_ID,&STATUS)

COBOL

-- CALL "DELDA1" USING

DB-ID,

STATUS.

FORTRAN

-- CALL DELDA1(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
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
NAME: DELDA2
PURPOSE: DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
LANGUAGE: C
FUNCTION TYPE: FUNCTION
SOURCE FILE: DELDA2
SOURCE FILE TYPE: .C
HOST: CDM
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C
-- DELDA2(#DB_ID, #RT_ID, $STATUS)

COBOL
-- CALL "DELDA2" USING
    DB-ID,
    RT-ID,
    STATUS.

FORTRAN
-- CALL DELDA2(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
AND RETURN VALUE BOTH WILL BE 0, OTHERWISE -1.

ARGUMENTS:

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: DELDBA1
PURPOSE: DELETE A RECORD FROM DATABASE AREA ENTITY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELDBA1
SOURCE FILE TYPE: .C
HOST: 
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C -- DELDBA1(DB_ID, STATUS)

COBOL -- CALL "DELDBA1" 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 DATABASE AREA.
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
NDML 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 DATABASE)

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

3-396
ERRPRO      -  PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:
---------------
DELDFL3      -  DELETE A RECORD FROM DATA FIELD ENTITY
UERROR       -  ISSUE A MESSAGE TO THE USER, CONSIDERED A
ADDRNUM      -  ADD A AVAILABLE NO OF A POOL NO BACK TO NO
               LINKED
DELTTEXT     -  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
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
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE WAS DESIGNED FOR USE WITH DRPDB (DROP DATA BASE)

THIS ROUTINE VERIFY THE EXISTANCE OF A RECORD TYPE. SELECT A RECORD TYPE ONE AT A TIME IN THE NDDL 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 = 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

3-398
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
- ERRFRO

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
NAME: DELDBS1
PURPOSE: DELETE A RECORD FROM DATABASE ENTITY
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 "DELBDS1" 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
DATABASE
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
NDL 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: 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

3-402
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
DELTTEXT - 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):
-------------------------
NDDDL/MAIN - MAIN PROGRAM FOR THE NDDDL COMMAND PROCESSOR
NAME: DELDFL1
PURPOSE: DELETE A RECORD FROM DATA_FIELD ENTITY
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 "DELDFL1" 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 *
ROUTINES CALLED:

- COSQL3 - ORACLE ROUTINE
- ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE
- COEXEC - ORACLE ROUTINE
- COBINDN - ORACLE ROUTINE
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: CDM
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(&DBID, &RTID, &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:

3-406
ARGUMENTS:
-------------------------------
DB_ID = INT *
RT_ID = CHAR *
STATUS = INT *

INCLUDE FILES:
-----------------------
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED:
---------------------
COSQL3 - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE
COFETCH - ORACLE ROUTINE
DELTXT - DELETE A RECORD FROM DESC_TEXT ENTITY
DRPDF - DELETE A RECORD FROM DATA_FIELD ENTITY
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
NDDDL COMMAND PROCESSOR Module Documentation

NAME: DELDFL3
PURPOSE: DELETE A RECORD FROM DATA_FIELD ENTITY
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(DBID, RTID, DFID, STATUS)

COBOL  --  CALL "DELDFL3" 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:

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

USED IN MAIN PROGRAM(S):

NDDL/MMAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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
DOCUMENTATION GROUP: PS41100
DESCRIPTION:

-----------------
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 *

3-410
STATUS - INT

ROUTINES CALLED:
----------------
COSQL3 - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE
COEXEC - ORACLE ROUTINE
COBINDW - 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
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:

-------------

3-412
DOMAIN_NO = INT *
STATUS = INT *

ROUTINES CALLED:
-------------
COSQL3 - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE
COKEXEC - ORACLE ROUTINE
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:
-------------------
DRPDOM - PROGRAM NAME

USED IN MAIN PROGRAM(S):
--------------------------
WDDL/MAIN - MAIN PROGRAM FOR THE WDDL COMMAND PROCESSOR
**NAME:** DELDSL1

**PURPOSE:** DELETE A RECORD FROM DF_SET_LINKAGE ENTITY

**LANGUAGE:** C

**MODULE TYPE:** FUNCTION

**FUNCTION TYPE:** INT ()

**SOURCE FILE:** DELDSL1

**SOURCE FILE TYPE:** .C

**HOST:**

**SUBSYSTEM:** CDM

**SUBDIRECTORY:** DOCUMENTATION

**DOCUMENTATION GROUP:** PS41100

**DESCRIPTION:**

**SYNOPSIS**

C -- DELDSL1(&DB_ID,&STATUS)

COBOL -- CALL "DELSL1" 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 ENTITY
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(DBID, RT_ID, STATUS)

COBOL -- CALL "DELDSL2" 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 *
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
NAME: DELDSL3
PURPOSE: DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
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(DBID, RTID, DFID, &STATUS)

COBOL --- CALL "DELDSL3" 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:
ROUTINES CALLED:

- COSQL3 - ORACLE ROUTINE
- ERRAPT - 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
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: CDM
SUBDIRECTORY: DOCUMENTATION
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(DATATYPE_NAME,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:

3-420
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(DOMAIN_NO, 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:

3-422
---

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:

- DRPDO - PROGRAM NAME

---

USED IN MAIN PROGRAM(S):

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

---

3-423
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
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE WILL RETRIEVE 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:

3-424
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO LINKED
DELTTEXT - DELETE A RECORD FROM DESC TEXT ENTITY
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
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
DOCUMENTATION GROUP: PS41100

DESCRIPTION:
--
WHERE EC_NO =: "INPUT PARAMETER 1"

SYNOPSIS
C
  -- DELEC(EC_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:
--

3-426
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.
FNDECH - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR A GIVEN MODEL

Used In Main Program(s):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NAME: DELECAL
PURPOSE: DELETE RECORD CLASS FROM ENTITY_NAME
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: DELECAL
SOURCE FILE: .C
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(EC_NAME,EC_NO,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

3-428
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
NDL COMMAND PROCESSOR Module Documentation

NAME:  DELECKW
PURPOSE:  DELETE A RECORD FROM EC_KEYW
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:
--------

3-430
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
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
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

---------------
WHERE EC_NO =: "INPUT PARAMETER 1 "

SYNOPSIS
C
-- DELECNM(&EC_NO, &STATUS)

COBOL
-- CALL "DELECNM" USING
EC_NO.
STATUS.

FORTRAN
-- CALL DELECNMM(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:

---------------
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, TAG_NO)

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 3-434
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 RECORD FROM INHERITED_ATT_USE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT()
SOURCE FILE: DELIAUC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
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:
----------

3-436
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
NAME: DELIAUK
PURPOSE: DELETE A RECORD FROM INHERITED_ATT_USE ENTITY

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

DESCRIPTION:
------------------
SYNOPSIS
C -- DELIAUK(*KC_NO, *STATUS)

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 *
ROUTINES CALLED:

---

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

CALLED DIRECTLY BY:

---

DELMDC Kü - DELETE ALL KEY CLASSES AND INHERITED KEYS FOR AN ENTITY

USED IN MAIN PROGRAM(S):

---

NDDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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($DB_ID, $RT_ID, $DF_ID, $TAG_NO $STATUS)

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:

<table>
<thead>
<tr>
<th>DB_ID</th>
<th>INT</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT_ID</td>
<td>CHAR</td>
</tr>
<tr>
<td>DF_ID</td>
<td>CHAR</td>
</tr>
<tr>
<td>TAG_NO</td>
<td>INT</td>
</tr>
<tr>
<td>STATUS</td>
<td>INT</td>
</tr>
</tbody>
</table>

ROUTINES CALLED:

- COSQL3 - ORACLE ROUTINE
- ERRPT - 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
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.

FOSETTRAN -- 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
NAME: DELISS1
PURPOSE: DELETE A RECORD FROM IMS_SEGMENT_SIZE
ENTITY
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(DBID,STATUS)

COBOL --   CALL "DELISSI" 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 *
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/MAN   - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NAME: DELISS2
PURPOSE: DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELISS2
SOURCE FILE TYPE: .C
HOST: CDM
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

----------
SYNOPSIS

C -- DELISS2(&DBID, &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 *
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
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
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
KEY_CLASS
WHERE KC_NO = "INPUT PARAMETER 1".
IF SUCCESSFULLY DELETED THE RECORD, THE STATUS AND
RETURN VALUE BOTH
WILL BE 0, OTHERWISE -1.
ARGUMENTS:
-----------
3-448
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:  
---------------------  
DELMDCIC - DELETE ALL KEY CLASSES AND INHERITED KEYS FOR AN ENTITY  
DRPKC - CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY CLASS".  
DELMTCIC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER  

USED IN MAIN PROGRAM(S):  
--------------------------  
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR  

3-449
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 *
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".

USED IN MAIN PROGRAM(S):

-------------
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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
DOCUMENTATION GROUP: PS41100

DESCRIPTION:
---------
WHERE TAG_NO = 1 :

SYNOPSIS
C -- DELTAGMT(&TAG_NO,&STATUS)
COBOL -- CALL "DELCMT" 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 ' 3-452
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
ND-DL 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:
KW_NO INT *
STATUS INT *

Routines Called:
- COSQCLS - 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:

3-456
---

**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

---

3-457
NAME: DELKWECE
PURPOSE: DELETE A RECORD FROM EC_KEYWORER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ( )
SOURCE FILE: DELKWECE
SOURCE FILE TYPE: .C
HOST: 
SUBSYSTEM: CDM
SUBDIRECTORY: 
DOCUMENTATION GROUP: PS41100

DESCRIPTION:
---------------
WHERE KW_NO = 1 :

SYNOPSIS

C -- DELKWECE(KW_NO,STATUS)

COBOL -- CALL "DELKWECE" USING
         KW-No.,
         STATUS.

FORTRAN -- CALL DELKWECE(KWNO,STATUS)

INPUT:

INT *KW_NO

OUTPUT:

INT *STATUS

DESCRIPTION:

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

ARGUMENTS:
---------------

3-458
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
NAME: DELKWRRC
PURPOSE: DELETE A RECORD FROM RC_KEYWORD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELKWRRC
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION GROUP: PS41100

DESCRIPTION:
---------------
WHERE KW_NO =: 1 :

SYNOPSIS
C -- DELKWRRC($KW_NO,$STATUS)
COBOL -- CALL "DELKWRRC" USING
         KW_NO,
         STATUS.
FORTRAN -- CALL DELKWRRC(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:
----------

3-460
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
NDDDL 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
DELCMPR - 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
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):
---------------------
NDNL/MAIN - MAIN PROGRAM FOR THE NDNL COMMAND PROCESSOR
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:

DELRCE - DELETE A RECORD FROM RELATION CLASS
DELRCKW - DELETE A RECORD FROM RC KEYWORD
DELETETEXT - 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
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: 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:

-----------

    STDYP     - 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 RECORD FROM INHERITED_ATT_USE
    SPRINTF   - ISSUE A MESSAGE TO THE USER, CONSIDERED A
    UERROR    - 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
NAME: DELMOD
PURPOSE: DELETE A RECORD FROM MODEL_CLASS ENTITY
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 *
ROUTINES CALLED:

- COSQLS - ORACLE ROUTINE
- ERRPT - HANDLE ANY ERROR CODE FROM ORACLE
- COEXEC - ORACLE ROUTINE
- COBINDS - ORACLE ROUTINE

CALLED DIRECTLY BY:

- DRPHOD - CONTROLS THE PROCESSING LOGIC TO DROP A MODEL

USED IN MAIN PROGRAM(S):

- HDOL/MAIN - MAIN PROGRAM FOR THE HDOL COMMAND PROCESSOR
NAME: DELMTEC
PURPOSE: DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT()
SOURCE FILE: DELMTEC
SOURCE FILE TYPE: C
HOST: SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C -- DELMTEC(#MODEL_NO)

COBOL -- CALL "DELMTEC" USING MODEL-No.

FORTRAN -- CALL DELMTEC (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

3-470
ARGUMENTS:

---------------

MODEL_NO = INT

INCLUDE FILES:

---------------

STDTYP - 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
DELTTEXT - 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
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-MO = 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

3-472
DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED ATTRIBUTE CLASSES FROM THE

FWDOAC - DELETES ALL OWNED ATTRIBUTES FOR A GIVEN ENTITY CLASS

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NAME: DELOACE
PURPOSE: DELETE A RECORD FROM OWNED_ATTRIBUITE 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 *
Routines Called:

- **COSQL3** - Oracle Routine
- **ERRRPT** - Handle any error code from Oracle
- **COEXEC** - Oracle Routine
- **COBINDN** - Oracle Routine

Called Directly By:

- **FNDECH** - Deletes all entities and associated objects for a given model

Used in Main Program(s):

- **NDDL/MAIN** - Main Program for the NDDL Command Processor
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:
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
NAME: DELPCB
PURPOSE: DELETE A RECORD FROM PSB_PCB ENTITY
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 "DELPCB" 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

3-478
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):

NDDDL/MAIN - MAIN PROGRAM FOR THE NDDDL COMMAND PROCESSOR
NAME:  DELPDFT  
PURPOSE:  DELETE A RECORD FROM PROJECT DATA FIELD ENTITY  
LANGUAGE:  C  
MODULE TYPE:  FUNCTION  
FUNCTION TYPE:  INT ( )  
SOURCE FILE:  DELPDFT  
SOURCE FILE TYPE:  C  
HOST:  SUBSYSTEM:  CDH  
SUBDIRECTORY:  DOCUMENTATION GROUP:  PS41100  

DESCRIPTION:  

SYNOPSIS  
C  
DELPDFT( #TAG_NO, #STATUS )  

COBOL  
CALL "DELPDFT" USING  
TAG_NO,  
STATUS  

FORTRAN  
CALL DELPDFT( TAG_NO, 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 '  

A 498
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 - RETRIEVES ALL THE PROJ DATA FIELD OR DF BEING DROPPED
- PDFREC - RETRIEVES ALL PROJECT DATA FIELDS FOR A GIVEN RECORD
- DRPMAP - COMMAND PROCESSOR FOR THE DROP MAP COMMAND

USED IN MAIN PROGRAM(S):

- WDDL MAIN - MAIN PROGRAM FOR THE WDDL COMMAND PROCESSOR
NDDDL 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
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 *
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
NAME
DELRRB1

PURPOSE
DELETE A RECORD FROM RC.BASED.REC.Set

LANGUAGE
C

MODULE TYPE
FUNCTION

FUNCTION TYPE
INT ()

SOURCE FILE
DELRRB1

SOURCE FILE TYPE
C

HOST

SUBSYSTEM
CDM

SUBDIRECTORY

DOCUMENTATION GROUP
PS41100

DESCRIPTION

SYNOPSIS
C

DELRRB1 (DBID, STATUS)

COBOL
CALL "DELRRB1" USING
DB-ID.
STATUS

FORTRAN
CALL DELRRB1 (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 '
ROUTINES CALLED:

- COSQL3  - ORACLE ROUTINE
- ERRPT    - HANDLE ANY ERROR CODE FROM ORACLE
- COELEC   - ORACLE ROUTINE
- COBINDS  - 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
NAME: DELRBR2
PURPOSE: DELETE A RECORD FROM RC_BASED_REC_SET
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELRBR2
SOURCE FILE TYPE: .C
HOST: 
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
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: 
----------

3-486
PS 620141100
1 November 1985

DB ID = INT *
SET ID = CHAR *
STATUS = INT *

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

- DRPSET - CONTROLS THE PROCESSING LOGIC FOR DELETING A SET FROM THE DATABASE
- 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):

- NDLDL/MAIN - MAIN PROGRAM FOR THE NDLDL COMMAND PROCESSOR
NAME: DELRBR3
PURPOSE: DELETE A RECORD FROM RC_BASED_REC_SET

LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT()
SOURCE FILE: DELRBR3
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C
-- DELRBR3(#DB_ID, #SET_ID, #RT_ID, #STATUS)

COBOL
-- CALL "DELRBR3" USING

-- DB_ID,
-- SET_ID,
-- RT_ID,

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:

<table>
<thead>
<tr>
<th>DB ID</th>
<th>INT *</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET ID</td>
<td>CHAR *</td>
</tr>
<tr>
<td>RT ID</td>
<td>CHAR *</td>
</tr>
<tr>
<td>STATUS</td>
<td>INT *</td>
</tr>
</tbody>
</table>

ROUTINES CALLED:

- COSQL3 - 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):

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

3-490
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):
---
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
---
NAME: DELRCKW
PURPOSE: DELETE A RECORD FROM RC_KEYWOD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT()
SOURCE FILE: DELRCKW
SOURCE FILE TYPE: C
HOST: 
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
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:
-----------

3-492
RC NO - INT
STATUS - INT

ROUTINES CALLED:

- OOSQL3 - ORACLE ROUTINE
- ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE
- COEXEC - ORACLE ROUTINE
- COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:

- DELMDRC - 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
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

MOST

SUBSYSTEM
CDM

SUBDIRECTORY

DOCUMENTATION GROUP
PS41100

DESCRIPTION

WHERE RC_NO = "INPUT PARAMETER 1".

SYNOPSIS

C
-- DELRCST(RC_NO, STATUS)

COBOL
-- CALL "DELRCST" USING
RC_NO,
STATUS.

FORTRAN
-- CALL DELRCST(RC_NO, 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'

3-494
STATUS - INI T

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: DELREUS
PURPOSE: C
LANGUAGE: FUNCTION
MODULE TYPE: INT ( )
FUNCTION TYPE: DELREUS
SOURCE FILE: .C
SOURCE FILE TYPE: 
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 *
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
NAME: DELRKM1
PURPOSE: DELETE A RECORD FROM RECORD_KEY_MEMBER
ENTITY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELRKM1
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
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 *
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
NAME: DELRKM2
PURPOSE: DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY

LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT()
SOURCE FILE: DELRKM2
SOURCE FILE TYPE: .C

DESCRIPTION:

SYNOPSIS

C -- DELRKM2(&DBID, &RT_ID, &WSTATUS)

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:

3-500
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:

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

USED IN MAIN PROGRAM(S):

---
NDDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
---
NAME: DELRK3M3
PURPOSE: DELETE A RECORD FROM RECORD KEY MEMBER ENTITY

LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELRK3M3
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
DESCRIPTION GROUP: PS41100

DESCRIPTION:
-----------

SYNOPSIS

C -- DELRK3M3(DB_ID, RT_ID, DF_ID, STATUS)

COBOL -- CALL "DELRK3M3" USING
DB-ID,
RT-ID,
DF-ID,
STATUS.

FORTRAN -- CALL DELRK3M3(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.

3-502
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
NAME: DELRKY1
PURPOSE: DELETE A RECORD FROM RECORD_KEY ENTITY
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 *
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
NAME: DELRKY2
PURPOSE: DELETE A RECORD FROM RECORD KEY ENTITY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ( )
SOURCE FILE: DELRKY2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C -- DELRKY2( &DB_ID, &RTID, &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 *
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
---

3-507
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(&DBID, 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:

---
DELDDBST - DELETE ALL RECORD SETS' Associations with the DATABASE
DRPSET - CONTROLS THE PROCESSING LOGIC FOR DELETING A SET FROM THE DATABASE
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET BEING DROPPED
SELSSTM - DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE BEING DROPPED
---

USED IN MAIN PROGRAM(S):

---
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
---
NAME: DELRST3
PURPOSE: DELETE A RECORD FROM RECORDSET ENTITY
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 RECORDSET.
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_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
NAME: DELRTY2
PURPOSE: DELETE A RECORD FROM RECORD_TYPE ENTITY
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:
--------------------
DELDRT     - 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
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 *
ROUTINES CALLED:

- COSQL3 - ORACLE ROUTINE
- ERRRT - 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(&DBID, &URTID, &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:
-----------
3-516
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
NAME: DELSDF3
PURPOSE: DELETE A RECORD FROM SEGMENT_DATA_FIELD
ENTITY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELSDF3
SOURCE FILE TYPE: .C
HOST: 
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
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.

3-518
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
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(MACNO, &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 *

3-520
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):

----------
NDDDL/MAIN - MAIN PROGRAM FOR THE NDDEL COMMAND PROCESSOR
NAME: DELESCR
PURPOSE: DELETE RECORD CLASS FROM SEC_RC_COMPONENT
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT()
SOURCE FILE: DELESCR
SOURCE FILE TYPE: .C
HOST: CDM
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION GROUP: PS41100

DESCRIPTION:
WHERE VIEWNO = "INPUT PARAMETER 1":

SYNOPSIS
C
-- DELESCR( VIEWNO, STATUS )

COBOL
-- CALL "DELESCR" USING VIEWNO, STATUS.

FORTRAN
-- CALL DELESCR(VIEWNO,STATUS)

INPUT:
INT *VIEWNO ;

OUTPUT:
INT *STATUS ;

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

ARGUMENTS:
---
VIEWNO = INT *

3-522
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
NAME: DELSN1
PURPOSE: DELETE A RECORD FROM SCHEMA NAMES ENTITY
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 -- CALL "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 *

3-524
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
NDNL COMMAND PROCESSOR Module Documentation

NAME: DELSTM1
PURPOSE: DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
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 *
ROUTINES CALLED:

- COSQL5 - ORACLE ROUTINE
- ERRPT - 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/MAN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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(&DBID, SETID, &STATUS)

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:

----------

3-528
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 DATABASE
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET BEING DROPPED
SELSMH - DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE BEING DROPPED

USED IN MAIN PROGRAM(S):
-------------------------
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NAME: DELSTM3
PURPOSE: DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: DELSTM3
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
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.
ARGUMENTS:

---

<table>
<thead>
<tr>
<th>DB_ID</th>
<th>INT *</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET_ID</td>
<td>CHAR</td>
</tr>
<tr>
<td>RT_ID_OF_MEMBER</td>
<td>CHAR *</td>
</tr>
<tr>
<td>STATUS</td>
<td>INT</td>
</tr>
</tbody>
</table>

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
NDDDL COMMAND PROCESSOR Module Documentation

NAME: DELTEXT
PURPOSE: DELETE A RECORD FROM DESC_TEXT ENTITY
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(CHAR*OBJECT_TYPE, INT*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:

3-532
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
DELDDBRT - DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE DATABASE
DELDDBST - 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 THE DATABASE.
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE DATABASE.
DRPDIV - DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED WITH VIEW
DRPDOM - 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.
DRPREE - 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 DATABASE.
DRPVIEW - DROP THE VIEW.
FNDACH - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL
FNDECH - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR A GIVEN MODEL
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET BEING DROPPED
SELSTN - 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
DELMTCG - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER

USED IN MAIN PROGRAM(S):

-----------------------------
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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
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
           (OBJ_TYPE, DESC_TYPE, OBJ_NO, STATUS)

INPUT:
   CHAR OBJ_TYPE[
   CHAR DESC_TYPE[
   INT *OBJ_NO

OUTPUT:
   INT *STATUS

DESCRIPTION

DELTXT DELETES ALL DESCRIPTION TEXT CORRESPONDING TO A

3-535

ARGUMENTS:

OBJ TYPE = CHAR []
DESC TYPE = CHAR []
OBJ NO = INT *
STATUS = INT *

INCLUDE FILES:

STDTYP - 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

3-536
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
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
STRNCPY
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

3-538
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, ECNO)

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 ;

3-539
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:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC_NO</td>
<td>INT*</td>
</tr>
<tr>
<td>MODEL_NAME</td>
<td>CHAR*</td>
</tr>
<tr>
<td>KEYWORD_FLAG</td>
<td>CHAR*</td>
</tr>
<tr>
<td>ALIAS_FLAG</td>
<td>CHAR*</td>
</tr>
<tr>
<td>DESC_FLAG</td>
<td>CHAR*</td>
</tr>
</tbody>
</table>

ROUTINES CALLED:

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSQL3</td>
<td>ORACLE ROUTINE</td>
</tr>
<tr>
<td>ERRRPT</td>
<td>HANDLE ANY ERROR CODE FROM ORACLE</td>
</tr>
<tr>
<td>COFETCH</td>
<td>ORACLE ROUTINE</td>
</tr>
<tr>
<td>STRNCMP</td>
<td>GENERATE NDDL DESCRIBE COMMANDS FOR A GIVEN OBJECT TYPE AND NO</td>
</tr>
<tr>
<td>GENDESC</td>
<td>GENERATE NDDL COMMANDS FOR A GIVEN OBJECT TYPE AND NO</td>
</tr>
<tr>
<td>BLECLST</td>
<td>THIS ROUTINE CREATES A ROW IN THE UNBOUNDED EC_LIST</td>
</tr>
<tr>
<td>GENENT</td>
<td>GENERATE A CREATE ENTITY..OWNED ATTRIBUTE..KEYWORD COMMAND</td>
</tr>
<tr>
<td>STRNCPY</td>
<td>GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT TYPE</td>
</tr>
<tr>
<td>GENALI</td>
<td>GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT TYPE</td>
</tr>
<tr>
<td>COEXEC</td>
<td>ORACLE ROUTINE</td>
</tr>
<tr>
<td>COBINDN</td>
<td>ORACLE ROUTINE</td>
</tr>
<tr>
<td>CODFINN</td>
<td>ORACLE ROUTINE</td>
</tr>
</tbody>
</table>

CALLED DIRECTLY BY:

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CESTRUC</td>
<td>GENERATES NDDL COMMANDS ON A FILE FOR ALL ENTITIES FOR THE STRUCT</td>
</tr>
</tbody>
</table>

USED IN MAIN PROGRAM(S):

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDDL/MAIN</td>
<td>MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR</td>
</tr>
</tbody>
</table>

3-540
NDDDL 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
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)]
FROM-MOD-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

3-541
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 - CONTROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.

USED IN MAIN PROGRAM(S):

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

3-543
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
BLRCKG1 - FOR EACH LEVEL OF RELATIONS IN STRUCTURE
COPETCH - 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

3-544
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.

FORTTRAN -- 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:

3-545
MODEL_NO = INT *

INCLUDE FILES:

STDTYP - 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.
TXTTYP - DETERMINE SOURCE OF DESCRIPTION TEXT
RDDESC - STORE DESCRIPTION ON THE CDM
LOADESC - LOAD DESCRIPTION FROM TEXT EDITOR
TOUNPER
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):

NDDDL/MAIN - MAIN PROGRAM FOR THE NDDDL COMMAND PROCESSOR
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_1D), 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 *

3-547
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 DATABASES

USED IN MAIN PROGRAM(S):

- NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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_USF, KEY_CLASS MEMBER,
SELECMN - RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS
UEERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ERRPRO

3-549
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
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

FORTRAN -- 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 3-551
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:

-------------
STDTYPE - 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 ADDRN useMemo - 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

3-552
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 = dspl [s9(9)]
return-status = dspl [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
CALLED DIRECTLY BY:
----------------------
DRPDOM - PROGRAM NAME

USED IN MAIN PROGRAM(S):
------------------------
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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, THEIR KEY CLASS NOS AND NAMES AND KEY CLASS TAG NUMBERS AND NAMES.

SYNOPSIS
C -- DPKCLST(&MODEL_NO, &ECNO);

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
SRVRTET - 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

3-557
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
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

3-559
GETECAL - 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
NAME: DRPATT
PURPOSE: CONTROLS THE DROPPING OF USER SPECIFIED
ATTRIBUTE CLASSES FROM THE CDM.

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
DELAC - DELETE A RECORD FROM ATTRIBUTE_CLASS
DELACNM - DELETE RECORD CLASS FROM ATTRIBUTE_NAME WHERE AC_NO
DELACKW - DELETE A RECORD FROM AC_KEYWOR
DELTTEXT - DELETE A RECORD FROM DESC_TEXT ENTITY
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-I COBOL  
**MODULE TYPE:** PROGRAM  
**SOURCE FILE:** DRPDB  
**SOURCE FILE TYPE:** .COB  
**HOST:** CDM  
**SUBDIRECTORY:** DOCUMENTATION  
**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.

ADDITIONALy 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 RETRUNED 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 ENTITY
DELSN1   - DELETE A RECORD FROM SCHEMA_NAMES ENTITY
DELDDBRT - DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE
           DATABASE
DELDAA1  - DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDDBDF - DELETE ALL DATA FIELD ASSOCIATIONS WITH THE
           DATABASE
DELDDBST - DELETE ALL RECORD SETS’ ASSOCIATIONS WITH THE
           DATABASE
DELSTM1  - DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELRKY1  - DELETE A RECORD FROM RECORD_KEY ENTITY
DELRKM1  - DELETE A RECORD FROM RECORD_KEY MEMBER ENTITY
DELSH1   - DELETE A RECORD FROM AUC_ST_MAPPING ENTITY
DELRBR1  - DELETE A RECORD FROM RC_BASED_REC_SET ENTITY
DELLSS1  - DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
DELDBA1  - DELETE A RECORD FROM DATA_BASE AREA ENTITY
DELDLS1  - DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELSDF1  - DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
PDFDB    - RETRIEVES PROJECT DATA FIELDS FOR A GIVEN DATA
           BASE
DELPCB   - DELETE A RECORD FROM PSB_PCB ENTITY
DELTTEXT  - DELETE A RECORD FROM DESC_TEXT ENTITY
ADDRNUM  - ADD A AVAILABLE NO OF A POOL NO BACK TO NO
           LINKED

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
WDDL COMMAND PROCESSOR Module Documentation

NAME: DRPDF
PURPOSE: DELETE A RECORD FROM DATA_FIELD ENTITY
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(DF_NO, 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 *
ROUTINES CALLED:
---------------
COSQL3 - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE
COEXEC - ORACLE ROUTINE
COBINDN - ORACLE ROUTINE

CALLED DIRECTLY BY:
---------------
DELDL2 - CONTROLS THE DELETING OF DATA FIELDS

USED IN MAIN PROGRAM(S):
---------------
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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
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:

-------------

3-568
DELTEXT   - DELETE A RECORD FROM DESC_TEXT ENTITY
ADDRNUM   - ADD A AVAILABLE NO OF A POOL NO BACK TO NO LINKED

: ERRPRO

: CALLED DIRECTLY BY:
              ----------------
   DRPVIEW    - DROP THE VIEW.

USED IN MAIN PROGRAM(S):
              ---------------------
   NDDL/MAIN   - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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 THAT THE DOMAIN TO BE DELETED EXISTS. 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 VERACD - DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH ATTRIBUTE CLASSES
DOMUSAG - DETERMINE IF DOMAIN ASSOCIATED WITH VIEWS, DATAFIELDS, ATTRIBUTES
DELT TWO - DELETE ALL DATA TYPES ASSOCIATED WITH A DOMAIN
DELDTD - DELETE RECORD CLASS FROM USER_DEF_DATA-TYPE
DELDOM - DELETE RECORD CLASS FROM DOMA IN_CLASS
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO LINKED
DELTTEXT - DELETE A RECORD FROM DESC_TEXT ENTITY
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/MA IN - 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):
-------------------------
3-572
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NDDDL 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 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

3-574
DELMTKC  - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
DRPRCE  - DROP A RELATION CLASS FOR AN ENTITY BEING DROPPED
DELECMN  - 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 ENTITY
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
NDSDL 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: CDM
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

3-576
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
DELDFL3  - 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    - RETRIEVES 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
NDDDL 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

3-578
DELECH - DELETE A RECORD FROM KEY CLASS MEMBER
DELCHPR - DELETE A RECORD FROM COMPLETE RELATION
DELEK - DELETE A RECORD FROM KEY CLASS
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO LINKED
CPPEXIT - 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)

MDDL MAIN - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR
EXODI COMMAND PROCESSOR Module Documentation

NAME DRPKV
PURPOSE DROP A KEYWORD ASSOCIATION FROM EITHER AN ATTRIBUTE. ENTITY

LANGUAGE VAX-11 COBOL
MODULE TYPE SUBROUTINE
SOURCE FILE DRPKV
SOURCE FILE TYPE COB
HOST
SUBSYSTEM CDM
SUBDIRECTORY
DOCUMENTATION GROUP P541100

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 - RECUR
OBJ. ID - DSPLY [59(9)]
RTN STATUS - DSPLY [59(9)]

INCLUDE FILES

LISTMOS - Valid List Numbers

ROUTINES CALLED

CPFONE - Extract the first token from the named list
LOWUPP - Convert a string to upper case characters
VERK - 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
DELKWRG - Delete a record from RC KEYWORD
CPFNXT - Access the next token in a parser list.

3-580
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
NDLL 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
HOST:
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_KEYW
DELKWEc - DELETE A RECORD FROM EC_KEYW
DELKWRC - DELETE A RECORD FROM RC_KEYW

3-582
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
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 COMMA' PROCESSOR
NDDL COMMAND PROCESSOR Module Documentation

NAME: DRPMAP
PURPOSE: COMMAND PROCESSOR FOR THE DROP MAP COMMAND
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ( )
SOURCE FILE: DRPMAP
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION

SYNOPSIS

C

DRPMAP

COBOL

CALL 'DRPMAP'

FORTRAN

CALL DRPMAP

INPUT

OUTPUT

DESCRIPTION

DRPMAP 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.

3-584
INCLUDE FILES:

- STDTYPE - 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
- SPRTNTF
- UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
- DELRQST - DELETE RECORD CLASS FROM RC BASED REC SET
- VERQST - REPORT WHETHER A RC TO SET HAP EXISTS
- DELPROFT - DELETE A RECORD FROM PROJECT DATA FIELD ENTITY
- DELASH - DELETE FROM ATTRIBUTE USE CLASS MAPPING
- PDFSRCH - DETERMINE IF A TAG HAS A PRIMARY MAP TO DATA FIELD
- VERASH - DETERMINE IF ANY AUC/SET MAPPINGS EXIST FOR AN AUC
- CPFNXN - 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
WDDL COMMAND PROCESSOR Module Documentation

NAME: DRPMGK
PURPOSE: DELETE MIGRATED KEY CLASSES ASSOCIATED WITH A KEY CLASS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: DRPMGK
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 NDDL 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 [89(9)]
RETURN-STATUS = DSPLY [89(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

3-586
CALLED DIRECTLY BY:

DRPEK - CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY CLASS".

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NAME: DRPMGRC
PURPOSE: DELETE MIGRATED KEY CLASS MEMBERS ASSOCIATED WITH A KEY CLASS
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

3-588
UError - Issue a message to the user, considered a
error

called directly by:

- DRPHIG - Controls the processing logic for the "drop
migrate" 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" CLAUSE

**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 CLAUSE 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.
- **DRPHMGRC** - DELETE MIGRATED KEY CLASS MEMBERS ASSOCIATED WITH A KEY CLASS
- **DELCPRC** - DELETE A RECORD FROM COMPLETE_RELATION

3-590
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
DELTTEXT - DELETE A RECORD FROM DESC TEXT ENTITY
FNDECH - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR
A GIVEN MODEL
FNDACHM - 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
ND DL 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

3-594
DELTEXT  - DELETE A RECORD FROM DESC_TEXT ENTITY
DELRCKW  - 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
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
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 ENTITY
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO
LINKED

DELTEXT - DELETE A RECORD FROM DESC TEXT ENTITY
DENDARK - DELETE A RECORD FROM DB AREA ASSIGNMENT ENTITY
DELDFL2 - CONTROLS THE DELETING OF DATA FIELDS
PDFREC - RETRIEVES ALL PROJECT DATA FIELDS FOR A GIVEN RECORD
DRLRKY2 - DELETE A RECORD FROM RECORD KEY ENTITY
DRLRKM2 - DELETE A RECORD FROM RECORD KEY MEMBER ENTITY
DELISS2 - DELETE A RECORD FROM IMS SEGMENT SIZE ENTITY
DELSDF2 - DELETE A RECORD FROM SEGMENT DATA FIELD ENTITY
DELDSSL2 - DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
SELRESSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET BEING DROPPED
SELRSTM - 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):

-----------------
NDDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NDDDL COMMAND PROCESSOR Module Documentation

NAME: DRPREL
PURPOSE: CONTROLS THE PROCESSING LOGIC FOR THE "DROP RELATION" COMMAND
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

3-598
DELRC - DELETE A RECORD FROM RELATION CLASS
DELRCKW - 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

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: DRPSET
PURPOSE: CONTROLS THE PROCESSING LOGIC FOR DELETING A SET FROM THE DATABASE.

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

DESCRIPTION:
---------------

THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR DELETING A SET FROM THE DATABASE.

THIS ROUTINE FETCHES INFORMATION REGARDING THE DATABASE AND THE SET TO BE DROPPED. AFTER VERIFYING THAT THE SET DOES EXIST, IT IS DELETED FROM THE DATABASE.

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 DATABASE FOR A GIVEN NAME AND HOST
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A SET
VERRSET - VERIFIES THE EXISTANCE OF A SET.
DELRS2 - DELETE A RECORD FROM RECORD_SET ENTITY
DELSTM2 - DELETE A RECORD FROM SET_TYPE MEMBER ENTITY
DELASM2 - DELETE A RECORD FROM AUC_ST_MAPPING ENTITY

3-600
DELRR2R - 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
DELTTEXT - 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:

3-602
MAP_TYPE = CHAR [ ]
TAG_NO = INT *
RC_NO = INT *

INCLUDE FILES:
----------------
STDTYP - STANDARD TYPE DEFINITIONS
LISTID - PROVIDES LIST OF PARSED OBJECTS

ROUTINES CALLED:
----------------
CPFNX1T - ACCESS THE NEXT TOKEN IN A PARSER LIST.
CPPCOR - ACCESS A TOKEN IN A CORRESPONDING NAMED LIST
FDNRCH - RETRIEVE 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
FNDASM - DETERMINES IF AN AUC TO SET TYPE MAPPING EXISTS
TOLOWER
GETMAPC - SELECT ALL PROJECT DATA FIELDS MATCHING A GIVEN TAG NO
STRNCMP
SPRINTF
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
DELIICS - DELETE RECORD CLASS FROM RC_BASED_REC_SET
DELIASM - DELETE RECORD CLASS FROM AUC_ST_MAPPING
DELPDF - 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

3-603
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 AND SEC-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
DELPIDI - DELETE RECORD CLASS FROM PROJECT_DATA_ITEM
DELSERCR - DELETE RECORD CLASS FROM SEC_RC_COMPONENT
DRPDIV - DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED
         WITH VIEW
DELDIV - DELETE FROM DATA_ITEM WHERE
DELSER - DELETE RECORD CLASS FROM SEC
ADDRNUM - ADD A AVAILABLE NO OF A POOL NO BACK TO NO
LINKED
DELTEXT   - DELETE A RECORD FROM DESC TEXT ENTITY
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
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:
-----------------
VERFDT - 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.
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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: CDM
SUBDIRECTORY: DOCUMENTATION
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-N01 = DSPLY [S9(9)]
MOD-N02 = 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

3-608
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
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: CDN
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
C --
ERRRPT( RETCODE, MESSG, MOD_NAME)

COBOL
CALL "ERRRPT" USING

FORTRAN --
CALL ERRRPT(RETCOD, MESSG, MODNAM)

INPUT:

RETCODE -- THE ORACLE ERROR CODE.

MESSG -- A CHARACTER STRING CONTAINING THE ERROR MESSAGE FOR THE ERROR LOG.

MOD_NAME -- THE NAME OF THE MODULE EXPERIENCING 3-610
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
GOERMSG - 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

3-611
BLRCKC1 - FOR EACH LEVEL OF RELATIONS IN STRUCTURE
CHIPLOOP - 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
DELACNMM - 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
DELCHPR - DELETE A RECORD FROM COMPLETE_RELATION
DELCPRC - DELETE A RECORD FROM COMPLETE_RELATION
DELDAA1 - DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDAA2 - DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDBA1 - DELETE A RECORD FROM DATA_BASE_AREA ENTITY
DFLDBS1 - 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
DELEG - DELETE RECORD CLASS FROM ENTITY_NAME
DELECAL - DELETE RECORD CLASS FROM ENTITY_CLASS
DELECKW - DELETE A RECORD FROM EC_KEYWORD
DELECNM - DELETE THE RECORD FROM ENTITY_NAME
DELAASM - DELETE RECORD CLASS FROM AUC_ST_MAPPING
DELIAUC - DELETE A RECORD FROM INHERITED_ATT_USE
DELAUK - DELETE A RECORD FROM INHERITED_ATT_USE ENTITY
DELIPDF - DELETE RECORD CLASS FROM PROJECT_DATA_FIELD
DELIRCSC - 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
DELKMCT - 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
DELOAC - DELETE A RECORD FROM OWNED ATTRIBUTE ENTITY
DELOWAC - DELETE A RECORD FROM OWNED ATTRIBUTE
DELPCB - 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
DELRCKW - DELETE A RECORD FROM RC KEYWORD
DELRNST - DELETE RECORD CLASS FROM_RC_BASED_REC_SET
DELRUS - DELETE A RECORD FROM...
DELRLKM1 - DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRLKM2 - DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRLKM3 - DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRLKY1 - DELETE A RECORD FROM RECORD_KEY ENTITY
DELRLKY2 - DELETE A RECORD FROM RECORD_KEY ENTITY
DELRLRT2 - DELETE A RECORD FROM RECORD_SET ENTITY
DELRLST2 - DELETE A RECORD FROM RECORD_SET ENTITY
DELRLST3 - 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
DELSOCR - DELETE RECORD CLASS FROM_SEC_RC_COMPONENT
DELSN1 - DELETE A RECORD FROM SCHEMA NAMES ENTITY
DELSSTM1 - DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSSTM2 - DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSSTH3 - DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSTEXT - DELETE A RECORD FROM DESC TEXT ENTITY
DELTXT - DELETE DESCRIPTION TEXT GIVEN THE OBJECT TYPE
DEPATT - SELECT ALL THE ATTRIBUTES IN THE
DEPRENT - SELECT ALL THE DEPENDANT ENTITY CLASSES
DEPREL - FOR EACH LEVEL OF RELATIONS IN STRUCTURE
DEPGEN - GENERATE
DLDSSL2 - DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELMIGRC - 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

3-613
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
INSAUC - INSERT A RECORD INTO INHERITED_ATT_USE
INSIIS - 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_KEYWOR
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
INSPWRRD - INSERT A RECORD INTO THE DB_PASSWORD. IF SUCCESSFUL,

INSRC - INSERT A RECORD INTO RELATION_CLASS
INSRCS - INSERT A RECORD INTO RC_BASED_REC_SET
INRESUS - 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_NAME ENTITY. IF

INSSDFL - INSERT A RECORD INTO THE SEGMENT_DATA_FIELD ENTITY. IF
INSSEC - INSERT A RECORD INTO SEC
INSSSECR - INSERT A RECORD INTO SEC_RC_COMPONENT

3-614
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
UPDACLNM - UPDATE ATTRIBUTE_NAME SET AC_NAME = 1
UPDECL - 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
UPDTDT - 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 ENTITY
WRTDSC4 - SELECT A RECORD FROM DESC_TEXT ENTITY

USED IN MAIN PROGRAM(S):

DELDLFL1 - DELETE A RECORD FROM DATA_FIELD ENTITY
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

3-615
NAME: EXCFLAG
PURPOSE: DETERMINE IF KEYWORDS, ALIASES, DESC 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 FLAG 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:
---------

3-616
CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST

CALLED DIRECTLY BY:

---

CMBENT - CONTROLS 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
**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  
**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:**  
---------------------  
REMVIEW - POPULATE VIEW DEFINITION AND ES-CS-MAPPING  

**USED IN MAIN PROGRAM(S):**  
-------------------------------  
3-618
MDDL/MAIN - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR
NDDDL 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: CDM
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):

-----------
3-620
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

**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:**

3-622
VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A MODEL
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
OPNFIL  - WRITES 80 CHAR NDDL COMMAND WITH PARAMETERS
ADDPARM - CHECKS/DELMITER
GENATT - GENERATE A CREATE ATTRIBUTE COMMAND
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN
GENALI  - OBJECT TYPE AND NO
GENALI  - GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT
CLSFIL  - TYPE
ERRPRO  - THIS ROUTINE Closes AN OUTPUT FILE, THE FILE WILL

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
NAME: FOOPENT
PURPOSE: DETERMINE IF COPY ENTITY WITH STRUCTURE
         OR RELATION
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: FOOPENT
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
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

3-624
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
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 EXITS.

EXT.

ARGUMENTS:
----------
LDA = RECRD
TEXTCUR = RECRD
FILE-NAME = DSPLY [X(30)]
OBJ-TYPE = DSPLY [X(30)]
DESC-TYPE = DSPLY [X(30)]
OBJ-NO = DSPLY [SS9(9)]
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
OEEXEC
ERRPRO
---------

CALLED DIRECTLY BY:

---------
RDDESC - STORE DESCRIPTION ON THE CDM
---------

USED IN MAIN PROGRAM(S):

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

3-628
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

3-630
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 NDDL 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:

3-631
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: FNDACN
PURPOSE: DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL

LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: FNDACM
SOURCE FILE TYPE: .PRC
HOST: CDM
SUBDIRECTORY: DOCUMENTATION
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

3-633
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:

DRPMOD - CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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
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
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

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
USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDRL COMMAND PROCESSOR
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:
3-639
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: FNDEC
PURPOSE: DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR A GIVEN MOD
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: FNDEC
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
DELETTEXT - DELETE A RECORD FROM DESC TEXT ENTITY
DLMDAUC - DELETE ATTRIBUTE USE CLASSES ASSOCIATED WITH

3-641
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
NAME: FNDOAC
PURPOSE: DELETES ALL OWNED ATTRIBUTES FOR A GIVEN ENTITY CLASS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: FNDOAC
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:
------------------------------
FNDOAC WILL RETRIEVE ALL OWNED ATTRIBUTES FOR AN ENTITY CLASS AND DEBATE 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:
-------------------
3-643
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING ENTITIES.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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

3-645
CALLED DIRECTLY BY:

-----------------------

DRP$MAP - DROP A SINGLE MAPPING

USED IN MAIN PROGRAM(S):

--------------------

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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
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

3-647
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS
            CHECKS/DELIMITER
ERRPRO

CALLED DIRECTLY BY:
---------------------
CMBENT - CONTROLS THE PROCESSING LOGIC FOR THE COMBINE
          ENTITY COMMAND.

USED IN MAIN PROGRAM(S):
------------------------
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NDDDL 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

3-649
USED IN MAIN PROGRAM(S):

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

3-652
NDDDL 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 NDDEL MODELLING COMMANDS

ROUTINES CALLED:
----------------
NEXTKC - THIS ROUTINE RETURNS A KC_NAME FOR A GIVEN EC_NO FROM
ADDPARM - WRITES 80 CHAR NDDEL COMMAND WITH PARAMENTERS CHECKS/DLIMITER
NEXTKCM - THIS ROUTINE RETURNS A KG_NO,KG_NAME FOR A GIVEN EC_NO

CALLED DIRECTLY BY:

3-653
ALLKEY - GENERATES KEY CLASS FOR AN ENTITY

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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 PARAMETERS
CHECKS/DELMITER
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 - CONTROLS THE PROCESSING LOGIC FOR THE COMBINE
ENTITY COMMAND.

MRGNODE - SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT ENTITY'S

TOPMODE - SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT ENTITY

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

3-656
NDDDL 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 NDDDL command with parameters
GENAKW - checks/delimiter
          retrieves all keywords for an attribute class

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

CMBOA - generate commands for attributes, its
         keywords, alias, desc
FCOPATT - generate NDVL commands from a copy attribute
          command
ALLATT - select all the attributes in from-model and
          generate

3-657
DEPATT - SELECT ALL THE ATTRIBUTES IN THE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NDDDL COMMAND PROCESSOR Module Documentation

NAME: GENDESC
PURPOSE: GENERATED NDDDL DESCRIBE COMMANDS FOR A GIVEN OBJECT TYPE AN
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

3-660
ADDPARM - WRITES 80 CHAR HDOL COMMAND WITH PARAMETERS CHECKS/DELFNTER
ERRPRO

cALLED DIRECTLY BY:

CMBOA  - GENERATE COMMANDS FOR ATTRIBUTES, ITS KEYWORDS, ALIAS, DESC
DEFFROM  - GENERATE CREATE RELATION, DESCRIBE COMMANDS IN THE TO-MODEL
FOOPATT  - GENERATE HDOL COMMANDS FROM A COPY ATTRIBUTE COMMAND
GENENT1  - GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND DESCrs.
INDFROM  - RETRIEVES RELATIONS, DETERMINES IND RC AND GENERATES HDOL
ALLATT  - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND GENERATE
ALLENT  - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND GENERATE
ALLEL  - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL GENERATE
DEPPATT  - SELECT ALL THE ATTRIBUTES IN THE DEPENDANT ENTITY CLASSES
DEPREL  - FOR EACH LEVEL OF RELATIONS IN STRUCTURE GENERATE
MKMOD2  - CONTROL THE LOGIC FOR PROCESSING THE REMAINING MODEL_2

USED IN MAIN PROGRAM(S):

HDOL/Main - MAIN PROGRAM FOR THE HDOL COMMAND PROCESSOR
NAME: GENEKV
PURPOSE: SELECT KEYWORD FOR ENTITY AND CREATE
KEYWORD PHRASE FOR CRT
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: GENEKV
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-N0 - DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMETERS
CHECKS/DELIMITER
ERRPRO

CALLED DIRECTLY BY:

GENENT - GENERATE A CREATE ENTITY.OWNED

3-662
ATTRIBUTE..KEYWORD COMMAND

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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

3-664
DEPENT - SELECT ALL THE DEPENDANT ENTITY CLASSES

USED IN MAIN PROGRAM(S):

NDDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NDDL COMMAND PROCESSOR Module Documentation

NAME: GENT1
PURPOSE: GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND DESCRS.
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: GENT1
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 [89(9)]
TO-MOD-NO = DSPLY [89(9)]
FROM-EC-NO = DSPLY [89(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 [89(9)]

INCLUDE FILES:

ERRPRO - PROCESS ERROR INCLUDE FILE

3-886
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 - CONTROLS 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
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:
3-668
GENENT  - GENERATE A CREATE ENTITY..OWNED
        ATTRIBUTE..KEYWORD COMMAND

USED IN MAIN PROGRAM(S):

NDDL/MAIN  - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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
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:
-------------
GENRNME - FORMAT THE CREATE RELATION CLAUSE
MIGREL - GENERATE A MIGRATES CLAUSE FOR A CREATE

3-670
RELATION COMMAND
ADDPARM       - writes 80 char NDDL command with parameters
               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
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 PARAMETERS
CHECKS DELIMITER
ERRPRO

CALLED DIRECTLY BY

GENREL - GENERATE CREATE RELATION MIGRATES KEYWORD
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 PARAMETERS
            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
MODEL 2

USED IN MAIN PROGRAM(S)

MDOL/MAIN - MAIN PROGRAM FOR THE MDOL COMMAND PROCESSOR
NAME GETACAL
PURPOSE VERIFY THE EXISTENCE OF AN ALIAS FOR AN ATTRIBUTE
LANGUAGE VAX 11 COBOL
MODULE TYPE SUBROUTINE
SOURCE FILE GETACAL
SOURCE FILE TYPE PBC
HOST
SUBSYSTEM CDN
SUBDIRECTORY
DOCUMENTATION GROUP PS41100

DESCRIPTION
VERIFY THE EXISTENCE 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

AM. NO = DISPLAY (S99999)
ALIAS ID = DISPLAY (X30)

INCLUDE FILES

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

ROUTINES CALLED

DELCAL DELETE RECORD CLASS FROM ATTRIBUTE NAME
VARERROR ISSUE A MESSAGE TO THE USER. CONSIDERED A
ERRPRO CALLED DIRECTLY BY

3-676
DROP THE ALIAS FOR AN ENTITY OR ATTRIBUTE.

USED IN MAIN PROGRAM(S):

MDOL/MAIN - MAIN PROGRAM FOR THE MDOL COMMAND PROCESSOR
NAME: GETDBST
PURPOSE: RETURN INFORMATION ABOUT THE CURRENT SESSIONS' DATABASE

LANGUAGE: VAX-11 COBOL
MODUL TYPE: SUBROUTINE
SOURCE FILE: GETDBST
SOURCE FILE TYPE: .COB

HOST: SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE RETURNS WITH INFORMATION ABOUT THE DATABASE, WHICH IS OBTAINED EITHER FROM THE PARSER LISTS OR FROM INFORMATION WHICH IS GLOBALLY AVAILABLE.

EXTRACT THE DATABASE 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 DATABASE 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:

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

CALLED DIRECTLY BY:

- DEFSDEF - CONTROLS THE PROCESSING LOGIC FOR THE DEFINE
  SET COMMAND

USED IN MAIN PROGRAM(S):

- NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
# NDOL 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:** CDM  
**SUBSYSTEM:** CDM  
**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 (80(9))  
- **TAG-NAME** = DSPLY (X(30))  
- **DOM-NO** = DSPLY (80(9))  
- **TAG-NO** = DSPLY (80(9))  

## INCLUDE FILES:

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

## ROUTINES CALLED:

- **ERRPRO**
CALLED DIRECTLY BY:

CHECKDMS - VERIFY THAT THE DATA ITEM AND ATTRIBUTE USE CLASS.

ENVVIEW - CREATE ES-CS-MAPPING FOR USER SPECIFIED RETRIEVE LIST

USED IN MAIN PROGRAM(S):

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB-ID</td>
<td>DSPLY [S9(9)]</td>
</tr>
<tr>
<td>DBMS-NAME</td>
<td>DSPLY [X(30)]</td>
</tr>
<tr>
<td>RT-ID</td>
<td>DSPLY [X(30)]</td>
</tr>
<tr>
<td>RT-NO</td>
<td>DSPLY [S9(9)]</td>
</tr>
<tr>
<td>RET-STATUS</td>
<td>DSPLY [S9(9)]</td>
</tr>
</tbody>
</table>

INCLUDE FILES:

<table>
<thead>
<tr>
<th>Include File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LISTNUMS</td>
<td>VALID LIST NUMBERS</td>
</tr>
</tbody>
</table>

ROUTINES CALLED:

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPFONE</td>
<td>EXTRACT THE FIRST TOKEN FROM THE NAMED LIST</td>
</tr>
<tr>
<td>UERROR</td>
<td>ISSUE A MESSAGE TO THE USER, CONSIDERED A</td>
</tr>
<tr>
<td>VERDBAS</td>
<td>VERIFY A DATA BASE FOR A GIVEN NAME AND HOST</td>
</tr>
<tr>
<td>VERRT</td>
<td>VERIFIES THE EXISTANCE OF A RECORD TYPE.</td>
</tr>
</tbody>
</table>

CALLED DIRECTLY BY:

3-682
DRPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP RECORD COMMAND.

USED IN MAIN PROGRAM(S):

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

3-684
USED IN MAIN PROGRAM(S):

MDDL/MAIN - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR
**NAME** | GETECNM  
---|---
**PURPOSE** | THIS ROUTINE SEARCHES THE UNBOUNDED EC LIST DATA  
**LANGUAGE** | C  
**MODULE TYPE** | FUNCTION  
**FUNCTION TYPE** | INT ()  
**SOURCE FILE** | GETECMN  
**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  
---  
GETECMN(EC_NO, EC_NAME, STATUS)  

COBOL  
---  
CALL "GETECNM" USING  
EC-NO,  
EC-NAME,  
STATUS  

FORTRAN  
---  
CALL GETECMN(ECNO, ECNAME, STATUS)  

**INPUT**  
INT *EC_NO;  
CHAR *EC_NAME;  
INT *STATUS;  

**OUTPUT**  
  
**DESCRIPTION**  
  
**ARGUMENTS**  
  
---  
EC_NO = INT *
EC NAME = CHAR *
STATUS = INT *

#include files:

keylist - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

routines called:

strcpy

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):

nndl/main - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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 = DPLY [S9(9)]
NDDL-STATUS = DPLY [S9(9)]

INCLUDE FILES:

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR
ORCLED A - WS DEFINITION FOR THE ORACLE LOGIN AREA
ERRFRO - 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
UVARN - 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
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:

3-690
OUTPUT:

- DB_ID,
- DB_NAME,
- HOST_NAME
- DBMS_NAME,
- STATUS.

DESCRIPTION

This routine will get the following 4 global variables:
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 variables 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' DATABASE INFO

USED IN MAIN PROGRAM(S):

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

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

ROUTINES CALLED:

ERRPRO
CALLED DIRECTLY BY:
-------------------
  DRPSMAP      - DROP A SINGLE MAPPING

USED IN MAIN PROGRAM(S):
------------------------
  NDDL/MAIN      - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
**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  
**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
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:

----------

STDTYP - 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
NAME: GETNUM

PURPOSE: 

LANGUAGE: C

MODULE TYPE: FUNCTION

FUNCTION TYPE: INT ()

SOURCE FILE: GETNUM

SOURCE FILE TYPE: .C

HOST: 

SUBSYSTEM: CDM

SUBDIRECTORY: 

DOCUMENTATION GROUP: PS41100

DESCRIPTION: 

SYNOPSIS

C -- GETNUM (POOL_NO, NEXT_NO)

COBOL -- CALL "GETNUM" USING
POOL-No,
NEXT-No.

FORTRAN -- CALL GETNUM (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 -

3-696
ERRPRO/UEERROR - 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 - RETRIEVES 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
NDDDL 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 (POOLNO, NEXTNO, STATUS)

COBOL -- CALL "GETNXNO" USING

POOLNO,

NEXTNO,

STATUS.

FORTRAN -- CALL GETNXNO (POOLNO, NEXTNO, STATUS)

INPUT:

INT *POOLNO ;

OUTPUT:

INT *NEXTNO ;

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 *
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
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:
---------------------

3-700
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 - RETRIEVE AN ENTITY CLASS NAME FOR A GIVEN NO. AND NAME TYPE
ERRPRO

3-702
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
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
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
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:

3-706
LISTID - PROVIDES LIST OF PARSED OBJECTS

ROUTINES CALLED:

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
STRNCHP
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
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
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:
-----------------
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 - RETRIEVE AND WRITE ALL THE KEYWORDS ASSOCIATED WITH AN ATTRIBUTE C
WRTDESC - SELECT A RECORD FROM DESC_TEXT ENTITY
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

3-709
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:
----------------------
GETNNUM
INSEC
UERROR
INSECNM
WRTECKW
WRDESC
WRTENAM
COPYAC
ERRPRO

CALLED DIRECTLY BY:
----------------------
COPENT

USED IN MAIN PROGRAM(S):
----------------------
NDDL/MAIN
NDDDL COMMAND PROCESSOR Module Documentation

NAME: INDFROM
PURPOSE: RETRIEVES RELATIONS, DETERMINES IND EC AND GENERATES NDNL
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: INDFROM
SOURCE FILE TYPE: .PRC
HOST: SUBSYSTEM: CDM
SUBDIRECTORY: DOCUMENTATION
DOCUMENTATION GROUP: PS41100
DESCRIPTION:
---------

THIS ROUTINE SELECTS ALL THE RELATIONS IN WHICH THE
ROM-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

3-712
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 PARAMETERS CHECKS/DLIMITER
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN OBJECT TYPE AND NO
ERRPRk

CALLED DIRECTLY BY:

CERELS - GENERATES NDDL COMMANDS ON A FILE FOR ALL ENTITIES IN A RELATION.
CMBENT - CONTROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

3-713
NDDDL 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 NDDDL LISTS
NDDDL - **** PURPOSE NOT FOUND BY STRIPPER ****

CALLED DIRECTLY BY:

3-714
PRCCMD - THIS ROUTINE IS CALLED TO HANDLE ALL USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
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