

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A181 954

DTIC FILE COPY

2

**AFWAL-TR-86-4006
Volume V
Part 6**



**INTEGRATED INFORMATION
SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 6 - NDDL Processor Product Specification
Sections 1.0 through 3.10.8 CFPNXT**

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

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

Approved for public release; distribution is unlimited.

PREPARED FOR:

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

**DTIC
ELECTE
JUN 24 1987
S E D**

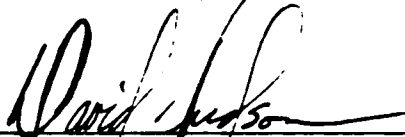
87 6 23 049

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

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

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




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

5 Aug 1986

DATE

FOR THE COMMANDER:



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

7 Aug 86

DATE

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/MLTC, W-PAFB, OH 45433 to help us maintain a current mailing list."

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

REPORT DOCUMENTATION PAGE

A181954

1a REPORT SECURITY CLASSIFICATION Unclassified		1b RESTRICTIVE MARKINGS	
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b DECLASSIFICATION/DOWNGRADING SCHEDULE		4. PERFORMING ORGANIZATION REPORT NUMBER(S)	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFVAL-TR-86-4006 Vol V, Part 6	
6a NAME OF PERFORMING ORGANIZATION General Electric Company Production Resources Consulting	6b OFFICE SYMBOL (If applicable) AFVAL/MLTC	7a NAME OF MONITORING ORGANIZATION AFVAL/MLTC	
6c ADDRESS (City, State and ZIP Code) 1 River Road Schenectady, NY 12345		7b ADDRESS (City, State and ZIP Code) WPAFB, OH 45433-6535	
8a NAME OF FUNDING/SPONSORING ORGANIZATION Materials Laboratory Air Force Systems Command, USAF	8b OFFICE SYMBOL (If applicable) AFVAL/MLTC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER 733619-80-C-5155	
8c ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433		10 SOURCE OF FUNDING NOS.	
11. TITLE (Include Security Classification) (See Reverse)		PROGRAM ELEMENT NO. 78011F	PROJECT NO. 7500
12. PERSONAL AUTHOR(S) Singh, S., Althoff, J. L., and Apicella, M. L.		TASK NO. 62	WORK UNIT NO. 01
13a TYPE OF REPORT Final Technical Report	13b TIME COVERED 22 Sept 1980 - 31 July 1985	14. DATE OF REPORT (Yr., Mo., Day) 1985 November	15. PAGE COUNT 330
16 SUPPLEMENTARY NOTATION ICAM Project Priority 6201		The computer software contained herein are theoretical and/or references that in no way reflect Air Force-owned or -developed computer software.	
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB GR.	
1908	0905		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>This document is the product specification establishing the design implementation of the IISS Configuration Item Neutral Data Definition Language NDDL which is the primary tool used for maintaining the Common Data Model (CDM) data base. → See p 1-1</p>			
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL David L. Judson		22b TELEPHONE NUMBER (Include Area Code) 615-255-8976	23. OFFICE SYMBOL AFVAL/MLTC

11. Title

Integrated Information Support System (IISS)
Vol V - Common Data Model Subsystem
Part 6 - NDDL Processor Product Specification
Sections 1.0 through 3.10.8 CPFNXT

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



PREFACE

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

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

The subcontractors and their contributing activities were as follows:

TASK 4.2

Subcontractors

Role

Boeing Military Aircraft
Company (BMAC)

Reviewer

D. Appleton Company
(DACOM)

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

General Dynamics/
Ft. Worth

Responsible for factory view
function and information
models

Subcontractors

Role

**Illinois Institute of
Technology**

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

North American Rockwell

Reviewer

Northrop Corporation

**Responsible for factory view
function and information
models**

Pritsker and Associates

Responsible for IDEF2 support

SofTech

Responsible for 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.**

PS 620141100
1 November 1985

Subcontractors

Role

Digital Equipment
Corporation (DEC)

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

McDonnell Douglas
Automation Company
(McAuto)

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

On-Line Software
International (OSI)

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

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

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

SofTech, Inc.

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

Software Performance
Engineering (SPE)

Responsible for directing the
work on performance evaluation
and analysis.

Structural Dynamics
Research Corporation
(SDRC)

Responsible for the User
Interface and Virtual Terminal
Interface Subsystems.

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

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

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

TABLE OF CONTENTS

	<u>Page</u>
<u>BOUND SEGMENT 1</u>	
SECTION 1.0	SCOPE 1-1
1.1	Identification 1-1
1.2	Functional Summary 1-1
SECTION 2.0	DOCUMENTS 2-1
2.1	Reference Documents 2-1
2.2	Terms and Abbreviations 2-1
SECTION 3.0	REQUIREMENTS 3-1
3.1	Structural Description 3-1
3.2	Functional Flow 3-1
3.3	Interfaces 3-2
3.3.1	Inputs/Outputs 3-3
3.4	Program Interrupts 3-3
3.5	Timing and Sequencing Description ... 3-3
3.6	Special Control Features 3-3
3.7	Storage Allocation 3-4
3.7.1	Database Definition 3-4
3.7.1.1	File Description 3-4
3.7.1.2	Table Description 3-4
3.7.1.3	Item Description 3-4
3.8	Object Code Creation 3-4
3.9	Adaptation Data 3-4
3.10	Detail Design Description 3-4
3.10.1	Main Program List 3-6
3.10.2	Module List 3-8
3.10.3	External Routines List 3-39
3.10.4	Include File List 3-42
3.10.5	Where Include File Used List 3-44
3.10.6	Where External Routine Used List .. 3-75
3.10.7	Main Program Parts List 3-94
3.10.8	Module Documentation 3-109
<u>BOUND SEGMENT 2</u>	
3.10.8	Module Documentation - CPFONE 3-315
<u>BOUND SEGMENT 3</u>	
3.10.8	Module Documentation - INITSES 3-717

TABLE OF CONTENTS (Continued)

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

PS 620141100
1 November 1985

FOREWORD

This is the first bound segment of four comprising Volume V, Part 6 of the Final Technical Report. It contains Sections 1.0 through 3.10.8 CPFNXT.

SECTION 1

SCOPE

1.1 Identification

This specification establishes the development, test and qualification requirements of a computer program identified as the Neutral Data Definition Language Processor (NDDL Processor). The NDDL Processor is one Configuration Item of the Integrated Information Support System (IISS) Common Data Model (CDM) subsystem.

1.2 Functional Summary

The NDDL Processor is a language used to manipulate and populate information in the Common Data Model (CDM) of the IISS CDM database. The language is modelled after SQL and the commands feature a combination of a few simple verbs (operators) along with the necessary parts of the CDM (objects).

The following functions will be performed by the NDDL processor:

1. Allow the user to populate and maintain the three schemas of the CDM: An external, conceptual and internal and the mappings between each.
2. Provide capabilities for manipulation of many IDEF-1 models and submodels needed during the process of developing a single integrated model of the conceptual schema. Only the integrated model may be mapped to the external and internal schemas.
3. Provide the user with three modes of operation:
 - a) Batch mode allows NDDL command files to be executed, and output may be redirected to a user specified file.
 - b) Interactive mode allow the user to enter NDDL commands at the terminal.
 - c) Forms mode allows the user the capability of using the IISS forms processor to display input and output screens of NDDL commands.
4. Allow the user to describe and maintain textual

descriptions of all objects in the CDM.

5. Define alias names for entities and attributes in the CDM.
6. Associate keywords for entities, attributes and relations.
7. Provide the user with additional modelling tools to Copy, Merge, Combine, and Compare conceptual entities, attributes and IDEF1 models.

SECTION 2

DOCUMENTS

2.1 Reference Documents

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

2.2 Terms and Abbreviations

Attribute Use Class: (AUC)

C/E Transformer: Conceptual Schema to External Schema Translation or Mapping program.

Conceptual Schema: (CS)

Common Data Model Processor: (CDMP)

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

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

Database Management System: (DBMS)

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

Domain: A logical definition of legal attribute class values.

Domain Constraint: Predicate that applies to a single domain.

External Schema: (ES)

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

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

Internal Schema: (IS)

ICAM Definition: (IDEF1)

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

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

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

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

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

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

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

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

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

SECTION 3
REQUIREMENTS

3.1 Structural Description

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

This CPCI uses many lower level modules to handle the following operations:

1. Initialize with NTH, forms processor, ORACLE DBMS, and other global variables and data structures.
2. Execute the commands sequentially in the order presented by the user.
3. Parse each command for syntactic correctness, using UNIX tools YACC and LEX
4. Transfer control to individual command processors for semantic validation of each command.
5. Access the CDM database to retrieve information.
6. MODIFY, INSERT or DELETE rows of CDM tables whenever information has to be updated.
7. Provide a single standard means of communicating errors to the NDDL user. Error handling allows the database transactions to be committed or rolled back.

3.2 Functional Flow

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

This CPCI has been designated to operate in a batch or interactive mode and forms mode. It must operate in the system environment established for IISS; that is, the Network Transaction Manager and UI/VTI. It currently can only be executed on the DEC VAX due to the dependence on the VAX Sort. This can be changed for execution on the IBM.

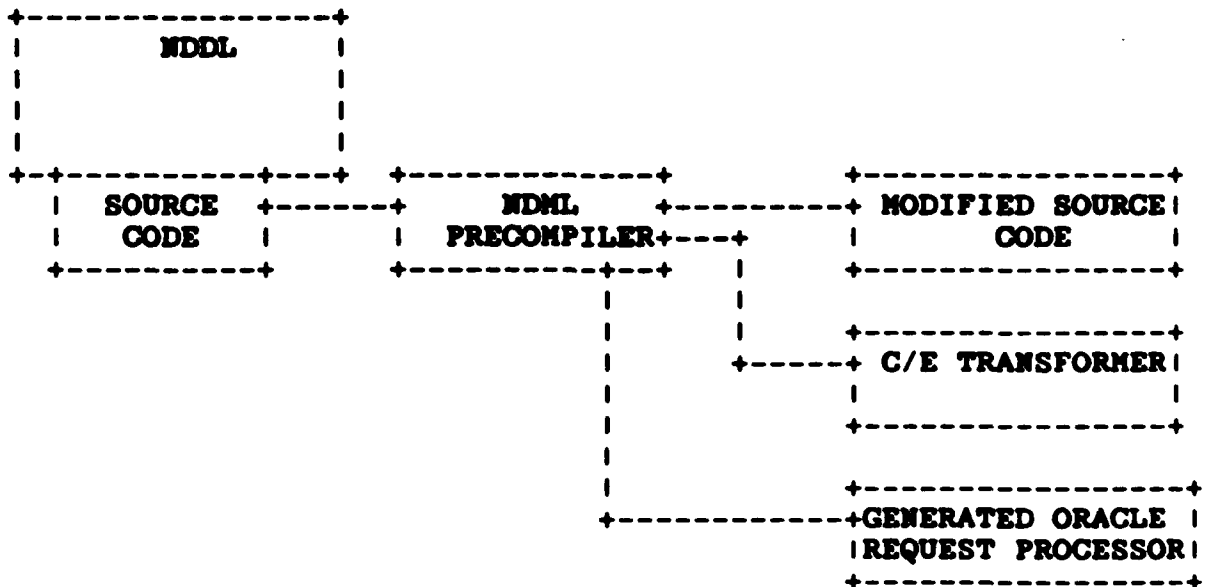
3.3 Interfaces

The NDDL processor makes use of the IISS Forms Processor directly for forms interactive input and output. NDDL also make use of the standard "C" input/output library to allow the user for interactive input or batch input via file redirection database access is through a combination of:

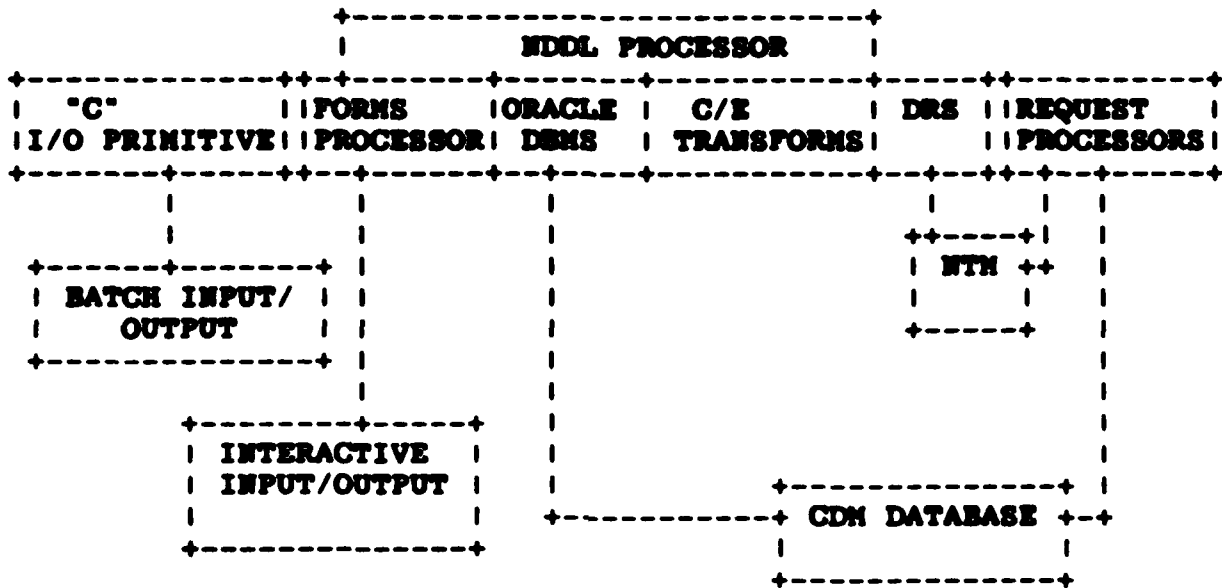
- a) ORACLE for recursive searches not supported by NDML and a few update routines.
- b) NDML for searches and a few update routines.

NDML routines are precompiled by the IISS NDML precompiler and ORACLE request processors are generated which communicates with NDDL through the DRS which uses IISS NTM services.

Precompile Interfaces



Runtime Interfaces



3.3.1 Inputs/Outputs

Input to components of the NDDL processor is done in an invisible manner without respect to the means in which the input was obtained. Output from the NDDL processor to the user is provided through a standard interface to allow for the same invisibility.

INPUT

OUTPUT

3.4 Program Interrupts

Not applicable to this CPCI.

3.5 Timing and Sequencing Description

Not applicable to this CPCI.

3.6 Special Control Features

When NDDL is activated, at the "args": prompt, entering a

-I will display an NDDL form to enable the user to execute NDDL commands.

3.7 Storage Allocation

3.7.1 Database Definition

The database used by this CPCI is the Common Data Model (CDM) database. The model is defined by the CDM1, the IDEFl model of the CDM, Reference Document Number 3.

3.7.1.1 File Description

No permanent files have been defined for this CPCI. It may use temporary scratch files for such things as input and results.

3.7.1.2 Table Description

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

3.7.1.3 Item Description

Not applicable to this CPCI.

3.8 Object Code Creation

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

3.9 Adaptation Data

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

PS 620141100
1 November 1985

3.10 Detail Design Description

The following sections have been computer generated for this CPCI.

3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module Documentation (section 3.10.8) and look at Module Type for each of the Main Program modules listed. Note whether the routine is a Program, Subroutine, or Function. If it is a Program, it is truly a main program entry point. If not, then it is merely called by other programs not being documented here.

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Main Program List

Module Name -----	Purpose -----
DELDFL1	DELETE A RECORD FROM DATA_FIELD ENITY
NDDL/MAIN	MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141100
1 November 1985

3.10.2 Module List

The following is a list of all the modules being documented here along with their purpose. Each module has a unique name, no matter what language it was written in.

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
ADDATT	ASSOCIATES EXISTING ATT WITH ENTITY IN CREAT ENTITY COMMAND
ADDDT	PROGRAM NAME
ADDEC	ADD THE ENTITY NAME TO THE TREE LIST STRUCTURE
ADDECNM	ADD THE EC_NAME AND EC_NO INTO KEYLIST
ADDKC	CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR CREATE/ALTER ENTITY
ADDKCLS	ADD KEY INFO TO THE UNBOUNDED KEY_CLASS_LIST STRUCTURE
ADDKG	ADD KCM_TAG NUMBER AND NAME TO STRUCTURE
ADDKM	ADDS SINGLE KEY CLASS MEMBER(AUC) TO KEY CLASS FOR ADD KEY
ADDKW	ADDS KEYWORDS FOR COMMANDS USING "ADD KEYWORD"(OPTIONAL)
ADDKWA	INSERT AN ATTRIBUTE KEYWORD
ADDKWE	INSERT AN ENTITY KEYWORD
ADDKWR	INSERT A RELATION KEYWORD
ADDHAP	ADD A CS-IS MAPPING
ADDHIG	PROCESS THE ADD MIGRATES...SET..CLAUSE
ADDNCORR	ADD A TOKEN TO A CORRESPONDING LISTS NEXT ENTRY
ADDNSTD	ADDS A USER DEFINED DATA TYPE
ADDOAC	ADD ATTRIBUTE AS AN OWNED ATTRIBUTE AND AS ATT USE CLASS

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
ADDPARM	WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS CHECKS/DELIMITER
ADDRCEC	POPULATES THE RC-DEPKC TABLE FOR ALL RELATIONS IN THE MODEL
ADDRNUM	ADD A AVAILABLE NO OF A POOL NO BACK TO NO LINKED
ADDSTD	PROGRAM NAME
ADDTXT	THIS ROUTINE WRITES 80 CHARACTERS OF DESCRIPTIVE TEXT
ADD_CORR	ADD A TOKEN TO CORRESPONDING LIST
ADD_TO_CNT	INCREMENT A LIST COUNTER
ADD_TO_LST	ADD A SINGLE TOKEN TO A PARSER OUTPUT LIST
ADPARM1	CREATES 80 CAHR NDDL COMMANDS WITH PARAMETERS AND DELIMITERS
AKCROW	THIS ROUTINE ADDS A ROW TO THE UNBOUNDED KEY_CLASS_LIST
ALLATT	SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND GENERATE
ALLENT	SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND GENERATE
ALLKEY	GENERATES KEY CLASS FOR AN ENTITY
ALLREL	FOR EACH LEVEL OF RELATIONS IN FROM-MODEL GENERATE
ALLVIEW	CREATE AN ES-CS-MAPPING FOR AUC TO DATA ITEM

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
ALTALI	CONTROLS ALTER ALIAS PROCESSING (ALIAS TO PRIM OR VICE VERSA)
ALTATT	THE ALTER ATTRIBUTE COMMAND PROCESSOR ALTERS
ALTCARD	PROCESS CARDINALITY FO USER SPECIFIED RELATION
ALTDOM	PROGRAM NAME
ALTDT	PROGRAM NAME
ALTENT	CONTROL PROCESSING FOR ALTER ENTITY CLASS COMMAND.
ALTMAP	ALTER MAP COMMAND PROCESSOR
ALTMOD	CONTROLS PROCESSING FOR ALTER MODEL COMMANDS
ALTREL	CONTROLS PROCESSING LOGIC FOR THE ALTER RELATION COMMAND
ALTSMAP	ALTER A SINGLE MAP
ATTKW	CONTROLS PROCESSING TO POPULATE KEYWORD TABLE FOR AUC KEYWORD
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)
BLECLST	THIS ROUTINE CREATES A ROW IN THE UNBOUNDED EC_LIST
BLKCL1	STORE ALL KEY CLASS INFO FOR AN ENTITY IN A LIST

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
BLKCLST	SELECT AND STORE KEY CLASS INFO FOR A GIVEN ENTITY
BLOOPCK	CHECK FOR LOOPS FROM THE ENTITY GIVEN UP THE HIERARCHY
BLRCKC	FOR EACH LEVEL OF RELATIONS IN FROM-MODEL FIND
BLRCKC1	FOR EACH LEVEL OF RELATIONS IN STRUCTURE
BLSECRC	CONTROLS PROCESS TO BUILD SEC-RC COMPONENTS
BLVWLST	CREATE BUILD VIEW LISTS FOR THE CREATE VIEW COMMAND
BRANCHR	PERFORMS MULTI-WAY CALL TO THE
CDP4A	VERIFY SURROGATE ENTITY CLASS STRUCTURE
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 STRUCTURE
CHGDOM	CHANGE THE DOMAIN FOR AN ALTERED ATTRIBUTE
CHGLOBL	CHANGES THE GLOBAL DB PARAMETERS
CHKATT	CHECK IF ATTRIBUTES HAVE BEEN CREATED ACCORDING TO STANDARDS
CHKAUCV	CHECK EXISTENCE OF AUC TO SET MAPPING
CHKCARD	OBTAINS THE USER-SPECIFIED CARDINALTY (IF ANY) FOR THE RELATION.
CHKDOMS	VERIFY THAT THE DATA ITEM AND ATTRIBUTE USE CLASS.

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
CHKINH	checks the RC-DEPKC table and determines when a key class can be add
CHKKEYS	CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
CHKLOOP	CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS
CHKMOD	DETERMINES WHETHER CERTAIN RULES ARE FULFILLED.
CHKOWN	checks each of the OWN-ec's key classes against the RC-DEPKC_LIST.
CHKREL	CHECK IF RELATIONS HAVE BEEN CREATED ACCORDING TO STANDARDS
CKDUPEC	POPULATES A TABLE WITH ENTITY NAMES.
CKRNLST	SEARCHES THE TABLE OF RENAME PAIR LOOKING FOR AN OLD-TAG ENTRY.
CLSFIL	THIS ROUTINE CLOSSES AN OUTPUT FILE. THE FILE WILL
CMBACAL	GENERATE CREATE ALIAS ATTRIBUTE.. AND ALIAS DESC TEXT COMMANDS
CMBALI	GENERATE CREATE ALIAS ENTITY..COMMAND
CMBEKW	GENERATE ADD KEYWORD CLAUSE FOR ENTITY KEYWORDS
CMBENT	CONROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.
CMBOA	GENERATE COMMANDS FOR ATTRIBUTES, ITS KEYWORDS,ALIAS,DESC

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
CMERKV	SELECT AND GENERATE RELATION CLASS KEYWORDS
CMPMOD	CONTROLS THE PROCESSING LOGIC TO COMPARE TWO MODELS.
COBINDN	ORACLE ROUTINE
COCLOSE	ORACLE ROUTINE
COCOF	ORACLE ROUTINE
COCOM	ORACLE ROUTINE
CODFINN	ORACLE ROUTINE
GOERMSG	ORACLE ROUTINE
GOEXEC	ORACLE ROUTINE
GOFETCH	ORACLE ROUTINE
COLOGOF	ORACLE ROUTINE
COLON	ORACLE ROUTINE
COMMIT	STORE THE REUSEABLE NUMBER TO THE DATA BASE.
COOPEN	ORACLE ROUTINE
COPATT	CONTROLS THE PROCESSING LOGIC FOR COPYING AN ATTRIBUTE.
COPENT	CONTROLS THE PROCESSING LOGIC FOR THE COPY ENTITY COMMAND.
COPYAC	CREATE AN ATTRIBUTE, ASSOCIATE WITH ENTITY, ADD KEY CLASSES
COROL	ORACLE ROUTINE

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
COSQL3	ORACLE ROUTINE
CPFCOR	ACCESS A TOKEN IN A CORRESPONDING NAMED LIST
CPFNXT	ACCESS THE NEXT TOKEN IN A PARSER LIST.
CPFONE	EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
CPFVAL	RETURN THE COUNTER OF LIST1 BASED ON ROW IN LIST2
CPYDES	COMMAND PROCESSOR FOR COPY DESCRIPTION
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
CRTEXT	CONTROL THE PROCESSING LOGIC FOR CREATING A NEW ENTITY CLASS.
CRTMAP	CREATE MAP COMMAND PROCESSOR
CRTMOD	CONTROL THE PROCESSING LOGIC IF CREATING A MODEL WITHIN THE SYSTEM.
CRTREL	CONTROLS THE LOGIC FOR VALIDATING AND CREATING A NEW RELATION CLASS.
CRTVIEW	CONTROLS THE PROCESSING LOGIC FOR THE CREATE VIEW COMMAND.

MDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
DEFAREA	PROCESSES THE AREA INFORMATION IF THE DBMS IS IDS-II, IDMS AND VAX-1
DEFCODL	PROCESSES THE DBMS TYPES: VAX-11, IDMS, IDS-II.
DEFDB	CONTROLS THE PROCESSING LOGIC FOR DEFINING A DATABASE TO THE SYSTEM.
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.
DEFREC	CONTROLS THE PROCESSING LOGIC FOR DEFINING A RECORD FOR THE SYSTEM.
DEFSET	CONTROLS THE PROCESSING LOGIC FOR THE DEFINE SET COMMAND.
DEFTOT	PROCESSES THE DBMS TYPE: TOTAL.
DEL1PDF	DELETE A RECORD FROM PROJECT_DATA_FIELD ENITY
DELAC	DELETE A RECORD FROM ATTRIBUTE_CLASS
DELACAL	DELETE RECORD CLASS FROM ATTRIBUTE_NAME
DELACKW	DELETE A RECORD FROM AC_KEYWORD
DELACNM	DELETE RECORD CLASS FROM ATTRIBUTE_NAME WHERE AC_NO

WDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
DELASH	DELETE FROM ATTRIBUTE USE CLASS MAPPING
DELASH1	DELETE A RECORD FROM AUC_ST_MAPPING ENTITY
DELASH2	DELETE A RECORD FROM AUC_ST_MAPPING ENTITY
DELAUC	DELETE INHERITED_ATT_USE, KEY_CLASS_MEMBER,
DELAUCK	DELETE ATTRIBUTE USE CLASS KEY MEMBER GIVEN
DELAUCL	DELETE A RECORD FROM ATTRIBUTE_USE_CL
DELCMPR	DELETE A RECORD FROM COMPLETE_RELATION
DELCPRC	DELETE A RECORD FROM COMPLETE_RELATION
DELDA1	DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDA2	DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDBA1	DELETE A RECORD FROM DATA_BASE_AREA ENTITY
DELDBDF	DELETE ALL DATA FIELD ASSOCIATIONS WITH THE DATABASE
DELDBRT	DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE DATABASE
DELDBS1	DELETE A RECORD FROM DATA_BASE ENTITY
DELDBST	DELETE ALL RECORD SETS' ASSOCIATIONS WITH THE DATABASE
DELDFL1	DELETE A RECORD FROM DATA_FIELD ENTITY
DELDFL2	CONTROLS THE DELETING OF DATA FIELDS

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
DELDL3	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
DELDTWO	DELETE ALL DATA TYPES ASSOCIATED WITH A DOMAIN
DELEC	DELETE THE RECORD FROM ENTITY_CLASS
DELECAL	DELETE RECORD CLASS FROM ENTITY_NAME
DELECKW	DELETE A RECORD FROM EC_KEYWORD
DELECNM	DELETE THE RECORD FROM ENTITY_NAME
DELIASH	DELETE RECORD CLASS FROM AUC_ST_MAPPING
DELIAUC	DELETE A REORD FROM INHERITED_ATT_USE
DELIAUK	DELETE A RECORD FROM INHERITED_ATT_USE ENTITY
DELIPDF	DELETE RECORD CLASS FROM PROJECT_DATA_FIELD
DELIRCS	DELETE RECORD CLASS FROM RC_BASED_REC_SET

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
DELISS1	DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
DELISS2	DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
DELKC	DELETE A RECORD FROM KEY_CLASS
DELKCM	DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKCMT	DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKW	DELETE A RECORD FROM AC_KEYWORD
DELKWAC	DELETE A RECORD FROM AC_KEYWORD
DELKVEC	DELETE A RECORD FROM EC_KEYWORD
DELKWRC	DELETE A RECORD FROM RC_KEYWORD
DELMDKC	DELETE ALL KEY CLASSES AND INHERITED KEYS FOR AN ENTITY
DELMDRC	DELETE ALL RELATION CLASSES FOR AN ENTITY
DELMIGK	DELETE MIGRATING KEY CLASS
DELMOD	DELETE A RECORD FROM MODEL_CLASS ENTITY
DELMTKC	DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
DELOAC	DELETE ALL OWNED AND INHERITED ATTRIBUTES
DELOACE	DELETE A RECORD FROM OWNED_ATTRIBUTE ENTITY
DELOWAC	DELETE A RECORD FROM OWNED_ATTRIBUTE
DELPCB	DELETE A RECORD FROM PSB_PCB ENTITY

MDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
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
DELRCW	DELETE A RECORD FROM RC_KEYWORD
DELRCS	DELETE RECORD CLASS FROM RC_BASED_REC_SET
DELREUS	
DELRKM1	DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRKM2	DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRKM3	DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRKY1	DELETE A RECORD FROM RECORD_KEY ENTITY
DELRKY2	DELETE A RECORD FROM RECORD_KEY ENTITY
DELRST2	DELETE A RECORD FROM RECORD_SET ENTITY
DELRST3	DELETE A RECORD FROM RECORD_SET ENTITY
DELRTY2	DELETE A RECORD FROM RECORD_TYPE ENTITY

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
DELSDF1	DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSDF2	DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSDF3	DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSEC	DELETE RECORD CLASS FROM SEC
DELSECR	DELETE RECORD CLASS FROM SEC_RC_COMPONENT
DELSN1	DELETE A RECORD FROM SCHEMA_NAMES ENTITY
DELSTM1	DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSTM2	DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSTM3	DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELTEXT	DELETE A RECORD FROM DESC_TEXT ENTITY
DELTXT	DELETE DESCRIPTION TEXT GIVEN THE OBJECT TYPE,
DEPATT	SELECT ALL THE ATTRIBUTES IN THE
DEPENT	SELECT ALL THE DEPENDANT ENTITY CLASSES
DEPFROM	GENERATE CREATE RELATION, DESCRIBE COMMANDS IN THE TO-MODEL
DEPREL	FOR EACH LEVEL OF RELATIONS IN STRUCTURE GENERATE
DESCRB	COMMAND PROCESSOR FOR THE NDDL DESCRIBE COMMAND
DLDSL2	DELETE A RECORD FROM DF_SET_LINKAGE ENTITY

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
DLMDAUC	DELETE ATTRIBUTE USE CLASSES ASSOCIATED WITH ENTITY
DLMIGRC	DELETE MIGRATING KEY CLASS
DOMUSAG	DETERMINE IF DOMAIN ASSOCIATED WITH VIEWS, DATAFIELDS, ATTRIBUTES
DPKCLST	CREATE AN KEY_CLASS_LIST TABLE CONTAINING ALL THE ENTITY
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 THE C
DRPDB	CONTROLS THE PROCESSING LOGIC FOR DELETING THE DATA BASE.
DRPDF	DELETE A RECORD FROM DATA_FIELD ENTITY
DRPDIV	DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED WITH VIEW
DRPDOM	PROGRAM NAME
DRPDT	DROP A DATA TYPE WITHIN A DOMAIN.
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".

DDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
DRPKV	DROP A KEYWORD ASSOCIATION FROM EITHER AN ATTRIBUTE, ENTITY OR RELATI
DRPKVC	OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP THEIR ASSOCIATIONS.
DRPHAP	COMMAND PROCESSOR FOR THE DROP MAP COMMAND
DRPMGKM	DELETE MIGRATED KEY CLASSES ASSOCIATED WITH A KEY CLASS
DRPMGRC	DELETE MIGRATED KEY CLASS MEMBERS ASSOCIATED WITH A KEY CLASS
DRPMIG	CONTROLS THE PROCESSING LOGIC FOR THE "DROP MIGRATES" CLAUSE.
DRPHOD	CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.
DRPRCE	DROP A RELATION CLASS FOR AN ENTITY BEING DROPPED
DRPREC	CONTROLS THE PROCESSING LOGIC FOR THE DROP RECORD COMMAND.
DRPREL	CONTROLS THE PROCESSING LOGIC FOR THE "DROP RELATION" COMMAND.
DRPSET	CONTROLS THE PROCESSING LOGIC FOR DELETING A SET FROM THE DATABASE.
DRPSMAP	DROP A SINGLE MAPPING
DRPVIEW	DROP THE VIEW.
DTUSAGE	DETERMINE ASSOCIATIONS OF A DATA TYPE
ENTKW	SEARCH FOR ENTITY KEYWORD MATCHES WITHIN TWO MODELS

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
ERRRPT	HANDLE ANY ERROR CODE FROM ORACLE.
EXCFLAG	DETERMINE IF KEYWORDS, ALIASES, DESCR ARE TO BE EXCLUDED
EXPRCLT	EXPAND RELATION LIST TO INCLUDE DEP AND IND ENTITIES
EXPRTL	EXPAND THE RETRIEVE LIST (SELECT..FROM..)
FCOPATT	GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE COMMAND
FCOPENT	DETERMINE IF COPY ENTITY WITH STRUCTURE OR RELATION
FILEINS	INSERT DESCRIPTION TEXT INTO CDM
FINDDOM	RETRIEVE A DOMAIN NUMBER FOR A GIVEN DOMAIN NAME
FMTIAUC	FORMATS A LINE FOR THE ..SET.. CLAUSE
FND1MEM	RETRIEVE RECORD TYPE OF MEMBER ASSOCIATED WITH A SET
FNDACM	DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL
FNDASA	VERIFY WHETHER A SET HAS BEEN MAPPED TO AN AUC
FNDASH	DETERMINES IF AN AUC TO SET TYPE MAPPING EXISTS
FNDAUC	DELETE ALL ATTRIBUTE USE CLASSES FOR A GIVEN ENTITY CLASS
FNDECM	DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR A GIVEN MODEL

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
FNDOAC	DELETES ALL OWNED ATTRIBUTES FOR A GIVEN ENTITY CLASS
FNDRCH	RETRIEVES ALL RC/RT MAPPINGS FOR A NAMED RECORD
FRTOREL	DETERMINE IF RELATION EXISTS BETWEEN INTRA-MOD ENTITIES
GENAKW	RETRIEVE ALL KEYWORDS FOR AN ATTRIBUTE CLASS
GENALI	GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT TYPE
GENALT1	GENERATE AN ALTER ENTITY..ADD KEY CLASS COMMAND
GENALTE	GENERATE AN ALTER ENTITY..ADD KEY CLASS COMMAND
GENATT	GENERATE A CREATE ATTRIBUTE.. COMMAND
GENDESC	GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN OBJECT TYPE AND NO
GENEKW	SELECT KEYWORD FOR ENTITY AND CREATE KEYWORD PHRASE FOR CRT ENT
GENENT	GENERATE A CREATE ENTITY..OWNED ATTRIBUTE..KEYWORD COMMAND
GENENT1	GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND DESCRS.
GENOA	SELECT OWNED ATT FOR ENTITY AND CREATES OWNED ATT FOR CRT ENT
GENREL	GENERATE CREATE RELATION ..MIGRATES..KEYWORD COMMAND

DDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
GENRKV	SELECT KEYWORDS FOR RELATION CREATES KV PHRASE FOR CREATE RC
GENRME	FORMAT THE CREATE RELATION CLAUSE
GETACAL	VERIFY THE EXISTANCE OF AN ALIAS FOR AN ATTRIBUTE
GETDBST	RETURN INFORMATION ABOUT THE CURRENT SESSIONS' DATABASE
GETDOM	RETRIEVES DOMAIN NUMBER BASED ON TAG NAME FOR AUC
GETDRT	FETCH DATA BASE, RECORD TYPE INFO FROM PARSER LISTS
GETECAL	VERIFIES THE EXISTANCE OF AN ALIAS FOR AN ENTITY
GETECNM	THIS ROUTINE SEARCHES THE UNBOUNDED EC_LIST DATA
GETECS	USING ENTITY CLASS VERIFIES CHECK MODEL RULES
GETGLOB	WILL PROVIDE GLOBAL VARIABLES
GETHAPC	SELECT ALL PROJECT DATA FIELDS MATCHING A GIVEN TAG NO
GETNCHR	GET THE NEXT CHARACTER FROM EITHER STANDARD INPUT
GETNNUM	
GETNXMO	
GETRCID	QUERIES CDM FOR INFORMATION ABOUT A RELATION CLASS

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
GETRCNM	RETRIEVE RELTION CLASS NAME, IND AND DEP ENTITY NAMES
GETRDM	RETURN WITH CURRENT SESSIONS' DATA BASE INFO
HALT	HALT WITH 'COMMIT' OR 'ROLLBACK'.
ICOPATT	INTERACTIVE COPY ATTRIBUTE, WITH KEYWORDS, ALIAS, DESCRS.
ICOPENT	INTERACTIVE COPY ENTITY WITH ATTRIBUTES, KEYWORDS, ALIAS, DESC
INDFROM	RETRIEVES RELATIONS, DETERMINES IND EC AND GENERATES NDDL
INITCMD	INITIALIZATION FOR EACH COMMAND TO BE PROCESSED
INITRDL	INITIALISE THE TABLE WHICH STORES A MODELS RELATIONS
INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
INSAC	INSERT A RECORD INTO ATTRIBUTE_CLASS
INSACNM	INSERT A RECORD INTO ATTRIBUTE_NAME
INSAREA	INSERT A RECORD INTO THE DATA_BASE AREA ENTITY. IF
INSAUC	INSERT A RECORD INTO ATTRIBUTE_USE_CL
INSAUCS	INSERT A RECORD INTO AUC_ST_MAPPING
INSCRC	INSERT A RECORD INTO COMPLETE_RELATION
INSDAA	INSERT A RECORD INTO THE DB_AREA_ASSIGNMENT ENTITY. IF

NDDL COMMAND PROCESSOR Module List

Module Name	Purpose
-----	-----
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 ENTIYTT_NAME
INSIAUC	INSERT A RECORD INTO INHERITED_ATT_USE
INSISS	INSERT A RECORD INTO THE IMS_SEGMENT_SIZE ENTITY. IF
INSKC	INSERT A RECORD INTO KEY_CLASS
INSKCM	INSERT A RECORD INTO KEY_CLASS_MEMBER
INSKV	INSERT A RECORD INTO KEYWORD
INSKWAC	INSERT A RECORD INTO AC_KEYWORD
INSKWEC	INSERT A RECORD INTO EC_KEYWORD
INSKWRC	INSERT A RECORD INTO RC_KEYWORD
INSMOD	INSERT A RECORD INTO MODEL_CLASS
INSOAC	INSERT A RECORD INTO OWNED_ATTRIBUTE

WDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
INSPCB	INSERT A RECORD INTO THE PSB_PCB ENTITY. IF
INSPDF	INSERT A RECORD INTO PROJECT_DATA_FIELD
INSPDI	INSERT A RECORD INTO PROJECT_DATA_ITEM
INSPSB	INSERT A RECORD INTO THE PSB ENTITY. IF
INSPWRD	INSERT A RECORD INTO THE DB_PASSWORD. IF SUCCESSFUL.
INSRC	INSERT A RECORD INTO RELATION_CLASS
INSRGRS	INSERT A RECORD INTO RC_BASED_REC_SET
INSREUS	INSERT A RECORD INTO REUSABLE_NUMBER
INSRKEY	INSERT A RECORD INTO THE RECORD_KEY ENTITY. IF
INSRKM	INSERT A RECORD INTO THE RECORD_KEY_MEMBER ENTITY. IF
INSRSET	INSERT A RECORD INTO THE RECORD_SET ENTITY. IF
INSRTYP	INSERT A RECORD INTO THE RECORD_TYPE ENTITY. IF
INSSCH	INSERT A RECORD INTO THE SCHEMA_NAMES ENTITY. IF
INSSDFL	INSERT A RECORD INTO THE SEGMENT_DATA_FIELD ENTITY. IF
INSSEC	INSERT A RECORD INTO SEC
INSSECR	INSERT A RECORD INTO SEC_RC_COMPONENT

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
INSSTM	INSERT A RECORD INTO THE SET_TYPE_MEMBER ENTITY. IF
KEYLOOK	RETRIVES KEYCLASS NAME AND NUMBER BASED ON TAG NO
LOADESC	LOAD DESCRIPTION FROM TEXT EDITOR
LOWUPP	CONVERT A STRING TO UPPER CASE CHARACTERS
MAPADF	MAP ON AUC TO A DATA FIELD
MAPASET	MAP AN AUC TO A SET
MAPRC	MAP A RELATION CLASS TO A SET
MIGREL	GENERATE A MIGRATES CLAUSE FOR A CREATE RELATION COMMAND
MKRNLST	FETCH LIST OF RENAME PAIRS FOR MIGRATES..SET..CLAUSE
MRGMOD	MERGE TWO IDEF MODELS INTO ONE
MRGMOD1	COPY MODEL-1 INTO A NEW MODEL (MODEL-3)
MRGMOD2	CONTROL THE LOGIC FOR PROCESSING THE REMAINING MODEL_2
MRGNODE	SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT ENTITY'S
NDDL/MAIN	MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
NEXTKC	THIS ROUTINE RETURNS A KC_NAME FOR A GIVEN EC_NO FROM
NEXTKCM	THIS ROUTINE RETURNS A KG_NO,KG_NAME FOR A GIVEN EC_NO

MDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
NRGET	
NRSTORE	STORE A AVAILABLE NO OF A POOL NO BACK TO NO LINKED
PIFROM	CREATE A VIEW USING A SINGLE ENTITY CLASS(ES-CS-MAPPING)
PDFDB	RETRIEVES PROJECT DATA FIELDS FOR A GIVEN DATA BASE
PDFDF	RETRIVES ALL THE PROJ DATA FIELD OR DF BEING DROPPED
PDFREC	RETRIEVES ALL PORJECT DATA FIELDS FOR A GIVEN RECORD
PDFSRCH	DETERMINES IF A TAG HAS A PRIMARY MAP TO DATA FIELD
PMFROM	CREATE A VIEW FROM MULTIPLE ENTITY CLASSES
PNOFROM	PROCESS A VIEW COMMAND FOR MULTIPLE ENTITY CLASSES
PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
PROCDT	PROGRAM NAME
RACKW2	COMPARES ATTRIBUTE CLASS KEYWORDS
RCCHK	CHECK IF A RELATION CLASS EXISTS FOR A GIVEN MODEL
RCCHK1	CHECK IF A RELATION CLASS EXISTS FOR A GIVEN MODEL
RDDESC	STORE DESCRIPTION ON THE CDM
RECKW2	COMPARES ENTITY CLASS KEYWORDS BETWEEN MODELS

DDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
RELKW	CHECKS RC KW BETWEEN MODELS
REVIEW	POPULATE VIEW DEFINITION AND ES-CS-MAPPING
RENAME	UPDATE EXISTING OBJECT NAME WITH NEW OBJECT NAME
RETACKW	POPULATES KEYWORD TABLE FOR ATTRIBUTE CLASS KEYWORDS
RETECKW	POPULATES KEYWORD TABLE FOR ENTITY CLASS KEYWORDS
RETRAC1	DETERMINES IF ANY ATTRIBUTE CLASS NAMES MATCH BETWEEN MODELS
RETRACP	SELECT ATTRIBUTE CLASS NAME USING THE ATTRIBUTE CLASS NUMBER
RETRCKW	POPULATES THE KEYWORD CDM TABLE FOR RELATION CLASS KEYWORDS
RETREC1	COMPARES ENTITY CLASS NAMES FOR TWO MODELS
RETRECP	RETRIEVE AN ENTITY CLASS NAME FOR A GIVEN NO. AND NAME TYPE
ROLBACK	ROLBACK THE TRANSACTIONS.
RRCKW2	COMPARES RELATION CLASS KEY WORDS FOR TWO MODELS
SELACNM	RETRIEVE THE PRIMARY NAME FOR ATTRIBUTE CLASS
SELECNM	RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS
SELIAUC	SELECTS ALL THE INHERITED TAG NAMES

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
SELIKEY	RETRIEVE ALL INHERITED KEYS FOR A GIVEN EC AND RC
SELECNM	RETRIEVE ALL RELATION CLASSES FOR WHICH AN ENTITY IS PART OF
SELRSET	DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET BEING DROPPED
SELSTM	DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE BEING DROPPED
SMVIEW	CREATE ES-CS-MAPPING FOR USER SPECIFIED RETRIEVE LIST
STRINS	INSERT DESCRIPTION TEXT INTO THE CDM
TERMSER	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
TXTTYP	DETERMINE SOURCE OF DESCRIPTION TEXT
UERROR	ISSUE A MESSAGE TO THE USER, CONSIDERED A
UPDAC	UPDATE ATTRIBUTE_CLASS SET DOMAIN_NO = : 1
UPDACAL	UPDATE MODEL_CLASS SET AC_NAME_TYPE = : 1
UPDACNM	UPDATE ATTRIBUTE_NAME SET AC_NAME = : 1
UPDECAL	UPDATE MODEL_CLASS SET EC_NAME_TYPE = : 1
UPDECNM	UPDATE MODEL_CLASS SET ENTITY_NAME = : 1
UPDIND	UPDATE USER_DEF_DATA_TYPE SET DATA_TYPE_IND = : 1

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
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
UWARN	ISSUE A MESSAGE TO THE USER, CONSIDERED A
VERACDT	DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH ATTRIBUTE CLASSES
VERACNM	RETRIEVE THE DOMAIN FOR AN ATTRIBUTE CLASS
VERALI	VERIFY THE EXISTENCE OF AN ALIAS NAME FOR AN ENTITY
VERAREA	VERIFY THE EXISTENCE OF AN AREA FOR A DATA BASE
VERASH	DETERMINE IF ANY AUC/SET MAPPINGS EXIST FOR AN AUC
VERATT	VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A MODEL
VERAUC	VERIFY THE EXISTENCE OF AN ATTRIBUTE USE CLASS

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
VERCRC	DETERMINE IF A KEY CLASS IS PART OF A COMPLETE RELATION
VERDB	VERIFY THE EXISTENCE OF A DATA BASE IN THE CDM
VERDBAS	VERIFY A DATA BASE FOR A GIVEN NAME AND HOST
VERDF	VERIFY THE EXISTENCE OF A DATA FIELD
VERDFDT	DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH A PROJECT DF
VERDFLD	VERIFIES THE EXISTENCE OF A DATA FIELD FOR A GIVEN DATA BASE
VERDI	VERIFY THE EXISTENCE OF A DATA ITEM IN A GIVEN VIEW
VERDIDT	DETERMINE IF A DATA TYPE IS ASSOCIATED WITH ANY DATA ITEMS
VERDOM	VERIFY THE EXISTENCE OF A DOMAIN CLASS IN THE CDM
VERDSL3	RETRIEVES SET ID FOR TOTAL DATA BASE
VERDSTP	VALIDATE A USER ENTERED DESCRIPTION TYPE
VERDSTX	VERIFY THE EXISTENCE OF DESC. TEXT FOR A GIVEN OBJECT
VERDT	VERIFY THE EXISTENCE OF A USER DEFINED DATA TYPE
VERDTD	VERIFY THE EXISTENCE OF A DATA TYPE IN A GIVEN DOMAIN
VERENT	VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A MODEL

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
VERKC	VERIFY THE EXISTENCE OF A KEY CLASS IN A MODEL
VERKW	verify the existence of a keyword.
VERKWE	VERIFY THE EXISTENCE OF ENTITY CLASS KEYWORD
VERKWR	VERIFIES THE EXISTANCE OF A RELATION CLASS KEYWORD.
VERMOD	verify the existence of a model and return the unique number.
VERNMA	verify the existence of an attribute class in a model.
VERNME	verify the existence of an entity class in a model.
VEROAC	verify the existence of an owned attribute class for an entity.
VEROBJ	VERIFY THAT THE OBJECT EXISTS.
VERPDF	verify the existence of a project_data_field occurrence.
VERPSB	VERIFIES THE EXISTANCE OF A PROGRAM STATUS BLOCK FOR AN IMS DATABASE
VERRC	verify the existence of a relation class.
VERRCBS	verify if there are any mapping to a set found in RC_BASED_REC_SET.
VERRCC	verify if the relation class is complete.
VERRCMP	verify whether a member has been mapped to a relation class.

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
VERRCNM	VERIFY EXISTANCE OF A RELATION IN A MODEL
VERRCST	REPORT WHETHER A RC TO SET MAP EXISTS.
VERREL	VERIFY THE DEPENDENT AND INDEPENDENT ENTITIES IN THE RELATION
VERRELS	VERIFY THE RELATION CLASS STRUCTURE
VERRK	verifies the existance of a key for a record.
VERRKM	verifies the existance of a data field for a record.
VERRSET	VERIFIES THE EXISTANCE OF A SET.
VERRT	VERIFIES THE EXISTANCE OF A RECORD TYPE.
VERSDT	For a given domain number, return its standard data type name.
VERSMS	fetch the member count for a given set.
VERTYP	verifies that a type identification for a data type is valid.
VERUDTN	verify the existence of a USER DEFINED DATA TYPE (USERDATATYPE)/USDF
VERVIEW	verify the existence of a view in the system.
VOMAPS	ENFORCES AUC TO SET MAPPING RULES.
WRTACKW	RETRIEVE AND WRITE ALL THE KEYWORDS ASSOCIATD WITH AN ATTRIBUTE CLAS
WRTALI	RETRIEVE ALL ALIAS NAMES OF AN ATTRIBUTE CLASS.

NDDL COMMAND PROCESSOR Module List

Module Name -----	Purpose -----
WRTANAM	RETRIEVE AND COPY ALL THE NAMES (PRIMARY AND ALIAS) OF AN ATTRIBUTE.
WRTDESC	SELECT A RECORD FROM DESC_TEXT ENITY
WRTDSC4	SELECT A RECORD FROM DESC_TEXT ENITY
WRTECKW	RETRIEVE AND COPY ALL THE KEYWORDS ASSOCIATED WITH AN ENTITY CLASS.
WRTENAM	RETRIEVE AND COPY ALL THE ALIAS NAMES OF AN ENTITY CLASS.
WRTLIN	THIS ROUTINE WRITES A NDDL COMMAND LINE (80 CHARACTERS)
YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141100
1 November 1985

3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. The specification in which any module is documented may be found in the Module Documentation Index (Document Number CM 620100001). See section 3.10.6 for a list of the modules that call each of these external routines.

NDDL COMMAND PROCESSOR External Routines List

Module Name	First User
-----	-----
ADDFRM	INITSES
ALLOC	NRSTORE
ERRPRO	ICOPATT
EXIT	NDDL/MAIN
FCLOSE	GLSFIL
FREE	COMMIT
GDATA	PRCCMD
GETCHAR	GETNCHR
INITEX	INITSES
INITFP	INITSES
LOGOFF	TERMSES
LOGON	INITSES
OBINDN	FILEINS
OCLOSE	COCLOSE
OCOF	COCOF
OCOM	COCOM
ODFINN	CODFINN
OERMSG	COERMSG
OEXEC	FILEINS
OFETCH	COFETCH
OISCR	PRCCMD
OLOGOF	COLOGOF
OLON	COLON
OOPEN	COOPEN
OPNFIL	CESTRUC
OPNFRM	INITSES
OROL	COROL
OSQL3	COSQL3
PDATA	PRCCMD
PMSGLS	PRCCMD
PRINTF	VEROBJ
PUTC	WRTLIN
RPLFRM	PRCCMD
SPRINTF	CPYDES
STRCAT	ADD_CORR
STRLEN	ADD_CORR
STRNCMP	ALLENT
STRNCPY	CPFNXT
TERMFP	TERMSES
TOLOWER	ADDMAP
TOUPPER	LOWUPP

PS 620141100
1 November 1985

NDL COMMAND PROCESSOR External Routines List

Module Name -----	First User -----
TRNDML	TERNSES
YLEX	YYPARSE

3.10.4 Include File List

The following is a list of all include files called in by modules being documented here. Each include file has a unique name regardless of the language being used. The purpose of each include file is listed as well. A more complete description of each include file is given in section 3.10.9. The purpose listed is the one that is in the source code of the include file.

A purpose of ***** PURPOSE NOT FOUND BY STRIPPER ***** indicates that a purpose statement was not written into the include file itself. The most common reason for this is that the include file comes from system libraries that were not developed by the project, such as 'C' libraries that are provided with the 'C' compiler.

See section 3.10.6 for a set of lists which show all the modules which call in each of these include files.

NDDL COMMAND PROCESSOR Include File List

File Name -----	Purpose -----
CHKMODL	DETERMINE IF CURRENT MODEL EXISTS FOR A SESSION
CMDS	COMMAND NO. FOR EACH NDDL/NDML COMMAND
ECLIST	CONTAINS A LIST OF ENTITY CLASS NUMBERS
ERRPRO	PROCESS ERROR INCLUDE FILE
FPCODE	FORM PROCESSOR RETURN CODES
FPPARM	FORM PROCESSOR PARAMETERS
KCLIST	PROCESS ERROR INCLUDE FILE
KEYLIST	DATA STRUCTURE FOR NDDL MODELLING COMMANDS
KWDTBL	KEYWORD TABLE
LISTID	PROVIDES LIST OF PARSED OBJECTS
LISTNOS	VALID LIST NUMBERS
LISTREL	LIST OF RELATIONS
LISTS	PROVIDES THE DIMENSIONS OF THE NDDL LISTS
NDDL	**** PURPOSE NOT FOUND BY STRIPPER ****
NDDLEX	**** PURPOSE NOT FOUND BY STRIPPER ****
NDDLAC.INP"	**** PURPOSE NOT FOUND BY STRIPPER ****
OK	GOOD RETURN CODE VALUE FOR UI
ORCLEDA	WS DEFINITION FOR THE ORACLE LOGIN AREA
RCDEPKC	LIST OF KEYS MIGRATED VIA A RELATION
RELTBL	LIST OF RELATION CLASSES IN A MODEL
RENLIST	LIST OF ATTRIBUTES AND INHERITED TAG PAIRS
SDLIST	SEC-DECOMPOSITION-LIST
SRVRET	AS THE RETURN GIVEN A TABLE-FULL ERROR
STDIO	**** PURPOSE NOT FOUND BY STRIPPER ****
STDYTP	STANDARD TYPE DEFINITIONS
UNIQENO	UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS
VWDI	LIST OF DATA ITEMS IN A VIEW
VWFROM	LIST OF ENTITIES SPECIFIED IN A VIEW
VWRC	LIST OF RELATION CLASSES INHERENT TO A VIEW
VWRETR	LIST OF ENTITIES AND TAGS SPECIFIED IN A VIEW

PS 620141100
1 November 1985

3.10.5 Where Include File Used List

The following lists each include file from 3.10.4 and all the modules documented in this specification which include them. The purpose of each module is listed as well.

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
CHKMODL		
	ALTENT	CONTROL PROCESSING FOR ALTER ENTITY CLASS COMMAND.
	ALTREL	CONTROLS PROCESSING LOGIC FOR THE ALTER RELATION COMMAND
	CMBENT	CONROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.
	COPATT	CONTROLS THE PROCESSING LOGIC FOR COPYING AN ATTRIBUTE.
	COPENT	CONTROLS THE PROCESSING LOGIC FOR THE COPY ENTITY COMMAND.
	CPYMOD	CONTROLS THE PROCESSING LOGIC FOR THE COPY MODEL COMMAND.
	CRTATT	CONTROLS THE PROCESSING LOGIC FOR CREATING AN ATTRIBUTE.
	CRTENT	CONTROL THE PROCESSING LOGIC FOR CREATING A NEW ENTITY CLASS.
	CRTREL	CONTROLS THE LOGIC FOR VALIDATING AND CREATING A NEW RELATION CLASS.

CMDS

BRANCHR	PERFORMS MULTI-WAY CALL TO THE
ERRRPT	HANDLE ANY ERROR CODE FROM ORACLE,
NDDL/MAIN	MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
UERROR	ISSUE A MESSAGE TO THE USER, CONSIDERED A
UWARN	ISSUE A MESSAGE TO THE USER, CONSIDERED A
YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

ECLIST

ALLKEY	GENERATES KEY CLASS FOR AN ENTITY
---------------	--

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	CHKOWN	checks each of the OWN-ec's key classes against the RC-DEPKC LIST.
	CKDUPEC	POPULATES A TABLE WITH ENTITY NAMES.
	MRGMOD	MERGE TWO IDEF MODELS INTO ONE
ERRPRO		
	ADDKWA	INSERT AN ATTRIBUTE KEYWORD
	ADDKWE	INSERT AN ENTITY KEYWORD
	ADDKWR	INSERT A RELATION KEYWORD
	ADDMIG	PROCESS THE ADD MIGRATES...SET..CLAUSE
	ADDOAC	ADD ATTRIBUTE AS AN OWNED ATTRIBUTE AND AS ATT USE CLASS
	ALLKEY	GENERATES KEY CLASS FOR AN ENTITY
	ALLVIEW	CREATE AN ES-CS-MAPPING FOR AUC TO DATA ITEM
	ATTKW	CONTROLS PROCESSING TO POPULATE KEYWORD TABLE FOR AUC KEYWORD
	BLKCL1	STORE ALL KEY CLASS INFO FOR AN ENTITY IN A LIST
	BLKCLST	SELECT AND STORE KEY CLASS INFO FOR A GIVEN ENTITY
	CDP4A	VERIFY SURROGATE ENTITY CLASS STRUCTURE
	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 STRUCTURE
	CHKATT	CHECK IF ATTRIBUTES HAVE BEEN CREATED ACCORDING TO STANDARDS
	CHKAUCV	CHECK EXISTENCE OF AUC TO SET MAPPING
	CHKKEYS	CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
	CHKMOD	DETERMINES WHETHER CERTAIN RULES ARE FULFILLED.
	CHKREL	CHECK IF RELATIONS HAVE BEEN CREATED ACCORDING TO STANDARDS
	CMBACAL	GENERATE CREATE ALIAS ATTRIBUTE.. AND ALIAS DESC TEXT COMMANDS

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	CMBALI	GENERATE CREATE ALIAS ENTITY. COMMAND
	CMBEKV	GENERATE ADD KEYWORD CLAUSE FOR ENTITY KEYWORDS
	CMBENT	CONROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.
	CMBOA	GENERATE COMMANDS FOR ATTRIBUTES, ITS KEYWORDS, ALIAS, DESC
	CMBRKW	SELECT AND GENERATE RELATION CLASS KEYWORDS
	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.
	COPYAC	CREATE AN ATTRIBUTE, ASSOCIATE WITH ENTITY, ADD KEY CLASSES
	CRTMOD	CONTROL THE PROCESSING LOGIC IF CREATING A MODEL WITHIN THE SYSTEM.
	DEFAREA	PROCESSES THE AREA INFORMATION IF THE DBMS IS IDS-II, IDMS AND VAX-1
	DEFCODL	PROCESSES THE DBMS TYPES: VAX-11, IDMS, IDS-II.
	DEFDB	CONTROLS THE PROCESSING LOGIC FOR DEFINING A DATABASE TO THE SYSTEM.
	DEFPLD	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.
	DEFREC	CONTROLS THE PROCESSING LOGIC FOR DEFINING A RECORD FOR THE SYSTEM.
	DEFTOT	PROCESSES THE DBMS TYPE: TOTAL.
	DELDBDF	DELETE ALL DATA FIELD ASSOCIATIONS WITH THE DATABASE
	DELDERT	DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE DATABASE

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	DELDBST	DELETE ALL RECORD SETS' ASSOCIATIONS WITH THE DATABASE
	DELDTNO	DELETE ALL DATA TYPES ASSOCIATED WITH A DOMAIN
	DELMDKC	DELETE ALL KEY CLASSES AND INHERITED KEYS FOR AN ENTITY
	DELMDRC	DELETE ALL RELATION CLASSES FOR AN ENTITY
	DELOAC	DELETE ALL OWNED AND INHERITED ATTRIBUTES
	DEPFROM	GENERATE CREATE RELATION, DESCRIBE COMMANDS IN THE TO-MODEL
	DLMDAUC	DELETE ATTRIBUTE USE CLASSES ASSOCIATED WITH ENTITY
	DOMUSAG	DETERMINE IF DOMAIN ASSOCIATED WITH VIEWS, DATAFIELDS, ATTRIBUTES
	DRPAC	DELETE OWNED ATTRIBUTES ASSOCIATED WITH ENTITY
	DRPATT	CONTROLS THE DROPPING OF USER SPECIFIED ATTRIBUTE CLASSES FROM THE C
	DRPDB	CONTROLS THE PROCESSING LOGIC FOR DELETING THE DATA BASE.
	DRPDIV	DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED WITH VIEW
	DRPFLD	CONTROLS THE PROCESSING LOGIC FOR DROPPING A DATA FIELD.
	DRPKC	CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY CLASS".
	DRPMGKM	DELETE MIGRATED KEY CLASSES ASSOCIATED WITH A KEY CLASS
	DRPMGRC	DELETE MIGRATED KEY CLASS MEMBERS ASSOCIATED WITH A KEY CLASS
	DRPMOD	CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.
	DRPRCE	DROP A RELATION CLASS FOR AN ENTITY BEING DROPPED
	DRPSET	CONTROLS THE PROCESSING LOGIC FOR DELETING A SET FROM THE DATABASE.
	ENTKW	SEARCH FOR ENTITY KEYWORD MATCHES WITHIN TWO MODELS
	FCOPATT	GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE COMMAND

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	FCOPENT	DETERMINE IF COPY ENTITY WITH STRUCTURE OR RELATION
	FILEINS	INSERT DESCRIPTION TEXT INTO CDM
	FINDDOM	RETRIEVE A DOMAIN NUMBER FOR A GIVEN DOMAIN NAME
	FND1MEM	RETRIEVE RECORD TYPE OF MEMBER ASSOCIATED WITH A SET
	FNDACM	DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL
	FNDASA	VERIFY WHETHER A SET HAS BEEN MAPPED TO AN AUC
	FNDASH	DETERMINES IF AN AUC TO SET TYPE MAPPING EXISTS
	FNDAUC	DELETE ALL ATTRIBUTE USE CLASSES FOR A GIVEN ENTITY CLASS
	FNDECM	DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR A GIVEN MODEL
	FNDOAC	DELETES ALL OWNED ATTRIBUTES FOR A GIVEN ENTITY CLASS
	FNDRCM	RETRIEVES ALL RC/RT MAPPINGS FOR A NAMED RECORD
	FRTOREL	DETERMINE IF RELATION EXISTS BETWEEN INTRA-MOD ENTITIES
	GENAKW	RETRIEVE ALL KEYWORDS FOR AN ATTRIBUTE CLASS
	GENDESC	GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN OBJECT TYPE AND NO
	GENEKW	SELECT KEYWORD FOR ENTITY AND CREATE KEYWORD PHRASE FOR CRT ENT
	GENENT1	GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND DESCRS.
	GENOA	SELECT OWNED ATT FOR ENTITY AND CREATES OWNED ATT FOR CRT ENT
	GENRKW	SELECT KEYWORDS FOR RELATION CREATES KW PHRASE FOR CREATE RC
	GETACAL	VERIFY THE EXISTANCE OF AN ALIAS FOR AN ATTRIBUTE
	GETDBST	RETURN INFORMATION ABOUT THE CURRENT SESSIONS' DATABASE

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	GETDOM	RETRIEVES DOMAIN NUMBER BASED ON TAG NAME FOR AUC
	GETECAL	VERIFIES THE EXISTANCE OF AN ALIAS FOR AN ENTITY
	GETECS	USING ENTITY CLASS VERIFIES CHECK MODEL RULES
	GETMAPC	SELECT ALL PROJECT DATA FIELDS MATCHING A GIVEN TAG NO
	GETRCID	QUERIES CDM FOR INFORMATION ABOUT A RELATION CLASS
	GETRCNM	RETRIEVE RELTION CLASS NAME, IND AND DEP ENTITY NAMES
	GETRDH	RETURN WITH CURRENT SESSIONS' DATA BASE INFO
	ICOPATT	INTERACTIVE COPY ATTRIBUTE, WITH KEYWORDS, ALIAS, DESCRS.
	ICOPENT	INTERACTIVE COPY ENTITY WITH ATTRIBUTES, KEYWORDS, ALIAS, DESC
	INDFROM	RETRIEVES RELATIONS, DETERMINES IND EC AND GENERATES NDDL
	KEYLOOK	RETRIVES KEYCLASS NAME AND NUMBER BASED ON TAG NO
	MRGMOD	MERGE TWO IDEF MODELS INTO ONE
	MRGMOD1	COPY MODEL-1 INTO A NEW MODEL (MODEL-3)
	PDFDB	RETRIEVES PROJECT DATA FIELDS FOR A GIVEN DATA BASE
	PDFDF	RETRIVES ALL THE PROJ DATA FIELD OR DF BEING DROPPED
	PDFREC	RETRIEVES ALL PORJECT DATA FIELDS FOR A GIVEN RECORD
	PDFSRCH	DETERMINES IF A TAG HAS A PRIMARY MAP TO DATA FIELD
	RACKW2	COMPARES ATTRIBUTE CLASS KEYWORDS
	RECKW2	COMPARES ENTITY CLASS KEYWORDS BETWEEN MODELS
	RELKW	CHECKS RC KW BETWEEN MODELS
	RETACKW	POPULATES KEYWORD TABLE FOR ATTRIBUTE CLASS KEYWORDS
	RETECKW	POPULATES KEYWORD TABLE FOR ENTITY CLASS KEYWORDS

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	RETRAC1	DETERMINES IF ANY ATTRIBUTE CLASS NAMES MATCH BETWEEN MODELS
	RETRACP	SELECT ATTRIBUTE CLASS NAME USING THE ATTRIBUTE CLASS NUMBER
	RETRCKW	POPULATES THE KEYWORD CDM TABLE FOR RELATION CLASS KEYWORDS
	RETREC1	COMPARES ENTITY CLASS NAMES FOR TWO MODELS
	RETRECP	RETRIEVE AN ENTITY CLASS NAME FOR A GIVEN NO. AND NAME TYPE
	RRCKW2	COMPARES RELATION CLASS KEY WORDS FOR TWO MODELS
	SELACNM	RETRIEVE THE PRIMARY NAME FOR ATTRIBUTE CLASS
	SELECNM	RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS
	SELIKEY	RETRIEVE ALL INHERITED KEYS FOR A GIVEN EC AND RC
	SELRCNM	RETRIEVE ALL RELATION CLASSES FOR WHICH AN ENTITY IS PART OF
	SELRSET	DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET BEING DROPPED
	SELSTM	DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE BEING DROPPED
	STRINS	INSERT DESCRIPTION TEXT INTO THE CDM
	VERACDT	DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH ATTRIBUTE CLASSES
	VERACNM	RETRIEVE THE DOMAIN FOR AN ATTRIBUTE CLASS
	VERALI	VERIFY THE EXISTENCE OF AN ALIAS NAME FOR AN ENTITY
	VERAREA	VERIFY THE EXISTENCE OF AN AREA FOR A DATA BASE
	VERASH	DETERMINE IF ANY AUC/SET MAPPINGS EXIST FOR AN AUC
	VERATT	VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A MODEL
	VERAUC	VERIFY THE EXISTENCE OF AN ATTRIBUTE USE CLASS
	VERCRC	DETERMINE IF A KEY CLASS IS PART OF A COMPLETE RELATION

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	VERDB	VERIFY THE EXISTENCE OF A DATA BASE IN THE CDM
	VERDBAS	VERIFY A DATA BASE FOR A GIVEN NAME AND HOST
	VERDF	VERIFY THE EXISTENCE OF A DATA FIELD
	VERDFDT	DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH A PROJECT DF
	VERDFLD	VERIFIES THE EXISTENCE OF A DATA FIELD FOR A GIVEN DATA BASE
	VERDI	VERIFY THE EXISTENCE OF A DATA ITEM IN A GIVEN VIEW
	VERDIDT	DETERMINE IF A DATA TYPE IS ASSOCIATED WITH ANY DATA ITEMS
	VERDOM	VERIFY THE EXISTENCE OF A DOMAIN CLASS IN THE CDM
	VERDSL3	RETRIEVES SET ID FOR TOTAL DATA BASE
	VERDSTP	VALIDATE A USER ENTERED DESCRIPTION TYPE
	VERDSTX	VERIFY THE EXISTENCE OF DESC. TEXT FOR A GIVEN OBJECT
	VERDT	VERIFY THE EXISTENCE OF A USER DEFINED DATA TYPE
	VERDTD	VERIFY THE EXISTENCE OF A DATA TYPE IN A GIVEN DOMAIN
	VERENT	VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A MODEL
	VERKC	VERIFY THE EXISTENCE OF A KEY CLASS IN A MODEL
	VERKW	verify the existencce of a keyword.
	VERKWE	VERIFY THE EXISTENCE OF ENTITY CLASS KEYWORD
	VERKWR	VERIFIES THE EXISTANCE OF A RELATION CLASS KEYWORD.
	VERMOD	verify the existence of a model and return the unique number.
	VERNMA	verify the existence of an attribute class in a model.
	VERNME	verify the existence of an entity class in a model.
	VEROAC	verify the existence of an owned attribute class for an entity.

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	VERPDF	verify the existence of a project_data_field occurrence.
	VERPSB	VERIFIES THE EXISTANCE OF A PROGRAM STATUS BLOCK FOR AN IMS DATABASE
	VERRC	verify the existence of a relation class.
	VERRCBS	verify if there are any mapping to a set found in RC BASED REC SET.
	VERRCC	verify if the relation class is complete.
	VERRCMP	verify whether a member has been mapped to a relation class.
	VERRCNM	VERIFY EXISTANCE OF A RELATION IN A MODEL
	VERRCST	REPORT WHETHER A RC TO SET MAP EXISTS.
	VERRK	verifies the existence of a key for a record.
	VERRKM	verifies the existence of a data field for a record.
	VERRSET	VERIFIES THE EXISTANCE OF A SET.
	VERRT	VERIFIES THE EXISTANCE OF A RECORD TYPE.
	VERSDT	For a given domain number, return its standard data type name.
	VERSMS	fetch the member count for a given set.
	VERTYP	verifies that a type identification for a data type is valid.
	VERUDTN	verify the existence of a USER DEFINED DATA TYPE (USERDATATYPE)/USDF
	VERVIEW	verify the existence of a view in the system.
	VOMAPS	ENFORCES AUC TO SET MAPPING RULES.
	WRTACKW	RETRIEVE AND WRITE ALL THE KEYWORDS ASSOCIATED WITH AN ATTRIBUTE CLASS
	WRTALI	RETRIEVE ALL ALIAS NAMES OF AN ATTRIBUTE CLASS.
	WRTANAM	RETRIEVE AND COPY ALL THE NAMES (PRIMARY AND ALIAS) OF AN ATTRIBUTE.
	WRTECKW	RETRIEVE AND COPY ALL THE KEYWORDS ASSOCIATED WITH AN ENTITY CLASS.
	WRTENAM	RETRIEVE AND COPY ALL THE ALIAS NAMES OF AN ENTITY CLASS.

MODEL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
FPCODE	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
	PROCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
FPPARM	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
	PROCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
ECLIST	COPTAC	CREATE AN ATTRIBUTE, ASSOCIATE WITH ENTITY, ADD KEY CLASSES
	KEYLOOK	RETRIVES KEYCLASS NAME AND NUMBER BASED ON TAG NO
KEYLIST	ADDBC	ADD THE ENTITY NAME TO THE TREE LIST STRUCTURE
	ADDBCNM	ADD THE BC NAME AND BC NO INTO KEYLIST
	ADDECLS	ADD KEY INFO TO THE UNBOUNDED KEY_CLASS_LIST STRUCTURE
	ADDEG	ADD ECH TAG NUMBER AND NAME TO STRUCTURE
	ADDBCEC	POPULATES THE BC-DEPEC TABLE FOR ALL RELATIONS IN THE MODEL
	AKCROW	THIS ROUTINE ADDS A ROW TO THE UNBOUNDED KEY_CLASS_LIST

DDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	ALKEY	GENERATES KEY CLASS FOR AN ENTITY
	BLECLST	THIS ROUTINE CREATES A ROW IN THE UNBOUNDED EC LIST
	CHKINH	checks the RC-DEPKC table and determines when a key class can be add
	CHKOWN	checks each of the OWN-ec's key classes against the RC-DEPKC LIST.
	GENALT1	GENERATE AN ALTER ENTITY..ADD KEY CLASS COMMAND
	GETECN	THIS ROUTINE SEARCHES THE UNBOUNDED EC LIST DATA
	HEKTEC	THIS ROUTINE RETURNS A EC_NAME FOR A GIVEN EC NO FROM
	HEKTECN	THIS ROUTINE RETURNS A EC_NO, EC_NAME FOR A GIVEN EC_NO
	SELIKEY	RETRIEVE ALL INHERITED KEYS FOR A GIVEN EC AND RC
KVDTBL		
	ATTKV	CONTROLS PROCESSING TO POPULATE KEYWORD TABLE FOR AUC KEYWORD
	ENTKV	SEARCH FOR ENTITY KEYWORD MATCHES WITHIN TWO MODELS
	RACKV2	COMPARES ATTRIBUTE CLASS KEYWORDS
	RECKV2	COMPARES ENTITY CLASS KEYWORDS BETWEEN MODELS
	RELV	CHECKS RC KV BETWEEN MODELS
	RETACKV	POPULATES KEYWORD TABLE FOR ATTRIBUTE CLASS KEYWORDS
	RETECKV	POPULATES KEYWORD TABLE FOR ENTITY CLASS KEYWORDS
	RETRCKV	POPULATES THE KEYWORD CDM TABLE FOR RELATION CLASS KEYWORDS
	RACKV2	COMPARES RELATION CLASS KEY WORDS FOR TWO MODELS

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----

LISTID

ADDMAP	ADD A CS-IS MAPPING
ALTATT	THE ALTER ATTRIBUTE COMMAND PROCESSOR ALTERS
ALTMAP	ALTER MAP COMMAND PROCESSOR
ALTMAP	ALTER A SINGLE MAP
CHGDOM	CHANGE THE DOMAIN FOR AN ALTERED ATTRIBUTE COMMAND PROCESSOR FOR COPY DESCRIPTION
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
DRPMAP	DROP A SINGLE MAPPING
HALT	HALT WITH 'COMMIT' OR 'ROLLBACK'.
TXTTY	DETERMINE SOURCE OF DESCRIPTION TEXT
VEROBJ	VERIFY THAT THE OBJECT EXISTS.
YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

LISTNOS

ADDATT	ASSOCIATES EXISTING ATT WITH ENTITY IN CREAT ENTITY COMMAND
ADDEK	CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR CREATE/ALTER ENTITY
ADDEK	ADDS KEYWORDS FOR COMMANDS USING "ADD KEYWORD"(OPTIONAL)
ADDNIG	PROCESS THE ADD MIGRATES...SET...CLAUSE
ALTAI	CONTROLS ALTER ALIAS PROCESSING (ALIAS TO PRIM OR VICE VERSA)
ALTCARD	PROCESS CARDINALITY FO USER SPECIFIED RELATION
ALTDOM	PROGRAM NAME

DDDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	ALTENT	CONTROL PROCESSING FOR ALTER ENTITY CLASS COMMAND.
	ALTMOD	CONTROLS PROCESSING FOR ALTER MODEL COMMANDS
	ALTREL	CONTROLS PROCESSING LOGIC FOR THE ALTER RELATION COMMAND
	BLVWLST	CREATE BUILD VIEW LISTS FOR THE CREATE VIEW COMMAND
	CHKCARD	OBTAINS THE USER-SPECIFIED CARDINALTY (IF ANY) FOR THE RELATION.
	CHKMOD	DETERMINES WHETHER CERTAIN RULES ARE FULFILLED.
	CMBENT	CONROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.
	CMPMOD	CONTROLS THE PROCESSING LOGIC TO COMPARE TWO MODELS.
	COPATT	CONTROLS THE PROCESSING LOGIC FOR COPYING AN ATTRIBUTE.
	COPENT	CONTROLS THE PROCESSING LOGIC FOR THE COPY ENTITY COMMAND.
	CPYMOD	CONTROLS THE PROCESSING LOGIC FOR THE COPY MODEL COMMAND.
	CRTALI	CREATES ALIASES FOR AN ENTITY OR ATTRIBUTE.
	CRTATT	CONTROLS THE PROCESSING LOGIC FOR CREATING AN ATTRIBUTE.
	CRTDOM	PROGRAM NAME CRTDOM
	CRTEXT	CONTROL THE PROCESSING LOGIC FOR CREATING A NEW ENTITY CLASS.
	CRTHOD	CONTROL THE PROCESSING LOGIC IF CREATING A MODEL WITHIN THE SYSTEM.
	CRTREL	CONTROLS THE LOGIC FOR VALIDATING AND CREATING A NEW RELATION CLASS.
	CRTVIEW	CONTROLS THE PROCESSING LOGIC FOR THE CREATE VIEW COMMAND.
	DEFAREA	PROCESSES THE AREA INFORMATION IF THE DBMS IS IDS-II, IDMS AND VAX-1
	DEFCODL	PROCESSES THE DBMS TYPES: VAX-11, IDMS, IDS-II.

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	DEFDB	CONTROLS THE PROCESSING LOGIC FOR DEFINING A DATABASE TO THE SYSTEM.
	DEFPLD	PROCESSES THE FIELD INFORMATION FOR THE DEFINE RECORD COMMAND.
	DEFINS	PROCESSES THE DBMS TYPE: IMS.
	DEFINSS	PROCESSES THE SEGMENT INFORMATION IF THE DBMS IS IMS.
	DEFKEY	PROCESSES THE KEY INFORMATION FOR THE DEFINE RECORD COMMAND.
	DEPORCL	PROCESSES THE DBMS TYPE: ORACLE.
	DEPREC	CONTROLS THE PROCESSING LOGIC FOR DEFINING A RECORD FOR THE SYSTEM.
	DEFSET	CONTROLS THE PROCESSING LOGIC FOR THE DEFINE SET COMMAND.
	DEPTOT	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 THE C
	DRPDB	CONTROLS THE PROCESSING LOGIC FOR DELETING THE DATA BASE.
	DRPDOH	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".
	DRPKV	DROP A KEYWORD ASSOCIATION FROM EITHER AN ATTRIBUTE, ENTITY OR RELATI
	DRPKVC	OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP THEIR ASSOCIATIONS.
	DRPNIG	CONTROLS THE PROCESSING LOGIC FOR THE "DROP MIGRATES" CLAUSE.
	DRPMOD	CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.
	DRPREC	CONTROLS THE PROCESSING LOGIC FOR THE DROP RECORD COMMAND.

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	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 LISTS
	GETRDH	RETURN WITH CURRENT SESSIONS' DATA BASE INFO
	MKRNLST	FETCH LIST OF RENAME PAIRS FOR MIGRATES..SET..CLAUSE
	MRCMOD	MERGE TWO IDEF MODELS INTO ONE
	PROCDT	PROGRAM NAME
	ROCNEK	CHECK IF A RELATION CLASS EXISTS FOR A GIVEN MODEL
	ROCNEK1	CHECK IF A RELATION CLASS EXISTS FOR A GIVEN MODEL
	RENAME	UPDATE EXISTING OBJECT NAME WITH NEW OBJECT NAME
LISTREL		
	FRTOREL	DETERMINE IF RELATION EXISTS BETWEEN INTRA-MOD ENTITIES
	SELRCNM	RETRIEVE ALL RELATION CLASSES FOR WHICH AN ENTITY IS PART OF
LISTS		
	ADDRCORR	ADD A TOKEN TO A CORRESPONDING LISTS NEXT ENTRY
	ADD_CORR	ADD A TOKEN TO CORRESPONDING LIST

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	ADD_TO_CN	INCREMENT A LIST COUNTER
	ADD_TO_LS	ADD A SINGLE TOKEN TO A PARSER OUTPUT LIST
	CPFCOR	ACCESS A TOKEN IN A CORRESPONDING NAMED LIST
	CPFNXT	ACCESS THE NEXT TOKEN IN A PARSER LIST.
	CPFONE	EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
	CPFVAL	RETURN THE COUNTER OF LIST1 BASED ON ROW IN LIST2
	INITCMD	INITIALIZATION FOR EACH COMMAND TO BE PROCESSED
	NDDL/MAIN	MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

NDDL

	ADDNOCORR	ADD A TOKEN TO A CORRESPONDING LISTS NEXT ENTRY
	ADD_CORR	ADD A TOKEN TO CORRESPONDING LIST
	ADD_TO_CN	INCREMENT A LIST COUNTER
	ADD_TO_LS	ADD A SINGLE TOKEN TO A PARSER OUTPUT LIST
	BRANCHR	PERFORMS MULTI-WAY CALL TO THE
	CPFCOR	ACCESS A TOKEN IN A CORRESPONDING NAMED LIST
	CPFNXT	ACCESS THE NEXT TOKEN IN A PARSER LIST.
	CPFONE	EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
	CPFVAL	RETURN THE COUNTER OF LIST1 BASED ON ROW IN LIST2
	ERRRPT	HANDLE ANY ERROR CODE FROM ORACLE.
	GETNCHR	GET THE NEXT CHARACTER FROM EITHER STANDARD INPUT
	INITCMD	INITIALIZATION FOR EACH COMMAND TO BE PROCESSED

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
	NDDL/MAIN	MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
	PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
	TERMSSES	ROUTINE TO TERMINATE AN NDDL
	UERROR	ISSUE A MESSAGE TO THE USER, CONSIDERED A
	UWARN	ISSUE A MESSAGE TO THE USER, CONSIDERED A
	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

NDDLEKY

	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

NDDLYAC.INP"

	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

OK

	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
	PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL

ORCLEDA

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	CHKKEYS	CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
	CHKMOD	DETERMINES WHETHER CERTAIN RULES ARE FULFILLED.
	GETECS	USING ENTITY CLASS VERIFIES CHECK MODEL RULES
RCDEPKC	ADDRCEC	POPULATES THE RC-DEPKC TABLE FOR ALL RELATIONS IN THE MODEL
	ALLKEY	GENERATES KEY CLASS FOR AN ENTITY
	CESTRUC	GENERATES NDDL COMMANDS ON A FILE FOR ALL ENTITIES FOR THE STRUCTURE
	CHKINH	checks the RC-DEPKC table and determines when a key class can be add
	CHKOWN	checks each of the OWN-ec's key classes against the RC-DEPKC LIST.
	CPYMOD	CONTROLS THE PROCESSING LOGIC FOR THE COPY MODEL COMMAND.
	INITRDL	INITIALISE THE TABLE WHICH STORES A MODELS RELATIONS
	MRGMOD	MERGE TWO IDEF MODELS INTO ONE
	MRGMOD1	COPY MODEL-1 INTO A NEW MODEL (MODEL-3)
RELTEL	CERELS	GENERATES NDDL COMMANDS ON A FILE FOR ALL ENTITIES IN A RELATION.
	CMBENT	CONROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.
	DEPFROM	GENERATE CREATE RELATION, DESCRIBE COMMANDS IN THE TO-MODEL
	FRTOREL	DETERMINE IF RELATION EXISTS BETWEEN INTRA-MOD ENTITIES

MDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	INDFROM	RETRIEVES RELATIONS, DETERMINES IND EC AND GENERATES MDDL
RENLIST	ADDNIG	PROCESS THE ADD MIGRATES...SET..CLAUSE
	CKRNLS	SEARCHES THE TABLE OF RENAME PAIR LOOKING FOR AN OLD-TAG ENTRY.
	MKRNLS	FETCH LIST OF RENAME PAIRS FOR MIGRATES...SET..CLAUSE
SDLIST	CDP4A	VERIFY SURROGATE ENTITY CLASS STRUCTURE
	VERRELS	VERIFY THE RELATION CLASS STRUCTURE
SRVRET	ADDKVA	INSERT AN ATTRIBUTE KEYWORD
	ADDKVE	INSERT AN ENTITY KEYWORD
	ADDKVR	INSERT A RELATION KEYWORD
	ADDNIG	PROCESS THE ADD MIGRATES...SET..CLAUSE
	ALLVIEW	CREATE AN ES-CS-MAPPING FOR AUC TO DATA ITEM
	BLKCL1	STORE ALL KEY CLASS INFO FOR AN ENTITY IN A LIST
	BLKCLST	SELECT AND STORE KEY CLASS INFO FOR A GIVEN ENTITY
	CHKATT	CHECK IF ATTRIBUTES HAVE BEEN CREATED ACCORDING TO STANDARDS
	CHKAUCV	CHECK EXISTENCE OF AUC TO SET MAPPING
	CHKREL	CHECK IF RELATIONS HAVE BEEN CREATED ACCORDING TO STANDARDS

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	CHBACAL	GENERATE CREATE ALIAS ATTRIBUTE.. AND ALIAS DESC TEXT COMMANDS
	CHBALI	GENERATE CREATE ALIAS ENTITY..COMMAND
	CHBEKV	GENERATE ADD KEYWORD CLAUSE FOR ENTITY KEYWORDS
	CHBOA	GENERATE COMMANDS FOR ATTRIBUTES, ITS KEYWORDS,ALIAS,DESC
	CHBRKV	SELECT AND GENERATE RELATION CLASS KEYWORDS
	COPYAC	CREATE AN ATTRIBUTE, ASSOCIATE WITH ENTITY, ADD KEY CLASSES
	DELDEDF	DELETE ALL DATA FIELD ASSOCIATIONS WITH THE DATABASE
	DELDERT	DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE DATABASE
	DELDBST	DELETE ALL RECORD SETS' ASSOCIATIONS WITH THE DATABASE
	DELDTNO	DELETE ALL DATA TYPES ASSOCIATED WITH A DOMAIN
	DELNDKC	DELETE ALL KEY CLASSES AND INHERITED KEYS FOR AN ENTITY
	DELNDRC	DELETE ALL RELATION CLASSES FOR AN ENTITY
	DELOAC	DELETE ALL OWNED AND INHERITED ATTRIBUTES
	DEPFRON	GENERATE CREATE RELATION,DESCRIBE COMMANDS IN THE TO-MODEL
	DLNDAUC	DELETE ATTRIBUTE USE CLASSES ASSOCIATED WITH ENTITY
	DONUSAG	DETERMINE IF DOMAIN ASSOCIATED WITH VIEWS,DATAFIELDS,ATTRIBUTES
	DRPAC	DELETE OWNED ATTRIBUTES ASSOCIATED WITH ENTITY
	DRPDIV	DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED WITH VIEW
	DRPNGEN	DELETE MIGRATED KEY CLASSES ASSOCIATED WITH A KEY CLASS
	DRPNGRC	DELETE MIGRATED KEY CLASS MEMBERS ASSOCIATED WITH A KEY CLASS
	DRPRCE	DROP A RELATION CLASS FOR AN ENTITY BEING DROPPED

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	FOOPATT	GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE COMMAND
	FINDDOM	RETRIEVE A DOMAIN NUMBER FOR A GIVEN DOMAIN NAME
	FEDINEM	RETRIEVE RECORD TYPE OF MEMBER ASSOCIATED WITH A SET
	FEDACH	DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL
	FEDASA	VERIFY WHETHER A SET HAS BEEN MAPPED TO AN AUC
	FEDASH	DETERMINES IF AN AUC TO SET TYPE MAPPING EXISTS
	FEDAUC	DELETE ALL ATTRIBUTE USE CLASSES FOR A GIVEN ENTITY CLASS
	FEDECH	DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR A GIVEN MODEL
	FEDOAC	DELETES ALL OWNED ATTRIBUTES FOR A GIVEN ENTITY CLASS
	FEDRCH	RETRIEVES ALL RC/RT MAPPINGS FOR A NAMED RECORD
	GENAKV	RETRIEVE ALL KEYWORDS FOR AN ATTRIBUTE CLASS
	GENDESC	GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN OBJECT TYPE AND NO
	GENEKV	SELECT KEYWORD FOR ENTITY AND CREATE KEYWORD PHRASE FOR CRT ENT
	GENROA	SELECT OWNED ATT FOR ENTITY AND CREATES OWNED ATT FOR CRT ENT
	GENREKV	SELECT KEYWORDS FOR RELATION CREATES EV PHRASE FOR CREATE RC
	GETACAL	VERIFY THE EXISTANCE OF AN ALIAS FOR AN ATTRIBUTE
	GETDOM	RETRIEVES DOMAIN NUMBER BASED ON TAG NAME FOR AUC
	GETECAL	VERIFIES THE EXISTANCE OF AN ALIAS FOR AN ENTITY
	GETECS	USING ENTITY CLASS VERIFIES CHECK MODEL RULES
	GETHAPC	SELECT ALL PROJECT DATA FIELDS MATCHING A GIVEN TAG NO

NDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	GETRCID	QUERIES CDM FOR INFORMATION ABOUT A RELATION CLASS
	INDFROM	RETRIEVES RELATIONS, DETERMINES IND EC AND GENERATES NDDL
	KEYLOOK	RETRIVES KEYCLASS NAME AND NUMBER BASED ON TAG NO
	PDFDS	RETRIEVES PROJECT DATA FIELDS FOR A GIVEN DATA BASE
	PDFDF	RETRIVES ALL THE PROJ DATA FIELD OR DF BEING DROPPED
	PDFREC	RETRIEVES ALL PORJECT DATA FIELDS FOR A GIVEN RECORD
	PDFSRCH	DETERMINE IF A TAG HAS A PRIMARY MAP TO DATA FIELD
	RACKV2	COMPARES ATTRIBUTE CLASS KEYWORDS
	RECKV2	COMPARES ENTITY CLASS KEYWORDS BETWEEN MODELS
	RELKV	CHECKS RC KV BETWEEN MODELS
	RETACKV	POPULATES KEYWORD TABLE FOR ATTRIBUTE CLASS KEYWORDS
	RETECKV	POPULATES KEYWORD TABLE FOR ENTITY CLASS KEYWORDS
	RETRAC1	DETERMINE IF ANY ATTRIBUTE CLASS NAMES MATCH BETWEEN MODELS
	RETRACP	SELECT ATTRIBUTE CLASS NAME USING THE ATTRIBUTE CLASS NUMBER
	RETRCKV	POPULATES THE KEYWORD CDM TABLE FOR RELATION CLASS KEYWORDS
	RETREC1	COMPARES ENTITY CLASS NAMES FOR TWO MODELS
	RETRECP	RETRIEVE AN ENTITY CLASS NAME FOR A GIVEN NO AND NAME TYPE
	RECKV2	COMPARES RELATION CLASS KEY WORDS FOR TWO MODELS
	SELACNM	RETRIEVE THE PRIMARY NAME FOR ATTRIBUTE CLASS
	SELECNM	RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS
	SELIKEY	RETRIEVE ALL INHERITED KEYS FOR A GIVEN EC AND RC

MDDL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	SELRCNM	RETRIEVE ALL RELATION CLASSES FOR WHICH AN ENTITY IS PART OF
	SELRSET	DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET BEING DROPPED
	SELSTH	DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE BEING DROPPED
	VERACDT	DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH ATTRIBUTE CLASSES
	VERACNM	RETRIEVE THE DOMAIN FOR AN ATTRIBUTE CLASS
	VERALI	VERIFY THE EXISTENCE OF AN ALIAS NAME FOR AN ENTITY
	VERAREA	VERIFY THE EXISTENCE OF AN AREA FOR A DATA BASE
	VERASH	DETERMINE IF ANY AUC/SET MAPPINGS EXIST FOR AN AUC
	VERATT	VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A MODEL
	VERAUC	VERIFY THE EXISTENCE OF AN ATTRIBUTE USE CLASS
	VERCRC	DETERMINE IF A KEY CLASS IS PART OF A COMPLETE RELATION
	VERDB	VERIFY THE EXISTENCE OF A DATA BASE IN THE CDM
	VERDBAS	VERIFY A DATA BASE FOR A GIVEN NAME AND HOST
	VERDF	VERIFY THE EXISTENCE OF A DATA FIELD
	VERDFDT	DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH A PROJECT DF
	VERDFLD	VERIFIES THE EXISTENCE OF A DATA FIELD FOR A GIVEN DATA BASE
	VERDI	VERIFY THE EXISTENCE OF A DATA ITEM IN A GIVEN VIEW
	VERDIDT	DETERMINE IF A DATA TYPE IS ASSOCIATED WITH ANY DATA ITEMS
	VERDON	VERIFY THE EXISTENCE OF A DOMAIN CLASS IN THE CDM
	VERDEL3	RETRIEVES SET ID FOR TOTAL DATA BASE
	VERDSTP	VALIDATE A USER ENTERED DESCRIPTION TYPE
	VERDSTI	VERIFY THE EXISTENCE OF DESC. TEXT FOR A GIVEN OBJECT

NDCL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
VERDT		VERIFY THE EXISTENCE OF A USER DEFINED DATA TYPE
VERDTD		VERIFY THE EXISTENCE OF A DATA TYPE IN A GIVEN DOMAIN
VERENT		VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A MODEL
VERKC		VERIFY THE EXISTENCE OF A KEY CLASS IN A MODEL
VERKW		verify the existence of a keyword.
VERKWE		VERIFY THE EXISTENCE OF ENTITY CLASS KEYWORD
VERKWR		VERIFIES THE EXISTANCE OF A RELATION CLASS KEYWORD.
VERMOD		verify the existence of a model and return the unique number.
VERMMA		verify the existence of an attribute class in a model.
VERMME		verify the existence of an entity class in a model.
VEROAC		verify the existence of an owned attribute class for an entity.
VERPDF		verify the existence of a project_data_field occurrence.
VERPSB		VERIFIES THE EXISTANCE OF A PROGRAM STATUS BLOCK FOR AN IMS DATABASE
VERRC		verify the existence of a relation class.
VERRCBS		verify if there are any mapping to a set found in RC_BASED_REC_SET.
VERRCC		verify if the relation class is complete.
VERRCMP		verify whether a member has been mapped to a relation class.
VERRCNM		VERIFY EXISTANCE OF A RELATION IN A MODEL
VERRCST		REPORT WHETHER A RC TO SET MAP EXISTS
VERRK		verifies the existance of a key for a record.
VERRKM		verifies the existance of a data field for a record.
VERRSET		VERIFIES THE EXISTANCE OF A SET.
VERRT		VERIFIES THE EXISTANCE OF A RECORD TYPE

SQL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	VERDET	For a given domain number, returns its standard data type name.
	VERDNC	fetch the number count for a given set
	VERTYP	verifies that a type identification for a data type is valid.
	VERDUTH	verify the existence of a USER DEFINED DATA TYPE (USERDATATYPE)/USER
	VERVIEW	verify the existence of a view in the system.
	VENAPS	ENFORCES AND TO SET MAPPING RULES
	VENACHV	RETRIEVE AND WRITE ALL THE KEYWORDS ASSOCIATED WITH AN ATTRIBUTE CLASS
	VENALI	RETRIEVE ALL ALIAS NAMES OF AN ATTRIBUTE CLASS.
	VENANM	RETRIEVE AND COPY ALL THE NAMES (PRIMARY AND ALIAS) OF AN ATTRIBUTE
	VENACHV	RETRIEVE AND COPY ALL THE KEYWORDS ASSOCIATED WITH AN ENTITY CLASS
	VENANM	RETRIEVE AND COPY ALL THE ALIAS NAMES OF AN ENTITY CLASS
STDIO		
	CLSPIL	THIS ROUTINE CLOSES AN OUTPUT FILE THE FILE WILL
	GETCHAR	GET THE NEXT CHARACTER FROM EITHER STANDARD INPUT
	WRFLIN	THIS ROUTINE WRITES A SQL COMMAND LINE (80 CHARACTERS)
	YTEROR PURPOSE NOT FOUND BY STRIPPER
	YTPARG PURPOSE NOT FOUND BY STRIPPER
	YTVRAP PURPOSE NOT FOUND BY STRIPPER
STDTP		
	ADDMAP	ADD A CS IS MAPPING

MODEL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	ALTATT	THE ALTER ATTRIBUTE COMMAND PROCESSOR ALTERS
	ALTRAP	ALTER MAP COMMAND PROCESSOR
	ALTRMAP	ALTER A SINGLE MAP
	CHGDOM	CHANGE THE DOMAIN FOR AN ALTERED ATTRIBUTE
	CHKLOOP	CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS
	CRTRAP	CREATE MAP COMMAND PROCESSOR
	DELASC	DELETE INHERITED ATT_USE, KEY CLASS MEMBER.
	DELAOK	DELETE ATTRIBUTE USE CLASS KEY MEMBER GIVEN
	DELMIGC	DELETE MIGRATING KEY CLASS
	DELMTEC	DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
	DELTTT	DELETE DESCRIPTION TEXT GIVEN THE OBJECT TYPE.
	DESCRS	COMMAND PROCESSOR FOR THE MODEL DESCRIBE COMMAND
	DLNIGC	DELETE MIGRATING KEY CLASS
	DRPRAP	COMMAND PROCESSOR FOR THE DROP MAP COMMAND
	DRPRMAP	DROP A SINGLE MAPPING
	GETCHAR	GET THE NEXT CHARACTER FROM EITHER STANDARD INPUT
	LOADDESC	LOAD DESCRIPTION FROM TEXT EDITOR
	MAPADF	MAP ON ASC TO A DATA FIELD
	MAPASST	MAP AN ASC TO A SET
	MAPRC	MAP A RELATION CLASS TO A SET
	REDESC	STORE DESCRIPTION ON THE CDM
	TETTP	DETERMINE SOURCE OF DESCRIPTION TEXT
	VEROBJ	VERIFY THAT THE OBJECT EXISTS

UNIGSDO

ADDEL	CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR CREATE ALTER ENTITY
ADDEV	ADD KEYWORDS FOR (X)MAPPING USING ALL KEYWORD (OPTIONAL)

DDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	ADDNIG	PROCESS THE ADD MIGRATE...SET..CLAUSE
	ADDQAC	ADD ATTRIBUTE AS AN OWNED ATTRIBUTE AND AS ATT USE CLASS
	ALTR00	CONTROLS PROCESSING FOR ALTER MODEL COMMANDS
	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)
	CHGDON	CHANGE THE DOMAIN FOR AN ALTERED ATTRIBUTE
	CRATT	CONTROLS THE PROCESSING LOGIC FOR CREATING AN ATTRIBUTE.
	CRTDON	PROGRAM NAME CRTDON
	CRTEXT	CONTROL THE PROCESSING LOGIC FOR CREATING A NEW ENTITY CLASS.
	CRTR00	CONTROL THE PROCESSING LOGIC IF CREATING A MODEL WITHIN THE SYSTEM
	CRTR01	CONTROLS THE LOGIC FOR VALIDATING AND CREATING A NEW RELATION CLASS
	CRTRVW	CONTROLS THE PROCESSING LOGIC FOR THE CREATE VIEW COMMAND
	DEFDB	CONTROLS THE PROCESSING LOGIC FOR DEFINING A DATABASE TO THE SYSTEM
	DEFSET	CONTROLS THE PROCESSING LOGIC FOR THE DEFINE SET COMMAND
	DELACC	DELETE INHERITED ATT USE KEY CLASS MEMBER.
	DELEDF	DELETE ALL DATA FIELD ASSOCIATIONS WITH THE DATABASE
	DELEDT	DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE DATABASE
	DELEDS	DELETE ALL RECORD SETS ASSOCIATIONS WITH THE DATABASE
	DELEFL	CONTROLS THE DELETING OF DATA FIELDS
	DELETED	DELETE ALL DATA TYPES ASSOCIATED WITH A DOMAIN
	DELEKRC	DELETE ALL KEY CLASSES AND INHERITED KEYS FOR AN ENTITY
	DELEKRC	DELETE ALL RELATION CLASSES FOR AN ENTITY
	DELEKRC	DELETE MIGRATING KEY CLASS

MODEL COMMAND PROCESSOR Where-include-file-used List

Include File	Module Name	Module Purpose
-----	-----	-----
	DELINTEC	DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
	ELMIDASC	DELETE ATTRIBUTE USE CLASSES ASSOCIATED WITH ENTITY
	ELMIGRC	DELETE MIGRATING KEY CLASS
	DRPATT	CONTROLS THE DROPPING OF USER SPECIFIED ATTRIBUTE CLASSES FROM THE C
	DRPDB	CONTROLS THE PROCESSING LOGIC FOR DELETING THE DATA BASE.
	DRPDIV	DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED WITH VIEW
	DRPDON	PROGRAM NAME
	DRPEFT	CONTROL THE PROCESSING LOGIC FOR DELETING ENTITIES
	DRPFLD	CONTROLS THE PROCESSING LOGIC FOR DROPPING A DATA FIELD.
	DRPEC	CONTROLS THE PROCESSING LOGIC FOR THE 'DROP KEY CLASS'
	DRPEUC	OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP THEIR ASSOCIATIONS
	DRPNOD	CONTROLS THE PROCESSING LOGIC TO DROP A MODEL
	DRPROE	DROP A RELATION CLASS FOR AN ENTITY BEING DROPPED
	DRPRC	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
	DRPVIEV	DROP THE VIEW
	PRDAON	DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL
	PRDON	DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR A GIVEN MODEL
	ICOPERT	INTERACTIVE COPY ENTITY WITH ATTRIBUTES KEYWORDS ALIAS DESC
	INSDFLD	INSERT A RECORD INTO THE DATA FIELD ENTITY
	INSDI	INSERT A RECORD INTO DATA ITEM

MSDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	INSDT	INSERT A RECORD INTO USER DEF DATA TYPE
	INSDTYP	INSERT A RECORD INTO THE RECORD_TYPE ENTITY. IF
	KEYLOOK	RETRIVES KEYCLASS NAME AND NUMBER BASED ON TAG NO
	SELASST	DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET BEING DROPPED
	SELSTH	DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE BEING DROPPED

VUDI

	BLVVLST	CREATE BUILD VIEW LISTS FOR THE CREATE VIEW COMMAND
	ORTVIEW	CONTROLS THE PROCESSING LOGIC FOR THE CREATE VIEW COMMAND.
	PIFROM	CREATE A VIEW USING A SINGLE ENTITY CLASS(ES-CS-MAPPING)
	PIFROM	CREATE A VIEW FROM MULTIPLE ENTITY CLASSES
	POPFROM	PROCESS A VIEW COMMAND FOR MULTIPLE ENTITY CLASSES
	RENVIEW	POPULATE VIEW DEFINITION AND ES-CS-MAPPING

VUFROM

	BLVVLST	CREATE BUILD VIEW LISTS FOR THE CREATE VIEW COMMAND
	ORTVIEW	CONTROLS THE PROCESSING LOGIC FOR THE CREATE VIEW COMMAND
	EXPRLT	EXPAND RELATION LIST TO INCLUDE DEF AND IND ENTITIES
	EXPRTLT	EXPAND THE RETRIEVE LIST (SELECT FROM)
	PIFROM	CREATE A VIEW USING A SINGLE ENTITY CLASS(ES-CS-MAPPING)
	PIFROM	CREATE A VIEW FROM MULTIPLE ENTITY CLASSES
	RENVIEW	POPULATE VIEW DEFINITION AND ES-CS-MAPPING

DDL COMMAND PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
VVRC	ELSBORC	CONTROLS PROCESS TO BUILD SEC-RC COMPONENTS
	BLVWLST	CREATE BUILD VIEW LISTS FOR THE CREATE VIEW COMMAND
	ORTVIEW	CONTROLS THE PROCESSING LOGIC FOR THE CREATE VIEW COMMAND.
	EXPRLT	EXPAND RELATION LIST TO INCLUDE DEP AND IND ENTITIES
	PNFROM PNOFROM	CREATE A VIEW FROM MULTIPLE ENTITY CLASSES PROCESS A VIEW COMMAND FOR MULTIPLE ENTITY CLASSES
	REVIEW VERRELS	POPULATE VIEW DEFINITION AND ES-CS-MAPPING VERIFY THE RELATION CLASS STRUCTURE
VVRETR	BLVWLST	CREATE BUILD VIEW LISTS FOR THE CREATE VIEW COMMAND
	ORTVIEW	CONTROLS THE PROCESSING LOGIC FOR THE CREATE VIEW COMMAND.
	EXPRLT	EXPAND THE RETRIEVE LIST (SELECT..FROM..)
	PIFROM	CREATE A VIEW USING A SINGLE ENTITY CLASS(ES-CS-MAPPING)
	PNFROM PNOFROM	CREATE A VIEW FROM MULTIPLE ENTITY CLASSES PROCESS A VIEW COMMAND FOR MULTIPLE ENTITY CLASSES
	REVIEW ENVIEW	POPULATE VIEW DEFINITION AND ES-CS-MAPPING CREATE ES-CS-MAPPING FOR USER SPECIFIED RETRIEVE LIST

PS 620141100
1 November 1985

3.10.6 Where External Routine Used List

The following lists each external function or routine listed in 3.10.3 and all the documented modules which call it. The purpose of each module is listed as well.

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
ADDFRM	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
ALLOC	ADDEC	ADD THE ENTITY NAME TO THE TREE LIST STRUCTURE
	ADDECNM	ADD THE EC NAME AND EC NO INTO KEYLIST
	ADDKCLS	ADD KEY INFO TO THE UNBOUNDED KEY CLASS LIST STRUCTURE
	ADDKG	ADD KCM TAG NUMBER AND NAME TO STRUCTURE
	NRSTORE	STORE A AVAILABLE NO OF A POOL NO BACK TO NO LINKED
ERRPRO	ADDKWA	INSERT AN ATTRIBUTE KEYWORD
	ADDKWE	INSERT AN ENTITY KEYWORD
	ADDKWR	INSERT A RELATION KEYWORD
	ADDMIG	PROCESS THE ADD MIGRATES...SET...CLAUSE
	ADDOAC	ADD ATTRIBUTE AS AN OWNED ATTRIBUTE AND AS ATT USE CLASS
	ADDRNUM	ADD A AVAILABLE NO OF A POOL NO BACK TO NO LINKED
	ADD_TO_CNTINCREMENT	A LIST COUNTER
	ADD_TO_LSTADD	A SINGLE TOKEN TO A PARSER OUTPUT LIST
	ALLKEY	GENERATES KEY CLASS FOR AN ENTITY
	ALLVIEW	CREATE AN ES-CS-MAPPING FOR AUC TO DATA ITEM
	ATTKW	CONTROLS PROCESSING TO POPULATE KEYWORD TABLE FOR AUC KEYWORD
	BLKCL1	STORE ALL KEY CLASS INFO FOR AN ENTITY IN A LIST
	BLKCLST	SELECT AND STORE KEY CLASS INFO FOR A GIVEN ENTITY
	BRANCHR	PERFORMS MULTI-WAY CALL TO THE
	CDP4A	VERIFY SURROGATE ENTITY CLASS STRUCTURE

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	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 STRUCTURE
	CHKATT	CHECK IF ATTRIBUTES HAVE BEEN CREATED ACCORDING TO STANDARDS
	CHKAUCV CHKKEYS	CHECK EXISTENCE OF AUC TO SET MAPPING CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
	CHKMOD	DETERMINES WHETHER CERTAIN RULES ARE FULFILLED,
	CHKREL	CHECK IF RELATIONS HAVE BEEN CREATED ACCORDING TO STANDARDS
	CMBACAL	GENERATE CREATE ALIAS ATTRIBUTE.. AND ALIAS DESC TEXT COMMANDS
	CMBALI CMBEKW	GENERATE CREATE ALIAS ENTITY..COMMAND GENERATE ADD KEYWORD CLAUSE FOR ENTITY KEYWORDS
	CMBENT	CONROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.
	CMBOA	GENERATE COMMANDS FOR ATTRIBUTES, ITS KEYWORDS, ALIAS, DESC
	CMBRKW	SELECT AND GENERATE RELATION CLASS KEYWORDS
	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.
	COPYAC	CREATE AN ATTRIBUTE, ASSOCIATE WITH ENTITY, ADD KEY CLASSES
	CRTMOD	CONTROL THE PROCESSING LOGIC IF CREATING A MODEL WITHIN THE SYSTEM.
	DEFAREA	PROCESSES THE AREA INFORMATION IF THE DBMS IS IDS-II, IDMS AND VAX-1
	DEFCODL	PROCESSES THE DBMS TYPES: VAX-11, IDMS, IDS-II.
	DEFDB	CONTROLS THE PROCESSING LOGIC FOR DEFINING A DATABASE TO THE SYSTEM.

DDL COMMAND PROCESSOR Where-external-routine-used List

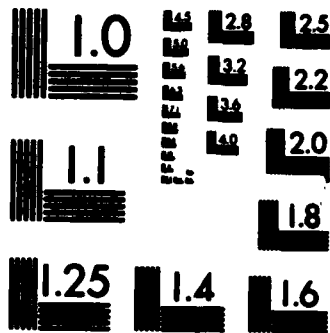
System Module	Module Name	Module Purpose
-----	-----	-----
	DEPFLD	PROCESSES THE FIELD INFORMATION FOR THE DEFINE RECORD COMMAND.
	DEFINS	PROCESSES THE DBMS TYPE: IMS.
	DEFINSS	PROCESSES THE SEGMENT INFORMATION IF THE DBMS IS IMS.
	DEFKEY	PROCESSES THE KEY INFORMATION FOR THE DEFINE RECORD COMMAND.
	DEFORCL	PROCESSES THE DBMS TYPE: ORACLE.
	DEPREC	CONTROLS THE PROCESSING LOGIC FOR DEFINING A RECORD FOR THE SYSTEM.
	DEFTOT	PROCESSES THE DBMS TYPE: TOTAL.
	DELDEBDF	DELETE ALL DATA FIELD ASSOCIATIONS WITH THE DATABASE
	DELDERT	DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE DATABASE
	DELDBST	DELETE ALL RECORD SETS' ASSOCIATIONS WITH THE DATABASE
	DELDTNO	DELETE ALL DATA TYPES ASSOCIATED WITH A DOMAIN
	DELMDKC	DELETE ALL KEY CLASSES AND INHERITED KEYS FOR AN ENTITY
	DELMDRC	DELETE ALL RELATION CLASSES FOR AN ENTITY
	DELOAC	DELETE ALL OWNED AND INHERITED ATTRIBUTES
	DEPFROM	GENERATE CREATE RELATION, DESCRIBE COMMANDS IN THE TO-MODEL
	DLMDAUC	DELETE ATTRIBUTE USE CLASSES ASSOCIATED WITH ENTITY
	DOMUSAG	DETERMINE IF DOMAIN ASSOCIATED WITH VIEWS, DATAFIELDS, ATTRIBUTES
	DRPAC	DELETE OWNED ATTRIBUTES ASSOCIATED WITH ENTITY
	DRPATT	CONTROLS THE DROPPING OF USER SPECIFIED ATTRIBUTE CLASSES FROM THE C
	DRPDB	CONTROLS THE PROCESSING LOGIC FOR DELETING THE DATA BASE.
	DRPDIV	DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED WITH VIEW
	DRPFLD	CONTROLS THE PROCESSING LOGIC FOR DROPPING A DATA FIELD.

MDDL COMMAND PROCESSOR Where-external-routine-used List

System Module	Module Name	Module Purpose
-----	-----	-----
	DRPKC	CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY CLASS".
	DRPMGKM	DELETE MIGRATED KEY CLASSES ASSOCIATED WITH A KEY CLASS
	DRPMGRC	DELETE MIGRATED KEY CLASS MEMBERS ASSOCIATED WITH A KEY CLASS
	DRPMOD	CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.
	DRPRCE	DROP A RELATION CLASS FOR AN ENTITY BEING DROPPED
	DRPSET	CONTROLS THE PROCESSING LOGIC FOR DELETING A SET FROM THE DATABASE.
	ENTKW	SEARCH FOR ENTITY KEYWORD MATCHES WITHIN TWO MODELS
	ERRRPT	HANDLE ANY ERROR CODE FROM ORACLE.
	FCOPATT	GENERATE MDDL COMMANDS FROM A COPY ATTRIBUTE COMMAND
	FCOPENT	DETERMINE IF COPY ENTITY WITH STRUCTURE OR RELATION
	FILEINS	INSERT DESCRIPTION TEXT INTO CDM
	FINDDOM	RETRIEVE A DOMAIN NUMBER FOR A GIVEN DOMAIN NAME
	FNDMEM	RETRIEVE RECORD TYPE OF MEMBER ASSOCIATED WITH A SET
	FNDACH	DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL
	FNDASA	VERIFY WHETHER A SET HAS BEEN MAPPED TO AN AUC
	FNDASH	DETERMINES IF AN AUC TO SET TYPE MAPPING EXISTS
	FNDAUC	DELETE ALL ATTRIBUTE USE CLASSES FOR A GIVEN ENTITY CLASS
	FNDECM	DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR A GIVEN MODEL
	FNDOAC	DELETES ALL OWNED ATTRIBUTES FOR A GIVEN ENTITY CLASS
	FNDRCM	RETRIEVES ALL RC/RT MAPPINGS FOR A NAMED RECORD
	FRTOREL	DETERMINE IF RELATION EXISTS BETWEEN INTRA-MOD ENTITIES

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module	Module Name	Module Purpose
-----	-----	-----
	GENAKV	RETRIEVE ALL KEYWORDS FOR AN ATTRIBUTE CLASS
	GENDESC	GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN OBJECT TYPE AND NO
	GENEKV	SELECT KEYWORD FOR ENTITY AND CREATE KEYWORD PHRASE FOR CRT ENT
	GENENT1	GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND DESCRS.
	GENOA	SELECT OWNED ATT FOR ENTITY AND CREATES OWNED ATT FOR CRT ENT
	GENRKV	SELECT KEYWORDS FOR RELATION CREATES KW PHRASE FOR CREATE RC
	GETACAL	VERIFY THE EXISTANCE OF AN ALIAS FOR AN ATTRIBUTE
	GETDBST	RETURN INFORMATION ABOUT THE CURRENT SESSIONS' DATABASE
	GETDOM	RETRIEVES DOMAIN NUMBER BASED ON TAG NAME FOR AUC
	GETECAL	VERIFIES THE EXISTANCE OF AN ALIAS FOR AN ENTITY
	GETECS	USING ENTITY CLASS VERIFIES CHECK MODEL RULES
	GETMAPC	SELECT ALL PROJECT DATA FIELDS MATCHING A GIVEN TAG NO
	GETNNUM	
	GETRCID	QUERIES CDM FOR INFORMATION ABOUT A RELATION CLASS
	GETRCNM	RETRIEVE RELTION CLASS NAME, IND AND DEP ENTITY NAMES
	GETRDH	RETURN WITH CURRENT SESSIONS' DATA BASE INFO
	ICOPATT	INTERACTIVE COPY ATTRIBUTE, WITH KEYWORDS, ALIAS, DESCRS.
	ICOPENT	INTERACTIVE COPY ENTITY WITH ATTRIBUTES, KEYWORDS, ALIAS, DESC
	INDFROM	RETRIEVES RELATIONS, DETERMINES IND EC AND GENERATES NDDL
	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

WDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	KEYLOOK	RETRIVES KEYCLASS NAME AND NUMBER BASED ON TAG NO
	MRGMOD	MERGE TWO IDEF MODELS INTO ONE
	MRGMOD1	COPY MODEL-1 INTO A NEW MODEL (MODEL-3)
	PDFDB	RETRIEVES PROJECT DATA FIELDS FOR A GIVEN DATA BASE
	PDFDF	RETRIVES ALL THE PROJ DATA FIELD OR DF BEING DROPPED
	PDFREC	RETRIEVES ALL PORJECT DATA FIELDS FOR A GIVEN RECORD
	PDFSRCH	DETERMINEES IF A TAG HAS A PRIMARY MAP TO DATA FIELD
	PRGCMO	THIS ROUTINE IS CALLED TO HANDLE ALL
	RACKW2	COMPARES ATTRIBUTE CLASS KEYWORDS
	RECKW2	COMPARES ENTITY CLASS KEYWORDS BETWEEN MODELS
	RELKW	CHECKS RC KW BETWEEN MODELS
	RETACKW	POPULATES KEYWORD TABLE FOR ATTRIBUTE CLASS KEYWORDS
	RETECKW	POPULATES KEYWORD TABLE FOR ENTITY CLASS KEYWORDS
	RETRAC1	DETERMINEES IF ANY ATTRIBUTE CLASS NAMES MATCH BETWEEN MODELS
	RETRACP	SELECT ATTRIBUTE CLASS NAME USING THE ATTRIBUTE CLASS NUMBER
	RETRCKW	POPULATES THE KEYWORD CDM TABLE FOR RELATION CLASS KEYWORDS
	RETREC1	COMPARES ENTITY CLASS NAMES FOR TWO MODELS
	RETRECP	RETRIEVE AN ENTITY CLASS NAME FOR A GIVEN NO. AND NAME TYPE
	RRCKW2	COMPARES RELATION CLASS KEY WORDS FOR TWO MODELS
	SELACNM	RETRIEVE THE PRIMARY NAME FOR ATTRIBUTE CLASS
	SELECNM	RETRIEVE THE PRIMARY NAME FOR AN ENTITY CLASS
	SELIKEY	RETRIEVE ALL INHERITED KEYS FOR A GIVEN EC AND RC
	SELRCNM	RETRIEVE ALL RELATION CLASSES FOR WHICH AN ENTITY IS PART OF

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	SELRSET	DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET BEING DROPPED
	SELSTH	DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE BEING DROPPED
	STRINS	INSERT DESCRIPTION TEXT INTO THE CDM
	VERACDT	DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH ATTRIBUTE CLASSES
	VERACNM	RETRIEVE THE DOMAIN FOR AN ATTRIBUTE CLASS
	VERALI	VERIFY THE EXISTENCE OF AN ALIAS NAME FOR AN ENTITY
	VERAREA	VERIFY THE EXISTENCE OF AN AREA FOR A DATA BASE
	VERASH	DETERMINE IF ANY AUC/SET MAPPINGS EXIST FOR AN AUC
	VERATT	VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A MODEL
	VERAUC	VERIFY THE EXISTENCE OF AN ATTRIBUTE USE CLASS
	VERCRC	DETERMINE IF A KEY CLASS IS PART OF A COMPLETE RELATION
	VERDB	VERIFY THE EXISTENCE OF A DATA BASE IN THE CDM
	VERDBAS	VERIFY A DATA BASE FOR A GIVEN NAME AND HOST
	VERDF	VERIFY THE EXISTENCE OF A DATA FIELD
	VERDFDT	DETERMINE ANY ASSOCIATIONS OF A DATA TYPE WITH A PROJECT DF
	VERDFLD	VERIFIES THE EXISTENCE OF A DATA FIELD FOR A GIVEN DATA BASE
	VERDI	VERIFY THE EXISTENCE OF A DATA ITEM IN A GIVEN VIEW
	VERDIDT	DETERMINE IF A DATA TYPE IS ASSOCIATED WITH ANY DATA ITEMS
	VERDOM	VERIFY THE EXISTENCE OF A DOMAIN CLASS IN THE CDM
	VERDSL3	RETRIEVES SET ID FOR TOTAL DATA BASE
	VERDSTP	VALIDATE A USER ENTERED DESCRIPTION TYPE
	VERDSTX	VERIFY THE EXISTENCE OF DESC. TEXT FOR A GIVEN OBJECT

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
VERDT		VERIFY THE EXISTENCE OF A USER DEFINED DATA TYPE
VERDTD		VERIFY THE EXISTENCE OF A DATA TYPE IN A GIVEN DOMAIN
VERENT		VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A MODEL
VERKC		VERIFY THE EXISTENCE OF A KEY CLASS IN A MODEL
VERKW		verify the existence of a keyword.
VERKWE		VERIFY THE EXISTENCE OF ENTITY CLASS KEYWORD
VERKWR		VERIFIES THE EXISTANCE OF A RELATION CLASS KEYWORD.
VERMOD		verify the existence of a model and return the unique number.
VERNMA		verify the existence of an attribute class in a model.
VERNME		verify the existence of an entity class in a model.
VEROAC		verify the existence of an owned attribute class for an entity.
VERPDF		verify the existence of a project_data_field occurrence.
VERPSB		VERIFIES THE EXISTANCE OF A PROGRAM STATUS BLOCK FOR AN IMS DATABASE
VERRC		verify the existence of a relation class.
VERRCBS		verify if there are any mapping to a set found in RC_BASED_REC_SET.
VERRCC		verify if the relation class is complete.
VERRCMP		verify whether a member has been mapped to a relation class.
VERRCNM		VERIFY EXISTANCE OF A RELATION IN A MODEL
VERRCST		REPORT WHETHER A RC TO SET MAP EXISTS.
VERRK		verifies the existance of a key for a record.
VERRKM		verifies the existance of a data field for a record.
VERRSET		VERIFIES THE EXISTANCE OF A SET.
VERRT		VERIFIES THE EXISTANCE OF A RECORD TYPE.

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	VERSDT	For a given domain number, return its standard data type name.
	VERSMS	fetch the member count for a given set.
	VERTYP	verifies that a type identification for a data type is valid.
	VERUDTN	verify the existence of a USER DEFINED DATA TYPE (USERDATATYPE)/USDF
	VERVIEW	verify the existence of a view in the system.
	VOMAPS	ENFORCES AUC TO SET MAPPING RULES.
	WRTACKW	RETRIEVE AND WRITE ALL THE KEYWORDS ASSOCIATED WITH AN ATTRIBUTE CLASS
	WRTALI	RETRIEVE ALL ALIAS NAMES OF AN ATTRIBUTE CLASS.
	WRTANAM	RETRIEVE AND COPY ALL THE NAMES (PRIMARY AND ALIAS) OF AN ATTRIBUTE.
	WRTECKW	RETRIEVE AND COPY ALL THE KEYWORDS ASSOCIATED WITH AN ENTITY CLASS.
	WRTENAM	RETRIEVE AND COPY ALL THE ALIAS NAMES OF AN ENTITY CLASS.
EXIT	NDDL/MAIN	MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
FCLOSE	CLSFIL	THIS ROUTINE CLOSSES AN OUTPUT FILE. THE FILE WILL
FREE	COMMIT	STORE THE REUSEABLE NUMBER TO THE DATA BASE.
	ROLBACK	ROLBACK THE TRANSACTIONS.

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
GDATA	PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
GETCHAR	GETNCHR	GET THE NEXT CHARACTER FROM EITHER STANDARD INPUT
INITEX	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
INITFP	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
LOGOFF	TERMSES	ROUTINE TO TERMINATE AN NDDL
LOGON	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
OBINDN	CHKKEYS	CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.

DDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	COBINDM	ORACLE ROUTINE
	FILEINS	INSERT DESCRIPTION TEXT INTO CDM
	STRINS	INSERT DESCRIPTION TEXT INTO THE CDM
OCLOSE	COCLOSE	ORACLE ROUTINE
OCOF	COCOF	ORACLE ROUTINE
OCOM	COCOM	ORACLE ROUTINE
ODFINN	CHKKEYS	CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
	CODFINN	ORACLE ROUTINE
OERMSG	COERMSG	ORACLE ROUTINE
OEXEC	CHKKEYS	CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
	COEXEC	ORACLE ROUTINE
	FILEINS	INSERT DESCRIPTION TEXT INTO CDM
	STRINS	INSERT DESCRIPTION TEXT INTO THE CDM

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
OFETCH	CHKKEYS	CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
	COFETCH	ORACLE ROUTINE
OISCR	PRGCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
OLOGOF	COLOGOF	ORACLE ROUTINE
OLON	COLON	ORACLE ROUTINE
OOPEN	CHKKEYS	CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
	COOPEN	ORACLE ROUTINE
OPNFIL	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 STRUCTURE
	CMBENT	CONROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND.

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	CPYMOD	CONTROLS THE PROCESSING LOGIC FOR THE COPY MODEL COMMAND.
	FCOPATT	GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE COMMAND
	MRGMOD	MERGE TWO IDEF MODELS INTO ONE
OPNFRM	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
OROL	COROL	ORACLE ROUTINE
OSQL3	CHKKEYS	CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
	COSQL3	ORACLE ROUTINE
PDATA	PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
PMSGLS	PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
	UERROR	ISSUE A MESSAGE TO THE USER, CONSIDERED A
	UWARN	ISSUE A MESSAGE TO THE USER, CONSIDERED A
PRINTF		

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	AKCROW	THIS ROUTINE ADDS A ROW TO THE UNBOUNDED KEY CLASS LIST
	BRANCHR COMMIT	PERFORMS MULTI-WAY CALL TO THE STORE THE REUSEABLE NUMBER TO THE DATA BASE.
	UERROR UWARN VEROBJ YYPARSE	ISSUE A MESSAGE TO THE USER, CONSIDERED A ISSUE A MESSAGE TO THE USER, CONSIDERED A VERIFY THAT THE OBJECT EXISTS. **** PURPOSE NOT FOUND BY STRIPPER ****
PUTC	VRTLIN	THIS ROUTINE WRITES A NDDL COMMAND LINE (80 CHARACTERS)
RPLFRM	PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
SPRINTF	ADDMAP AKCROW ALTATT ALTMAP ALTSMAP BLOOPCK CHGDOM CHKLOOP CPYDES CRTMAP DELAUC	ADD A CS-IS MAPPING THIS ROUTINE ADDS A ROW TO THE UNBOUNDED KEY CLASS LIST THE ALTER ATTRIBUTE COMMAND PROCESSOR ALTERS ALTER MAP COMMAND PROCESSOR ALTER A SINGLE MAP CHECK FOR LOOPS FROM THE ENTITY GIVEN UP THE HIERARCHY CHANGE THE DOMAIN FOR AN ALTERED ATTRIBUTE CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS COMMAND PROCESSOR FOR COPY DESCRIPTION CREATE MAP COMMAND PROCESSOR DELETE INHERITED_ATT_USE, KEY_CLASS_MEMBER,

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	DELAUCK	DELETE ATTRIBUTE USE CLASS KEY MEMBER GIVEN
	DELMIGK	DELETE MIGRATING KEY CLASS
	DELMTKC	DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
	DESCRB	COMMAND PROCESSOR FOR THE NDDL DESCRIBE COMMAND
	DELMIGRC	DELETE MIGRATING KEY CLASS
	DRPMAP	COMMAND PROCESSOR FOR THE DROP MAP COMMAND
	DRPSMAP	DROP A SINGLE MAPPING
	ERRRPT	HANDLE ANY ERROR CODE FROM ORACLE,
	INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
	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
	LOADESC	LOAD DESCRIPTION FROM TEXT EDITOR
	MAPADF	MAP ON AUC TO A DATA FIELD
	MAPASET	MAP AN AUC TO A SET
	MAPRC	MAP A RELATION CLASS TO A SET
	MRGMOD2	CONTROL THE LOGIC FOR PROCESSING THE REMAINING MODEL 2
	MRGNODE	SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT ENTITY'S
	PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
	RDDESC	STORE DESCRIPTION ON THE CDM
	TLOOPCK	CHECK FOR LOOPS FROM THE ENTITY GIVEN DOWN THE HIERARCHY
	VEROBJ	VERIFY THAT THE OBJECT EXISTS.
	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
STRCAT	ADDNCORR	ADD A TOKEN TO A CORRESPONDING LISTS NEXT ENTRY
	ADD_CORR	ADD A TOKEN TO CORRESPONDING LIST

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module	Module Name	Module Purpose
--------------------------	------------------------	---------------------------

ADD_TO_LSTADD A SINGLE TOKEN TO A PARSER OUTPUT LIST

STRLEN

ADDNCORR	ADD A TOKEN TO A CORRESPONDING LISTS NEXT ENTRY
ADD_CORR	ADD A TOKEN TO CORRESPONDING LIST
ADD_TO_LSTADD	ADD A SINGLE TOKEN TO A PARSER OUTPUT LIST

STRNCMP

ADDMAP	ADD A CS-IS MAPPING
ALLATT	SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND GENERATE
ALLENT	SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND GENERATE
ALTMAP	ALTER MAP COMMAND PROCESSOR
ALTSMAP	ALTER A SINGLE MAP
DEPATT	SELECT ALL THE ATTRIBUTES IN THE
DEPENT	SELECT ALL THE DEPENDANT ENTITY CLASSES
DRPSMAP	DROP A SINGLE MAPPING
HALT	HALT WITH 'COMMIT' OR 'ROLLBACK'.
INITSES	PERFORM ANY SESSION INITIALIZATION NECESSARY
PRCCHD	THIS ROUTINE IS CALLED TO HANDLE ALL
VEROBJ	VERIFY THAT THE OBJECT EXISTS.

STRNCPY

ADDECNM	ADD THE EC_NAME AND EC_NO INTO KEYLIST
ADDKCLS	ADD KEY INFO TO THE UNBOUNDED KEY_CLASS_LIST STRUCTURE
ADDKG	ADD_KCM_TAG NUMBER AND NAME TO STRUCTURE
ALLATT	SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND GENERATE
ALLENT	SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND GENERATE

NDDL COMMAND PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	ALLREL	FOR EACH LEVEL OF RELATIONS IN FROM-MODEL GENERATE
	ALTMAP	ALTER MAP COMMAND PROCESSOR
	CPFCOR	ACCESS A TOKEN IN A CORRESPONDING NAMED LIST
	CPFNXT	ACCESS THE NEXT TOKEN IN A PARSER LIST.
	CPFONE	EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
	DEPATT	SELECT ALL THE ATTRIBUTES IN THE
	DEPENT	SELECT ALL THE DEPENDANT ENTITY CLASSES
	DEPREL	FOR EACH LEVEL OF RELATIONS IN STRUCTURE GENERATE
	DPKCLST	CREATE AN KEY_CLASS_LIST TABLE CONTAINING ALL THE ENTITY
	GETECNM	THIS ROUTINE SEARCHES THE UNBOUNDED EC LIST DATA
	GETGLOB	WILL PROVIDE GLOBAL VARIABLES
	MRGMOD2	CONTROL THE LOGIC FOR PROCESSING THE REMAINING MODEL 2
	NEXTKC	THIS ROUTINE RETURNS A KC_NAME FOR A GIVEN EC NO FROM
	NEXTKCM	THIS ROUTINE RETURNS A KG_NO,KG_NAME FOR A GIVEN EC NO
	PRCCMD	THIS ROUTINE IS CALLED TO HANDLE ALL
	SELIAUC	SELECTS ALL THE INHERITED TAG NAMES
	UERROR	ISSUE A MESSAGE TO THE USER, CONSIDERED A
	UWARN	ISSUE A MESSAGE TO THE USER, CONSIDERED A
TERMFP	TERMSSES	ROUTINE TO TERMINATE AN NDDL
TOLOWER	ADDMAP	ADD A CS-IS MAPPING
	ALTMAP	ALTER MAP COMMAND PROCESSOR
	ALTSMAP	ALTER A SINGLE MAP
	DRPSMAP	DROP A SINGLE MAPPING

3.10.7 Main Program Parts List

The following lists each Main Program listed in 3.10.1 and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more than once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external routine". The Purpose of the Main Program module is listed as well.

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
DELDL1		Purpose-->DELETE A RECORD FROM DATA_FIELD KNITY
	COBINDM	Well-defined module
	COERMSG	Well-defined module
	GOEXEC	Well-defined module
	COSQL3	Well-defined module
	ERRPRO	External routine
	ERRRPT	Well-defined module
	OBINDM	External routine
	OERMSG	External routine
	OEXEC	External routine
	OSQL3	External routine
	SPRINTF	External routine

NDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
NDDL/MAIN	Purpose--	MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR
	ADDATT	Well-defined module
	ADDDT	Well-defined module
	ADDEC	Well-defined module
	ADDECNM	Well-defined module
	ADDFRM	External routine
	ADDKC	Well-defined module
	ADDKCLS	Well-defined module
	ADDKG	Well-defined module
	ADDKH	Well-defined module
	ADDKV	Well-defined module
	ADDKWA	Well-defined module
	ADDKWE	Well-defined module
	ADDKWR	Well-defined module
	ADDMAP	Well-defined module
	ADDHIG	Well-defined module
	ADDNCORR	Well-defined module
	ADDNSTD	Well-defined module
	ADDOAC	Well-defined module
	ADDPARM	Well-defined module
	ADDRCEC	Well-defined module
	ADDRNUM	Well-defined module
	ADDSTD	Well-defined module
	ADDTXT	Well-defined module
	ADD_CORR	Well-defined module
	ADD_TO_CNT	Well-defined module
	ADD_TO_LST	Well-defined module
	ADPARM1	Well-defined module
	AKCROW	Well-defined module
	ALLATT	Well-defined module
	ALLENT	Well-defined module
	ALLKEY	Well-defined module
	ALLOC	External routine
	ALLREL	Well-defined module
	ALLVIEW	Well-defined module
	ALTALI	Well-defined module
	ALTATT	Well-defined module
	ALTCARD	Well-defined module
	ALTDOM	Well-defined module

WDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	ALTDT	Well-defined module
	ALTENT	Well-defined module
	ALTMAP	Well-defined module
	ALTMOD	Well-defined module
	ALTREL	Well-defined module
	ALTSMAP	Well-defined module
	ATTKW	Well-defined module
	BLDATT	Well-defined module
	BLDATT1	Well-defined module
	BLECLST	Well-defined module
	BLKCL1	Well-defined module
	BLKCLST	Well-defined module
	BLOOPCK	Well-defined module
	BLRCKC	Well-defined module
	BLRCKC1	Well-defined module
	BLSECR	Well-defined module
	BLVWLST	Well-defined module
	BRANCHR	Well-defined module
	CDP4A	Well-defined module
	CERELS	Well-defined module
	GESTRUC	Well-defined module
	CHGDOM	Well-defined module
	CHGLOBL	Well-defined module
	CHKATT	Well-defined module
	CHKAUCV	Well-defined module
	CHKCARD	Well-defined module
	CHKDOMS	Well-defined module
	CHKINH	Well-defined module
	CHKKEYS	Well-defined module
	CHKLOOP	Well-defined module
	CHKMOD	Well-defined module
	CHKOWN	Well-defined module
	CHKREL	Well-defined module
	CKDUPEC	Well-defined module
	CKRNLST	Well-defined module
	CLSFIL	Well-defined module
	CMBACAL	Well-defined module
	CMBALI	Well-defined module
	CMBEKW	Well-defined module
	CMBENT	Well-defined module

NDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	CMBOA	Well-defined module
	CMBRKW	Well-defined module
	CMPMOD	Well-defined module
	COBINDN	Well-defined module
	COCLOSE	Well-defined module
	COCOF	Well-defined module
	COCOM	Well-defined module
	CODFINN	Well-defined module
	COERMSG	Well-defined module
	COEXEC	Well-defined module
	COFETCH	Well-defined module
	COLOGOF	Well-defined module
	COLON	Well-defined module
	COMMIT	Well-defined module
	COOPEN	Well-defined module
	COPATT	Well-defined module
	COPENT	Well-defined module
	COPYAC	Well-defined module
	COROL	Well-defined module
	COSQL3	Well-defined module
	CPFCOR	Well-defined module
	CPFNXT	Well-defined module
	CPFONE	Well-defined module
	CPFVAL	Well-defined module
	CPYDES	Well-defined module
	CPYMOD	Well-defined module
	CRTALI	Well-defined module
	CRTATT	Well-defined module
	CRTDOM	Well-defined module
	CRTENT	Well-defined module
	CRTMAP	Well-defined module
	CRTMOD	Well-defined module
	CRTREL	Well-defined module
	CRTVIEW	Well-defined module
	DEFAREA	Well-defined module
	DEFCODL	Well-defined module
	DEFDB	Well-defined module
	DEFFLD	Well-defined module
	DEFIMS	Well-defined module
	DEFIMSS	Well-defined module

MDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	DEFKEY	Well-defined module
	DEFORCL	Well-defined module
	DEFREC	Well-defined module
	DEFSET	Well-defined module
	DEFTOT	Well-defined module
	DEL1PDF	Well-defined module
	DELAC	Well-defined module
	DELACAL	Well-defined module
	DELACKV	Well-defined module
	DELACNM	Well-defined module
	DELASM	Well-defined module
	DELASM1	Well-defined module
	DELASM2	Well-defined module
	DELAUC	Well-defined module
	DELAUCK	Well-defined module
	DELAUCL	Well-defined module
	DELCMPR	Well-defined module
	DELCPRC	Well-defined module
	DELDA1	Well-defined module
	DELDA2	Well-defined module
	DELDB1	Well-defined module
	DELDEDF	Well-defined module
	DELDBRT	Well-defined module
	DELDBS1	Well-defined module
	DELDBST	Well-defined module
	DELDFL2	Well-defined module
	DELDFL3	Well-defined module
	DELDIV	Well-defined module
	DELDOM	Well-defined module
	DELDSL1	Well-defined module
	DELDSL2	Well-defined module
	DELDSL3	Well-defined module
	DELDT	Well-defined module
	DELDTD	Well-defined module
	DELDTNO	Well-defined module
	DELEC	Well-defined module
	DELEGAL	Well-defined module
	DELECKW	Well-defined module
	DELECNM	Well-defined module
	DELIASM	Well-defined module

HDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	DELIAUC	Well-defined module
	DELIAUK	Well-defined module
	DELIPDF	Well-defined module
	DELIRCS	Well-defined module
	DELISS1	Well-defined module
	DELISS2	Well-defined module
	DELKC	Well-defined module
	DELKCN	Well-defined module
	DELKCMT	Well-defined module
	DELKW	Well-defined module
	DELKWAC	Well-defined module
	DELKVEC	Well-defined module
	DELKVRC	Well-defined module
	DELMDKC	Well-defined module
	DELMDRC	Well-defined module
	DELMIGK	Well-defined module
	DELMOD	Well-defined module
	DELMTKC	Well-defined module
	DELOAC	Well-defined module
	DELOACE	Well-defined module
	DELOWAC	Well-defined module
	DELPCB	Well-defined module
	DELPDFT	Well-defined module
	DELPDI	Well-defined module
	DELRBR1	Well-defined module
	DELRBR2	Well-defined module
	DELRBR3	Well-defined module
	DELRC	Well-defined module
	DELRCCKW	Well-defined module
	DELRCST	Well-defined module
	DELREUS	Well-defined module
	DELRKM1	Well-defined module
	DELRKM2	Well-defined module
	DELRKM3	Well-defined module
	DELRKY1	Well-defined module
	DELRKY2	Well-defined module
	DELRST2	Well-defined module
	DELRST3	Well-defined module
	DELRTY2	Well-defined module
	DELSDF1	Well-defined module

NDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	DELSDF2	Well-defined module
	DELSDF3	Well-defined module
	DELSEC	Well-defined module
	DELSECR	Well-defined module
	DELSN1	Well-defined module
	DELSTN1	Well-defined module
	DELSTN2	Well-defined module
	DELSTN3	Well-defined module
	DELTEXT	Well-defined module
	DELTYT	Well-defined module
	DEPATT	Well-defined module
	DEPENT	Well-defined module
	DEPFROM	Well-defined module
	DEPREL	Well-defined module
	DESCRB	Well-defined module
	DLDSL2	Well-defined module
	DLNDAUC	Well-defined module
	DLNIGRC	Well-defined module
	DOMUSAG	Well-defined module
	DPKCLST	Well-defined module
	DRPAC	Well-defined module
	DRPALI	Well-defined module
	DRPATT	Well-defined module
	DRPDB	Well-defined module
	DRPDF	Well-defined module
	DRPDIV	Well-defined module
	DRPDOM	Well-defined module
	DRPDT	Well-defined module
	DRPENT	Well-defined module
	DRPFLD	Well-defined module
	DRPKC	Well-defined module
	DRPKW	Well-defined module
	DRPKWC	Well-defined module
	DRPHAP	Well-defined module
	DRPHGKH	Well-defined module
	DRPHGRC	Well-defined module
	DRPHIG	Well-defined module
	DRPHOD	Well-defined module
	DRPRCE	Well-defined module
	DRPREC	Well-defined module

WDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
-----	-----	-----
	DRPREL	Well-defined module
	DRPSET	Well-defined module
	DRPSMAP	Well-defined module
	DRPVIEW	Well-defined module
	DTUSAGE	Well-defined module
	ENTKW	Well-defined module
	ERRPRO	External routine
	ERRRPT	Well-defined module
	EXCFLAG	Well-defined module
	EXIT	External routine
	EXPRCLT	Well-defined module
	EXPRTL	Well-defined module
	FCLOSE	External routine
	FCOPATT	Well-defined module
	FCOPENT	Well-defined module
	FILEINS	Well-defined module
	FINDDOM	Well-defined module
	FMTIAUC	Well-defined module
	FND1MEM	Well-defined module
	FNDACH	Well-defined module
	FNDASA	Well-defined module
	FNDASH	Well-defined module
	FNDAUC	Well-defined module
	FNDECM	Well-defined module
	FNDOAC	Well-defined module
	FNDRCM	Well-defined module
	FREE	External routine
	FRTOREL	Well-defined module
	GDATA	External routine
	GENAKW	Well-defined module
	GENALI	Well-defined module
	GENALT1	Well-defined module
	GENALTE	Well-defined module
	GENATT	Well-defined module
	GENDESC	Well-defined module
	GENEKW	Well-defined module
	GENENT	Well-defined module
	GENENT1	Well-defined module
	GENOA	Well-defined module
	GENREL	Well-defined module

NDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	GENRKV	Well-defined module
	GENRNME	Well-defined module
	GETACAL	Well-defined module
	GETCHAR	External routine
	GETDBST	Well-defined module
	GETDOM	Well-defined module
	GETDRT	Well-defined module
	GETECAL	Well-defined module
	GETECNM	Well-defined module
	GETECS	Well-defined module
	GETGLOB	Well-defined module
	GETMAPC	Well-defined module
	GETNCHR	Well-defined module
	GETNNUM	Well-defined module
	GETNXNO	Well-defined module
	GETRCID	Well-defined module
	GETRCNM	Well-defined module
	GETRDH	Well-defined module
	HALT	Well-defined module
	ICOPATT	Well-defined module
	ICOPENT	Well-defined module
	INDFROM	Well-defined module
	INITCMD	Well-defined module
	INITEX	External routine
	INITFP	External routine
	INITRDL	Well-defined module
	INITSES	Well-defined module
	INSAC	Well-defined module
	INSACNM	Well-defined module
	INSAREA	Well-defined module
	INSAUC	Well-defined module
	INSAUCS	Well-defined module
	INSCRC	Well-defined module
	INSDAA	Well-defined module
	INSDB	Well-defined module
	INSDFLD	Well-defined module
	INSDI	Well-defined module
	INSDOM	Well-defined module
	INSDSL	Well-defined module
	INSDT	Well-defined module

NDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	INSEC	Well-defined module
	INSECNM	Well-defined module
	INSIAUC	Well-defined module
	INSISS	Well-defined module
	INSKC	Well-defined module
	INSKCM	Well-defined module
	INSKW	Well-defined module
	INSKWAC	Well-defined module
	INSKWEC	Well-defined module
	INSKWRC	Well-defined module
	INSMOD	Well-defined module
	INSOAC	Well-defined module
	INSPCB	Well-defined module
	INSPDF	Well-defined module
	INSPDI	Well-defined module
	INSPSB	Well-defined module
	INSPWRD	Well-defined module
	INSRC	Well-defined module
	INSRCRS	Well-defined module
	INSREUS	Well-defined module
	INSRKEY	Well-defined module
	INSRKM	Well-defined module
	INSRSET	Well-defined module
	INSRTYP	Well-defined module
	INSSCH	Well-defined module
	INSSDFL	Well-defined module
	INSSEC	Well-defined module
	INSSECR	Well-defined module
	INSSTM	Well-defined module
	KEYLOOK	Well-defined module
	LOADESC	Well-defined module
	LOGOFF	External routine
	LOGON	External routine
	LOWUPP	Well-defined module
	MAPADF	Well-defined module
	MAPASET	Well-defined module
	MAPRC	Well-defined module
	MIGREL	Well-defined module
	MKRNLST	Well-defined module
	MRGMOD	Well-defined module

MDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	MRGMOD1	Well-defined module
	MRGMOD2	Well-defined module
	MRGNODE	Well-defined module
	NEXTKC	Well-defined module
	NEXTKCM	Well-defined module
	NRGET	Well-defined module
	NRSTORE	Well-defined module
	OBINDN	External routine
	OCLOSE	External routine
	OCOF	External routine
	OCOM	External routine
	ODFINN	External routine
	OERMSG	External routine
	OEXEC	External routine
	OFETCH	External routine
	OISCR	External routine
	OLOGOF	External routine
	OLON	External routine
	OOPEN	External routine
	OPNFIL	External routine
	OPNFRM	External routine
	OROL	External routine
	OSQL3	External routine
	PIFROM	Well-defined module
	PDATA	External routine
	PDFDB	Well-defined module
	PDFDF	Well-defined module
	PDFREC	Well-defined module
	PDFSRCH	Well-defined module
	PMFROM	Well-defined module
	PMSGLS	External routine
	PNOFROM	Well-defined module
	PRCCMD	Well-defined module
	PRINTF	External routine
	PROCDT	Well-defined module
	PUTC	External routine
	RACKW2	Well-defined module
	RCCHEK	Well-defined module
	RCCHEK1	Well-defined module
	RDDESC	Well-defined module

NDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	RECKW2	Well-defined module
	RELKV	Well-defined module
	REVIEW	Well-defined module
	RENAME	Well-defined module
	RETACKV	Well-defined module
	RETECKV	Well-defined module
	RETRAC1	Well-defined module
	RETRACP	Well-defined module
	RETRCKV	Well-defined module
	RETREC1	Well-defined module
	RETRECP	Well-defined module
	ROLBACK	Well-defined module
	RPLFRM	External routine
	RRCKW2	Well-defined module
	SELACNM	Well-defined module
	SELECNM	Well-defined module
	SELIAUC	Well-defined module
	SELIKEY	Well-defined module
	SELRCNM	Well-defined module
	SELRSET	Well-defined module
	SELSTM	Well-defined module
	SMVIEW	Well-defined module
	SPRINTF	External routine
	STRCAT	External routine
	STRINS	Well-defined module
	STRLEN	External routine
	STRNCMP	External routine
	STRNCPY	External routine
	TERMFP	External routine
	TERMSER	Well-defined module
	TLOOPCK	Well-defined module
	TOLOWER	External routine
	TOPNODE	Well-defined module
	TOUPPER	External routine
	TRMNDML	External routine
	TXTTYP	Well-defined module
	UERROR	Well-defined module
	UPDAC	Well-defined module
	UPDACAL	Well-defined module
	UPDACNM	Well-defined module

NDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	UPDECAL	Well-defined module
	UPDECNM	Well-defined module
	UPDIND	Well-defined module
	UPDMNAM	Well-defined module
	UPDMOD	Well-defined module
	UPDNXNO	Well-defined module
	UPDRGNM	Well-defined module
	UPDTDOM	Well-defined module
	UPDTRC	Well-defined module
	UPDTKW	Well-defined module
	UPDTRC	Well-defined module
	UPDVIEW	Well-defined module
	UWARN	Well-defined module
	VERACDT	Well-defined module
	VERACNM	Well-defined module
	VERALI	Well-defined module
	VERAREA	Well-defined module
	VERASH	Well-defined module
	VERATT	Well-defined module
	VERAUC	Well-defined module
	VERCRC	Well-defined module
	VERDB	Well-defined module
	VERDBAS	Well-defined module
	VERDF	Well-defined module
	VERDFDT	Well-defined module
	VERDFLD	Well-defined module
	VERDI	Well-defined module
	VERDIDT	Well-defined module
	VERDOM	Well-defined module
	VERDSL3	Well-defined module
	VERDSTP	Well-defined module
	VERDSTX	Well-defined module
	VERDT	Well-defined module
	VERDTD	Well-defined module
	VERENT	Well-defined module
	VERKC	Well-defined module
	VERKW	Well-defined module
	VERKWE	Well-defined module
	VERKWR	Well-defined module
	VERMOD	Well-defined module

NDDL COMMAND PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	VERNMA	Well-defined module
	VERNME	Well-defined module
	VEROAC	Well-defined module
	VEROBJ	Well-defined module
	VERPDF	Well-defined module
	VERPSB	Well-defined module
	VERRC	Well-defined module
	VERRCBS	Well-defined module
	VERRCC	Well-defined module
	VERRCMP	Well-defined module
	VERRCNM	Well-defined module
	VERRCST	Well-defined module
	VERREL	Well-defined module
	VERRELS	Well-defined module
	VERRK	Well-defined module
	VERRKM	Well-defined module
	VERRSET	Well-defined module
	VERRT	Well-defined module
	VERSDT	Well-defined module
	VERSMS	Well-defined module
	VERTYP	Well-defined module
	VERUDTN	Well-defined module
	VERVIEW	Well-defined module
	VOMAPS	Well-defined module
	WRTACKW	Well-defined module
	WRTALI	Well-defined module
	WRTANAM	Well-defined module
	WRTDESC	Well-defined module
	WRTDSC4	Well-defined module
	WRTECKW	Well-defined module
	WRTENAM	Well-defined module
	WRTLIN	Well-defined module
	YYERROR	Well-defined module
	YYLEX	External routine
	YYPARSE	Well-defined module

3.10.8 Module Documentation

The following documentation describes information which is specific to each individual module being documented in this specification as listed in section 3.10.2. It provides a compact way of getting information that would be otherwise buried within each module's source code.

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

NAME:	Name of program Module.
PURPOSE:	Purpose of Module as detailed in the source code.
LANGUAGE:	Programming language source code is written in. The choices are: C VAX-11 FORTRAN (I/S-1 Workbench 'C') VAX-11 COBOL
MODULE TYPE:	Whether a Program, Subroutine, or Function.
SOURCE FILE:	Name of Source File from file specification.
SOURCE FILE TYPE:	Source File Extension from file specification.
HOST:	Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.
SUBSYSTEM:	IISS sub-system this file resides in.
SUBDIRECTORY:	Sub-directory of that subsystem in which this file resides.
DOCUMENTATION GROUP:	Name of documentation group of which this source file is a member.
DESCRIPTION:	A description of the module as obtained from the source code.

- ARGUMENTS:** The arguments with which this routine is called if it is a Subroutine or a Function.
- INCLUDE FILES:** A list of all the files that are included into this module as well as their purposes.
- ROUTINES CALLED:** Subroutines or Functions, either documented or external, called by this module, if any.
- CALLED DIRECTLY BY:** The documented routines which call this module, if any.
- USED IN MAIN PROGRAM(S):** The documented Main Programs which contain this module in their parts list according to the list in section 3.10.7.

The Module Documentation is arranged alphabetically according to Module Name.

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDATT
PURPOSE: ASSOCIATES EXISTING ATT WITH ENTITY IN
GREAT ENTITY COMMAND
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDATT
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

ADDATT ASSOCIATES AN EXISTING ATTRIBUTE WITH AN
ENTITY CLASS IN THE CREATE ENTITY COMMAND.

ARGUMENTS:

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

INCLUDE FILES:

LISTNOS - VALID LIST NUMBERS

ROUTINES CALLED:

CPHONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
ADDOAC - ADD ATTRIBUTE AS AN OWNED ATTRIBUTE AND AS ATT
USE CLASS
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.

CALLED DIRECTLY BY:

PS 620141100
1 November 1985

ALTENT - CONTROL PROCESSING FOR ALTER ENTITY CLASS
COMMAND.
CRTENT - CONTROL THE PROCESSING LOGIC FOR CREATING A NEW
ENTITY CLASS.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

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

DESCRIPTION:

THE ADD DATA TYPE CLAUSE ADDS THE DATA TYPE NAME,
TYPE, SIZE AND LENGTH OF FIELD, GIVEN THE DOMAIN
NUMBER

SELECT THE DATA TYPE NAME, ID, SIZE,
NAMES OF DECIMALS, AND WHETHER THE TYPE
IS STANDARD OR USER-DEFINED. ADD THE
PARTICULAR TYPE GIVEN THE DOMAIN NUMBER.
REPEAT FOR EACH DATA TYPE TO BE ADDED.

ARGUMENTS:

DOM-NO, DATA-TYPE-NAME, TYPE-ID -
MAX-SIZE, NO-OF-DEC, STD-USER -

ROUTINES CALLED:

ADDSTD - PROGRAM NAME
ADDNSTD - ADDS A USER DEFINED DATA TYPE
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

CALLED DIRECTLY BY:

PROCDT - PROGRAM NAME

USED IN MAIN PROGRAM(S):

PS 620141100
1 November 1985

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDEC
PURPOSE: ADD THE ENTITY NAME TO THE TREE LIST
STRUCTURE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: AKCROW
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C -- ADDEC(SEC_NO)
COBOL -- CALL "ADDEC" USING
EC-NO.
FORTRAN -- CALL ADDEC(ECNO)

INPUT:
INT *EC_NO ;

OUTPUT:

DESCRIPTION:
ADD EC_NO INTO KEYLIST TREE DATA STRUCTURE

ARGUMENTS:

EC_NO - INT *

INCLUDE FILES:

KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

PS 620141100
1 November 1985

ROUTINES CALLED:

ALLOC

CALLED DIRECTLY BY:

**AKCROW - THIS ROUTINE ADDS A ROW TO THE UNBOUNDED
KEY_CLASS_LIST**

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: ADDECNM
PURPOSE: ADD THE EC_NAME AND EC_NO INTO KEYLIST
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: BLECLST
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C -- ADDECNM(EC_NO,EC_NAME)
COBOL -- CALL "ADDECNM" USING

EC_NO,

EC_NAME.

FORTRAN -- CALL ADDECNM (EC_NO,EC_NAME)

INPUT:

INT *EC_NO ;
CHAR *EC_NAME ;

OUTPUT:

DESCRIPTION

THIS ROUTINE ADDS THE EC_NO AND EC_NAME TO THE
KEYLIST

DATA STRUCTURE

ARGUMENTS:

EC_NO = INT *

PS 620141100
1 November 1985

EC_NAME - CHAR *

INCLUDE FILES:

KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

ROUTINES CALLED:

ALLOC
STRNCPY

CALLED DIRECTLY BY:

BLECLST - THIS ROUTINE CREATES A ROW IN THE UNBOUNDED
EC_LIST

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDKC
PURPOSE: CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR
CREATE/ALTER ENTI
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDKC
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

ADDKC CONTROLS THE PROCESSING OF THE KEY CLASS
CLAUSE FOR THE CREATE AND ALTER ENTITY COMMANDS.
KEY CLASS CLAUSE CONSISTS OF AN KEY CLASS
NAME, FOLLOWED BY AN OPTIONAL LIST OF ATTRIBUTE USE
CLASS NAMES. THE CLAUSE IS RECURSIVE. KEY CLASS NAME
IS ON A-KC-LST, NO. OF ATTRIBUTE FOR KEY CLASS IN
A-NUM-LST AND LIST OF ATTRIBUTE IN A-AUC-LST.

ARGUMENTS:

MODEL-NO = DSPLY [S9(9)]
EC-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
CPFVAL - RETURN THE COUNTER OF LIST1 BASED ON ROW IN
LIST2
VERKC - VERIFY THE EXISTENCE OF A KEY CLASS IN A MODEL
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
GETNNUM -

PS 620141100
1 November 1985

INSKC - INSERT A RECORD INTO KEY CLASS
CPFNYT - ACCESS THE NEXT TOKEN IN A PARSER LIST.
ADDKH - ADDS SINGLE KEY CLASS MEMBER(AUC) TO KEY CLASS
FOR ADD KEY

CALLED DIRECTLY BY:

ALTENT - CONTROL PROCESSING FOR ALTER ENTITY CLASS
COMMAND.
CRTENT - CONTROL THE PROCESSING LOGIC FOR CREATING A NEW
ENTITY CLASS.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDKCLS
PURPOSE: ADD KEY INFO TO THE UNBOUNDED
KEY_CLASS_LIST STRUCTURE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: AKCROW
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C -- ADDKCLS(%KC_NO, %KC_NAME)
COBOL -- CALL "ADDKCLS" USING
KC-NO,
KC-NAME,
FORTRAN -- CALL ADDKCLS(KCNO, KCNAME)

INPUT:

INT *KC_NO ;
CHAR *KC_NAME[30] ;

OUTPUT:

DESCRIPTION:

THIS ROUTINE ADDS KC_NO, KC_NAME TO THE UNBOUNDED
KEY_CLASS_LIST

ARGUMENTS:

KC_NO = INT *
KC_NAME = CHAR *

INCLUDE FILES:

KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

PS 620141100
1 November 1985

ROUTINES CALLED:

ALLO
STRNCPY

CALLED DIRECTLY BY:

AKCROW - THIS ROUTINE ADDS A ROW TO THE UNBOUNDED
 KEY_CLASS_LIST

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: ADDKG
PURPOSE: ADD KCM_TAG NUMBER AND NAME TO STRUCTURE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: AKCROW
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

```
C          --  ADDKG(%EC NO, %KCM_TAG_NO,  
              %KCM_TAG_NAME) ;  
  
COBOL     --  CALL "ADDEG"  USING  
              KCM-TAG-NO,  
              KCM-TAG-NAME.  
  
FORTRAN   --  CALL ADDEG( KCMTAGNO, KCMTAGNAME)
```

INPUT:

```
INT  *KCM_TAG_NO ;  
CHAR *KCM_TAG_NAME[30] ;
```

OUTPUT:

DESCRIPTION:

THIS ROUTINE ADDS KEY CLASS MEMBER INFO TO THE UNBOUNDED
TO THE KEY CLASS LIST TREE STRUCTURE

ARGUMENTS:

```
KCM_TAG_NO -      INT *  
KCM_TAG_NAME -    CHAR *
```

INCLUDE FILES:

PS 620141100
1 November 1985

KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

ROUTINES CALLED:

ALLOC
STRNCPY

CALLED DIRECTLY BY:

AKCROW - THIS ROUTINE ADDS A ROW TO THE UNBOUNDED
KEY_CLASS_LIST

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDKM
PURPOSE: ADDS SINGLE KEY CLASS MEMBER(AUC) TO KEY CLASS FOR ADD KEY
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDKM
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

ADDKM ADDS A SINGLE KEY CLASS MEMBER (ATTRIBUTE USE CLASS) TO A KEY CLASS. USED BY ANY COMMANDS CONTAINING AN ADD EKY CLASS CLAUSE.

ARGUMENTS:

MODEL-NO = DSPLY [S9(9)]
AUC-NAME = DSPLY [X(30)]
KC-NO = DSPLY [S9(9)]
EC-NO = DSPLY [S9(9)]

ROUTINES CALLED:

VERAUC - VERIFY THE EXISTENCE OF AN ATTRIBUTE USE CLASS
ADDOAC - ADD ATTRIBUTE AS AN OWNED ATTRIBUTE AND AS ATT USE CLASS
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
INSKCM - INSERT A RECORD INTO KEY_CLASS_MEMBER

CALLED DIRECTLY BY:

ADDKC - CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR CREATE/ALTER ENTITY

PS 620141100
1 November 1985

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDKW
PURPOSE: ADDS KEYWORDS FOR COMMANDS USING "ADD
KEYWORD"(OPTIONAL)
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDKW
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

ADDKW PROCESSES THE "ADD KEYWORD" CLAUSE FOR
ANY COMMAND USING KEYWORDS. THE CLAUSE IS
OPTIONAL AND MAY NOT EXIST.

ARGUMENTS:

OBJECT-TYPE = RECRD
OBJ-ID-NO = DSPLY [S9(9)]
RTN-STATUS = 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
LOWUPP - CONVERT A STRING TO UPPER CASE CHARACTERS
VERKW - verify the existence of a keyword.
GETNNUM -
INSKW - INSERT A RECORD INTO KEYWORD
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ADDKWA - INSERT AN ATTRIBUTE KEYWORD
ADDKWE - INSERT AN ENTITY KEYWORD

PS 620141100
1 November 1985

ADDKWR - INSERT A RELATION KEYWORD
CPPWXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.

CALLED DIRECTLY BY:

ALTENT - CONTROL PROCESSING FOR ALTER ENTITY CLASS
COMMAND.
ALTREL - CONTROLS PROCESSING LOGIC FOR THE ALTER
RELATION COMMAND
CRTATT - CONTROLS THE PROCESSING LOGIC FOR CREATING AN
ATTRIBUTE.
CRTENT - CONTROL THE PROCESSING LOGIC FOR CREATING A NEW
ENTITY CLASS.
CRTREL - CONTROLS THE LOGIC FOR VALIDATING AND CREATING
A NEW RELATION CLA
ALTATT - THE ALTER ATTRIBUTE COMMAND PROCESSOR ALTERS

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: ADDKWA
PURPOSE: INSERT AN ATTRIBUTE KEYWORD
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDKWA
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CHECKS FOR THE PRESENCE OF A KEYWORD ON AN
ATTRIBUTE, AND IF NOT FOUND, INSERTS IT.

ARGUMENTS:

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

INCLUDE FILES:

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

ROUTINES CALLED:

INSKWAC - INSERT A RECORD INTO AC KEYWORD
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ERRPRO

CALLED DIRECTLY BY:

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

PS 620141100
1 November 1985

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: ADDKWE
PURPOSE: INSERT AN ENTITY KEYWORD
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDKWE
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CHECKS FOR THE PRESENCE OF A KEYWORD ON AN
ENTITY, AND IF NOT FOUND, INSERTS IT.

ARGUMENTS:

KW-NO = DSPLY [S9(9)]
EC-NO = DSPLY [S9(9)]
RETURN-CODE = DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

INSKVEC - INSERT A RECORD INTO EC KEYWORD
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ERRPRO

CALLED DIRECTLY BY:

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

PS 620141100
1 November 1985

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDKWR
PURPOSE: INSERT A RELATION KEYWORD
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDKWR
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CHECKS FOR THE PRESENCE OF A KEYWORD ON A
RELATION CLASS, AND IF NOT FOUND, INSERTS IT.
-

ARGUMENTS:

KW-NO = DSPLY [S9(9)]
RC-NO = DSPLY [S9(9)]
RETURN-CODE = DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

INSKWR - INSERT A RECORD INTO RC_KEYWORD
ERRPRO

CALLED DIRECTLY BY:

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

PS 620141100
1 November 1985

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDMAP
PURPOSE: ADD A CS-IS MAPPING
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ADDMAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

```
C          --  ADDMAP (MAP_TYPE, &TAG_NO, &RC_NO,  
              &AC_NO,&SM_FLAG)
```

```
COBOL     --  CALL  "ADDMAP"      USING  
                                              MAP-TYPE  
                                              TAG-NO  
                                              RC-NO  
                                              AC-NO  
                                              SM-FLAG.
```

```
FORTRAN   --  CALL  ADDMAP (MAPTYPE, TAGNO, RCNO,  
              ACNO, SMFLAG)
```

INPUT:

```
CHAR MAP_TYPE[]  
INT *TAG_NO  
INT *RC_NO  
INT *AC_NO  
INT *SM_FLAG;
```

OUTPUT:

DESCRIPTION

ADDMAP ADDS CS-IS MAPPING TO THE CDM AFTER EXTRACTING
VARIOUS TOKENS
FROM THEIR LISTS AND VERIFYING THAT THE DATABASE NAME
EXISTS. IF
SM-FLAG EQUALS ZERO, PROCESS THE COMPLETE LIST,
OTHERWISE PERFORM
1 MAPPING.

ARGUMENTS:

MAP_TYPE - CHAR []
TAG_NO - INT *
RC_NO - INT *
AC_NO - INT *
SM_FLAG - INT *

INCLUDE FILES:

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

ROUTINES CALLED:

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
TOLOWER
STRNCMP
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.
CPFCOR - ACCESS A TOKEN IN A CORRESPONDING NAMED LIST
MAPRC - MAP A RELATION CLASS TO A SET
MAPADF - MAP ON AUC TO A DATA FIELD
SPRINTF
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
MAPASET - MAP AN AUC TO A SET
TOUPPER
VERDB - VERIFY THE EXISTENCE OF A DATA BASE IN THE CDM

CALLED DIRECTLY BY:

ALTHAP - ALTER MAP COMMAND PROCESSOR
CRTMAP - CREATE MAP COMMAND PROCESSOR

PS 620141100
1 November 1985

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

DDL COMMAND PROCESSOR Module Documentation

NAME: ADDMIG
PURPOSE: PROCESS THE ADD MIGRATES...SET..CLAUSE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDMIG
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CONTROLS THE "ADD MIGRATES" CLAUSE OF THE CREATE
AND ALTER RELATION COMMANDS. MIGRATES THE KEY
CLASS FROM THE INDEPENDENT TO THE DEPENDENT
ENTITY CLASS. THE MIGRATES CLAUSE IS OPTIONAL.
THE CLAUSE CONSISTS OF A KEY-CLASS NAME AND AN
OPTIONAL SET CLAUSE.

-

ARGUMENTS:

MODEL-NO = DSPLY [S9(9)]
RC-NO = DSPLY [S9(9)]
IND-EC-NO = DSPLY [S9(9)]
DEP-EC-NO = DSPLY [S9(9)]
RETURN-CODE = DSPLY [S9(9)]

INCLUDE FILES:

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR
LISTNOS - VALID LIST NUMBERS
UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS
RENLIST - LIST OF ATTRIBUTES AND INHERITED TAG PAIRS
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

PS 620141100
1 November 1985

CPFOME	- 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.
MKRNLST	- FETCH LIST OF RENAME PAIRS FOR MIGRATES..SET..CLAUSE
GKRNLST	- SEARCHES THE TABLE OF RENAME PAIR LOOKING FOR AN OLD-TAG ENTRY.
VERAUC	- VERIFY THE EXISTENCE OF AN ATTRIBUTE USE CLASS
GETNUM	-
INSAUC	- INSERT A RECORD INTO ATTRIBUTE_USE_CL
INSIAUC	- INSERT A RECORD INTO INHERITED_ATT_USE
INSCRC	- INSERT A RECORD INTO COMPLETE_RELATION
ERRPRO	

CALLED DIRECTLY BY:

ALTREL	- CONTROLS PROCESSING LOGIC FOR THE ALTER RELATION COMMAND
CRTREL	- CONTROLS THE LOGIC FOR VALIDATING AND CREATING A NEW RELATION CLA

USED IN MAIN PROGRAM(S):

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

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDNOCORR
PURPOSE: ADD A TOKEN TO A CORRESPONDING LISTS NEXT ENTRY
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 -- ADDNOCORR(CORR_LST, LST, TOKEN)

COBOL -- CALL "ADDNOCORR" USING

CORR-LST,

LIST,

TOKEN)

FORTRAN -- CALL ADDNOCORR(CLIST, LIST, TOKEN)

INPUT:

LST1

--

THE CORRESPONDING LIST, ITS LAST ENTRY INDEX WILL BE USED TO DETERMINE THE INDEX OF LST2

LST2 --

THE LIST TO WHICH THE TOKEN WILL BE ADDED

TOKEN

--

THE CHARACTER STRING REPRESENTING THE
TOKEN ISOLATED BY THE PARSER

OUTPUT:

DESCRIPTION

ADD A STRING TO THE LIST IDENTIFIED BY LST2 IN THE POSITION
CORRESPONDING TO THE POSITION AFTER THE LAST USED ENTRY
OF THE LIST IDENTIFIED BY LST1.

ARGUMENTS:

LST1 - INT
LST2 - INT
STRING - CHAR *

INCLUDE FILES:

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

ROUTINES CALLED:

STRLEN
STRCAT

CALLED DIRECTLY BY:

YYPARSE - **** PURPOSE NOT FOUND BY STRIPPER ****

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADMSTD
PURPOSE: ADDS A USER DEFINED DATA TYPE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADMSTD
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE ADDS A "USER" DEFINED DATA TYPE

AFTER VERIFYING THAT THE DATA TYPE
TO BE ADDED DOES NOT ALREADY EXIST,
CHECK IF A LEGAL DATA TYPE, WITH
CORRECT DECIMAL SPECIFICATIONS BEFORE
INSERTING, AS A "USER" DEFINED DATA
TYPE.

ARGUMENTS:

DOM-NO = DSPLY [S9(9)]
DATA-TYPE-NAME = DSPLY [X(30)]
TYPE-ID = DSPLY [X]
MAX-SIZE = DSPLY [S9(9)]
NO-OF-DEC = DSPLY [S9(9)]
RETURN-STATUS = DSPLY [S9(9)]

ROUTINES CALLED:

VERDT - VERIFY THE EXISTENCE OF A USER DEFINED DATA TYPE
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
VERTYP - verifies that a type identification for a data
type is valid.
INSDT - INSERT A RECORD INTO USER_DEF_DATA_TYPE

CALLED DIRECTLY BY:

PS 620141100
1 November 1985

ADDDT - PROGRAM NAME

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDOAC
PURPOSE: ADD ATTRIBUTE AS AN OWNED ATTRIBUTE AND
AS ATT USE CLASS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDOAC
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

GIVEN AN ATTRIBUTE NAME, ADD IT AS AN OWNED
ATTRIBUTE AND AS AN ATTRIBUTE USE CLASS.

ARGUMENTS:

MODEL-NO = DSPLY [S9(9)]
EC-NO = DSPLY [S9(9)]
AC-NAME = DSPLY [X(30)]
NEW-TAG-NO = DSPLY [S9(9)]

INCLUDE FILES:

UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A
MODEL
VEROAC - verify the existence of an owned attribute
class for an entity.
VERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
INSOAC - INSERT A RECORD INTO OWNED_ATTRIBUTE

PS 620141100
1 November 1985

GETNUM -
INSAUC - INSERT A RECORD INTO ATTRIBUTE_USE_CL
ERRPRO

CALLED DIRECTLY BY:

ADDATT - ASSOCIATES EXISTING ATT WITH ENTITY IN CREAT
ENTITY COMMAND
ADDKM - ADDS SINGLE KEY CLASS MEMBER(AUC) TO KEY CLASS
FOR ADD KEY
COPYAC - CREATE AN ATTRIBUTE, ASSOCIATE WITH ENTITY, ADD
KEY CLASSES

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDPARM
PURPOSE: WRITES 80 CHAR NDDL COMMAND WITH
PARAMETERS CHECKS/DELIMIT
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDPARM
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

Strings a parameter and a delimiter into an
80-character NDDL command line. Performs line overflow
checking and indentation of continued command lines.
Writes a command line if:
1) routine is called with delimiter = semicolon
(end of command)
2) parameter will not fit on current command line
3) routine is called with parameter and delimiter both blank

ARGUMENTS:

TEXT-PARM = DSPLY [X(60)]
DELIMTR = DSPLY [X]

ROUTINES CALLED:

WRTLIN - THIS ROUTINE WRITES A NDDL COMMAND LINE (80
CHARACTERS)

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.
- CMBRKW** - SELECT AND GENERATE RELATION CLASS KEYWORDS
- CPYMOD** - CONTROLS THE PROCESSING LOGIC FOR THE COPY MODEL COMMAND.
- DEPFROM** - GENERATE CREATE RELATION, DESCRIBE COMMANDS IN THE TO-MODEL
- FCOPATT** - GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE COMMAND
- FMTIAUC** - FORMATS A LINE FOR THE ..SET.. CLAUSE
- FRTOREL** - DETERMINE IF RELATION EXISTS BETWEEN INTRA-MOD ENTITIES
- GENAKW** - RETRIEVE ALL KEYWORDS FOR AN ATTRIBUTE CLASS
- GENALI** - GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT TYPE
- GENALT1** - GENERATE AN ALTER ENTITY..ADD KEY CLASS COMMAND
- GENALTE** - GENERATE AN ALTER ENTITY..ADD KEY CLASS COMMAND
- GENATT** - GENERATE A CREATE ATTRIBUTE.. COMMAND
- GENDESC** - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN OBJECT TYPE AND NO
- GENEKW** - SELECT KEYWORD FOR ENTITY AND CREATE KEYWORD PHRASE FOR CRT ENT
- GENENT** - GENERATE A CREATE ENTITY..OWNED ATTRIBUTE..KEYWORD COMMAND
- GENOA** - SELECT OWNED ATT FOR ENTITY AND CREATES OWNED ATT FOR CRT ENT
- GENREL** - GENERATE CREATE RELATION ..MIGRATES..KEYWORD COMMAND
- GENRKW** - SELECT KEYWORDS FOR RELATION CREATES KW PHRASE FOR CREATE RC
- GENRNME** - FORMAT THE CREATE RELATION CLAUSE
- INDFROM** - RETRIEVES RELATIONS, DETERMINES IND EC AND GENERATES NDDL
- MIGREL** - GENERATE A MIGRATES CLAUSE FOR A CREATE RELATION COMMAND
- MRGMOD** - MERGE TWO IDEF MODELS INTO ONE
- MRGMOD2** - CONTROL THE LOGIC FOR PROCESSING THE REMAINING MODEL_2

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDRCEC
PURPOSE: POPULATES THE RC-DEPKC TABLE FOR ALL
RELATIONS IN THE MODEL
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADDRCEC
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

-
ADDRCEC populates the RC-DEPKC table.
The table contains information for every relation in a model
in sequence by level, rc-no, and dep-ec-no. For each rc-o
and dep-ec-no added to the table, an NDML routine is
called to select all the key classes for the dep-ec
which inherit attributes via the migrated relation (rc-no).

-
ADDRCEC populates the RC-DEPKC table.
The table contains information for every relation in a model
in sequence by level, rc-no, and dep-ec-no. For each rc-o
and dep-ec-no added to the table, an NDML routine is
called to select all the key classes for the dep-ec
which inherit attributes via the migrated relation (rc-no).

ARGUMENTS:

MOD-NO = DSPLY [S9(9)]
MOD-NAME = DSPLY [X(30)]
RC-NO = DSPLY [S9(9)]
DEP-EC-NO = DSPLY [S9(9)]
RC-DEPKC-LIST = RECRD
RTN-STATUS = DSPLY [S9(9)]

PS 620141100
1 November 1985

INCLUDE FILES:

KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS
RCDEPKC - LIST OF KEYS MIGRATED VIA A RELATION

ROUTINES CALLED:

SELIKEY - RETRIEVE ALL INHERITED KEYS FOR A GIVEN EC AND
RC
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

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

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADDRNUM
PURPOSE: ADD A AVAILABLE NO OF A POOL NO BACK TO
NO LINKED
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ADDRNUM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

LIST DATA STRUCTURE.

SYNOPSIS

C	--	ADDRNUM(PPOOL_NO, AVAIL_NO)	
COBOL	--	CALL "ADDRNUM"	USING POOL-NO, AVAIL-NO.
FORTRAN	--	CALL ADDRNUM(PPOOLNO, AVAILNO)	

INPUT:

INT *POOL_NO ;
INT *AVAIL_NO ;

OUTPUT:

DESCRIPTION

THIS ROUTINE PUT THE AVAILABLE NO OF THE POOL NO BACK
TO THE NO
LINKED LIST. THE ROUTINE CALLS 'ALLOC' FUNCTION TO
ALLOCATE SPACE
FOR STORING THE AVAIL NO AND POOL NO.

ARGUMENTS:

POOL_NO = INT *

AVAIL_NO = INT *

ROUTINES CALLED:

NRSTORE - STORE A AVAILABLE NO OF A POOL NO BACK TO NO
LINKED
NRPRO

CALLED DIRECTLY BY:

DELDDDF - DELETE ALL DATA FIELD ASSOCIATIONS WITH THE
DATABASE
DELDDST - DELETE ALL RECORD TYPE ASSOCIATIONS WITH THE
DATABASE
DELDDST - DELETE ALL RECORD SETS' ASSOCIATIONS WITH THE
DATABASE
DELDTBO - DELETE ALL DATA TYPES ASSOCIATED WITH A DOMAIN
DELDDDC - DELETE ALL KEY CLASSES AND INHERITED KEYS FOR
AN ENTITY
DELDDRC - DELETE ALL RELATION CLASSES FOR AN ENTITY
DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED
ATTRIBUTE CLASSES FROM THE
DRPDS - CONTROLS THE PROCESSING LOGIC FOR DELETING THE
DATA BASE.
DRPDIV - DELETE DATA ITEM DESCRIPTION TEXTS ASSOCIATED
WITH VIEW
DRPDON - PROGRAM NAME
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING
ENTITIES.
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A
DATA FIELD.
DRPEC - CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY
CLASS".
DRPEVC - OBTAIN THE USED IDENTIFIED KEYWORD. THEN DROP
THEIR ASSOCIATIONS.
DRPHOD - CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.
DRPCE - 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 DATABASES
DRPVIEW - DROP THE VIEW

PS 620141100
1 November 1985

FNDACH - DELETE ALL ATTRIBUTE CLASSES FROM A GIVEN MODEL
FNDECM - DELETES ALL ENTITIES AND ASSOCIATED OBJECTS FOR
A GIVEN MODEL
SELRSET - DELETE ALL ASSOCIATIONS IN THE CDM FOR A SET
BEING DROPPED
SELSTH - DELETES ALL OWNER/MEMBERS OF THE RECORD TYPE
BEING DROPPED
DELAUC - DELETE INHERITED ATT USE, KEY CLASS MEMBER,
DELDPL2 - CONTROLS THE DELETING OF DATA FIELDS
DELNICK - DELETE MIGRATING KEY CLASS
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
DLNIGRC - DELETE MIGRATING KEY CLASS

USED IN MAIN PROGRAM(S):

MDDL/MAIN - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

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

DESCRIPTION:

THIS ROUTINE ADDS THE "STD" DATA TYPE
FOR A DOMAIN
CLASS.

VERIFY THAT THE DATA TYPE IS DEFINED AS
"STD", THEN MAKE SURE THAT THE DATA-TYPE
NAME TO BE ADDED DOES NOT PREVIOUSLY
EXIST. NOW CHECK IF IT IS A LEGAL TYPE
AND HAS CORRECT DECIMAL SPECIFICATIONS.
NOW INSERT THE DATA TYPE.

ARGUMENTS:

DOM-NO - DPLY [9(9)]
DATA-TYPE-NAME - DPLY [X(30)]
TYPE-ID - DPLY [X]
MAX-SIZE - DPLY [9(9)]
NO-OF-DEC - DPLY [9(9)]
RETURN-STATUS - DPLY [9(9)]

ROUTINES CALLED:

VERSDT - For a given domain number, return its standard
data type name.
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
VERDT - VERIFY THE EXISTENCE OF A USER DEFINED DATA TYPE
VERTYP - verifies that a type identification for a data
type is valid.

PS 620141100
1 November 1985

INSDT - INSERT A RECORD INTO USER_DEF_DATA_TYPE

CALLED DIRECTLY BY:

ADDOT - PROGRAM NAME

USED IN MAIN PROGRAM(S):

WDDL/MAIN - MAIN PROGRAM FOR THE WDDL COMMAND PROCESSOR

HDDL COMMAND PROCESSOR Module Documentation

NAME: ADDTXT
PURPOSE: THIS ROUTINE WRITES 80 CHARACTERS OF
DESCRIPTIVE TEXT
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ADDTXT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

TO THE USER'S FILE OF GENERATED HDDL COMMANDS.
THE
FIRST LINE OF TEXT IS PRECEDED BY A DOUBLE
QUOTE
MARK IN COLUMN 1 AND A NEWLINE CHARACTER IN
COLUMN 2.
THE LAST LINE OF TEXT IS FOLLOWED BY THE SAME
THING.

NOTE -- THE DESCRIBE COMMAND PROCESSOR WILL
INCLUDE
THE TRAILING BLANKS BETWEEN THE QUOTE
MARK
AND THE END OF THE LINE AS PART OF THE
DESCRIPTIVE TEXT IF THE NEWLINE
CHARACTER
IS OMITTED.

SYNOPSIS

C -- ADDTXT (TXT-LINE, FLAG)

COBOL -- CALL "ADDTXT" USING TXT-LINE
QUOTE-FLAG.

FORTRAN -- CALL ADDTXT (TXTLIN, FLAG)

INPUT:

CHAR *TIT-LINE[80]
INT FLAG

OUTPUT:

ARGUMENTS:

TIT LINE - CHAR *
FLAG - INT *

ROUTINES CALLED:

VRTLIN - THIS ROUTINE WRITES A NDDL COMMAND LINE (80
CHARACTERS)

CALLED DIRECTLY BY:

GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN
OBJECT TYPE AND NO

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

HDDL COMMAND PROCESSOR Module Documentation

NAME: ADD CORR
PURPOSE: ADD A TOKEN TO CORRESPONDING 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 -- ADD_CORR(CORR_LIST, LIST, TOKEN)

COBOL -- CALL "ADD_CORR" USING

CORR-LIST,

LIST,

TOKEN.

FORTRAN

CALL ADD_CORR(CLIST, LIST, TOKEN) --

INPUT:

LST1 - THE INTEGER IDENTIFYING THE CORRESPONDING LIST,
NOT THE LIST TO WHICH THE TOKEN WILL
BE ADDED TO.

LST2 - THE INTEGER IDENTIFYING THE LIST TO WHICH THE
TOKEN WILL BE ADDED TO

STRING - THE CHARACTER STRING REPRESENTING THE TOKEN
TO BE ADDED TO THE END OF LST2.

OUTPUT:

DESCRIPTION

ADD THE TOKEN TO THE TOKEN ACCUMULATOR STRING AND
GET ITS LENGTH. THEN USING THE CURRENT ROW INDEX
OF THE CORRESPONDING LIST, STORE THE POINTERS
TO THE ACCUMULATOR STRING IN THE LIST DATA STRUCTURE

ARGUMENTS:

LST1 - INT
LST2 - INT
STRING - CHAR *

INCLUDE FILES:

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

ROUTINES CALLED:

STRLEN
STRCAT

CALLED DIRECTLY BY:

YYPARSE - **** PURPOSE NOT FOUND BY STRIPPER ****

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADD_TO_CNT
PURPOSE: INCREMENT A LIST COUNTER
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 -- ADD_TO_CNT(LIST1, LIST2)

COBOL -- CALL "ADD_TO_CNT" USING

LIST1,

LIST2,

FORTRAN -- CALL ADD_TO_CNT (LIST1, LIST2)

INPUT:

LST1 -- THE LIST WHICH WILL BE USED AS A COUNTER
LST2 -- THE LIST WHICH GIVES THE CURRENT INDEX INTO
LIST1

OUTPUT:

DESCRIPTION

PS 620141100
1 November 1985

AN ENTRY IN LIST1 WILL BE A COUNTER TO THE
NUMBER OF TIMES THIS ROUTINE IS CALLED.
THE CURRENT ROW OF LIST 2 WILL BE USED AS
THE INDEX INTO LIST1.

ARGUMENTS:

LIST1 - INT
LIST2 - INT

INCLUDE FILES:

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

ROUTINES CALLED:

ERRPRO

CALLED DIRECTLY BY:

YYPARSE - **** PURPOSE NOT FOUND BY STRIPPER ****

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADD_TO_LST
PURPOSE: ADD A SINGLE TOKEN TO A PARSER OUTPUT 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 -- ADD_TO_LST(LIST, TOKEN)
COBOL -- CALL "ADD_TO_LST" USING

LIST-NO.

TOKEN.

FORTRAN

-- CALL ADD_TO_LST(LIST, TOKEN)

INPUT:

LIST - AN INTEGER IDENTIFYING THE LIST TO WHICH THE
TOKEN SHOULD BE ADDED

TOKEN - A CHARACTER STRING REPRESENTING A SINGLE
TOKEN AS ISOLATED BY THE PARSER

OUTPUT:

DESCRIPTION

THE GIVEN TOKEN WILL BE ADDED TO THE END OF THE SPECIFIED

LIST BY PLACING IT IN A TOKEN ACCUMULATOR STRING AND
STORING THE BEGINNING CHARACTER POSITION AND LENGTH
IN THE PROPER PLACE IN THE LIST DATA STRUCTURE

ARGUMENTS:

LST - INT
STRING - CHAR *

INCLUDE FILES:

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

ROUTINES CALLED:

ERRPRO
STLEN
STRCAT

CALLED DIRECTLY BY:

YYPARSE - **** PURPOSE NOT FOUND BY STRIPPER ****

USED IN MAIN PROGRAM(S):

HDCL/MAIN - MAIN PROGRAM FOR THE HDCL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ADPARN1
PURPOSE: CREATES 80 CHARACTER NDDL COMMANDS WITH
PARAMETERS AND DELIMITER
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ADPARN1
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDN
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

Strings a parameter and a delimiter into an
80-character NDDL command line. Performs line overflow
checking and indentation of continued command lines.
Writes a command line if:
1) routine is called with delimiter - semicolon
(end of command)
2) parameter will not fit on current command line
3) routine is called with parameter and delimiter both blank

Note: This routine is merely a copy of "ADDPARM". The
purpose of the copy is to allow multiple commands
to be constructed at the same time.

ARGUMENTS:

TEXT-PARM - DISPLAY [X(60)]
DELINTR - DISPLAY [X]

ROUTINES CALLED:

WRTLIN - THIS ROUTINE WRITES A NDDL COMMAND LINE (80
CHARACTERS)

CALLED DIRECTLY BY:

PS 620141100
1 November 1965

CMBEKV - GENERATE ADD KEYWORD CLAUSE FOR ENTITY KEYWORDS
CMBOA - GENERATE COMMANDS FOR ATTRIBUTES, ITS
KEYWORDS, ALIAS, DESC
GEMENT1 - GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND
DESCRS.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

DDL COMMAND PROCESSOR Module Documentation

NAME: AKCROW
PURPOSE: THIS ROUTINE ADDS A ROW TO THE UNBOUNDED
KEY_CLASS_LIST
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: AKCROW
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

```
C          -- AKCROW(SEC_NO, SEC_NO, SEC_NAME,  
          SEC_TAG_NO,  
          KCM_TAG_NAME) ;  
  
COBOL     -- CALL "AKCROW" USING  
          EC-NO,  
          KC-NO,  
          KC-NAME,  
          KCM-TAG-NO,  
          KCM-TAG-NAME.  
  
FORTRAN   -- CALL AKCROW(ECNO, KCNO, KCNAME,  
          KCMTAGNO, KCMTAGNAME)
```

INPUT:

```
INT *EC_NO ;  
INT *KC_NO ;  
CHAR *KC_NAME[30] ;  
INT *KCM_TAG_NO ;  
CHAR *KCM_TAG_NAME[30] ;
```

OUTPUT:

DESCRIPTION:

THIS ROUTINE ADDS A ROW TO THE UNBOUNDED KEY_CLASS_LIST

ADD EC_NO, TAG_NO AND EOM_TAG_NO

ARGUMENTS:

EC_NO - INT *
EC_NO - INT *
EC_NAME - CHAR *
EOM_TAG_NO - INT *
EOM_TAG_NAME - CHAR *

INCLUDE FILES:

KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

ROUTINES CALLED:

SPRINTF
PRINTF
ADDEG - ADD EOM_TAG NUMBER AND NAME TO STRUCTURE
ADDEC - ADD THE ENTITY NAME TO THE TREE LIST STRUCTURE
ADDEKLS - ADD KEY INFO TO THE UNBOUNDED KEY_CLASS_LIST
STRUCTURE

CALLED DIRECTLY BY:

BLKCL1 - STORE ALL KEY CLASS INFO FOR AN ENTITY IN A LIST
BLKCLST - SELECT AND STORE KEY CLASS INFO FOR A GIVEN
ENTITY
DPKCLST - CREATE AN KEY_CLASS_LIST TABLE CONTAINING ALL
THE ENTITY

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

DDL COMMAND PROCESSOR Module Documentation

NAME: ALLATT
PURPOSE: SELECT ALL THE ATTRIBUTES IN FROM-MODEL
AND GENERATE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ALLATT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

DDL COMMANDS TO CREATE THE ATTRIBUTES ,
THEIR KEYWORDS,
ALIAS AND DESCRIPTIONS IN THE TO-MODEL

SYNOPSIS

```
C          -- ALLATT(%MODEL_NO, %MODEL_NAME,  
           %KEYWORD_FLAG,  
           %ALIAS_FLAG, %DESC_FLAG) ;  
  
COBOL     -- CALL "ALLATT"    USING  
           MODEL-NO,  
           MODEL-NAME,  
           KEYWORD-FLAG,  
           ALIAS-FLAG,  
           DESC-FLAG.  
  
FORTRAN   -- CALL ALLATT(MODELNO, MODELNAME,  
           KEYWORDFLAG, ALIASFLAG,  
           DESCFLAG)
```

INPUT:

```
INT  *MODEL_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
INFORMATIONS TO GENERATE NDDL COMMAND STATEMENT.

ARGUMENTS:

MODEL_NO - INT *
MODEL_NAME - CHAR *
KEYWORD_FLAG - CHAR *
ALIAS_FLAG - CHAR *
DESC_FLAG - CHAR *

ROUTINES CALLED:

COSQLS - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,
COFETCH - ORACLE ROUTINE
STRNCMP
GENATT - GENERATE A CREATE ATTRIBUTE.. COMMAND
STRNCPY
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN
OBJECT TYPE AND NO
GENALI - GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT
TYPE
COEXEC - ORACLE ROUTINE
COBINDN - ORACLE ROUTINE
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

CPYMOD - CONTROLS THE PROCESSING LOGIC FOR THE COPY
MODEL COMMAND.
MRGMOD1 - COPY MODEL-1 INTO A NEW MODEL (MODEL-3)

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

DDL COMMAND PROCESSOR Module Documentation

NAME: ALLENT
PURPOSE: SELECT ALL THE ATTRIBUTES IN FROM-MODEL
AND GENERATE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ALLENT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

DDL COMMANDS TO CREATE THE ENTITY, THEIR
KEYWORDS,
ALIAS AND DESCRIPTIONS IN THE TO-MODEL

SYNOPSIS

```
C          --  ALLENT(%MODEL_NO, %MODEL_NAME,  
              %KEYWORD_FLAG,  
              %ALIAS_FLAG, %DESC_FLAG) ;  
  
COBOL     --  CALL "ALLENT"  USING  
              MODEL-NO,  
              MODEL-NAME,  
              KEYWORD-FLAG,  
              ALIAS-FLAG,  
              DESC-FLAG.  
  
FORTRAN   --  CALL ALLENT(MODELNO, MODELNAME,  
              KEYWORDFLAG, ALIASFLAG,  
              DESCFLAG)
```

INPUT:

```
INT  *MODEL_NO ;  
CHAR *MODEL_NAME ;  
INT  *KEYWORD_FLAG ;  
INT  *ALIAS_FLAG ;
```

INT *DESC_FLAG :

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO RETRIEVE RECORDS FROM
ENTITY_CLASS. ENTITY NAME THEN USE THE INFORMATIONS TO
GENERATE
DDL COMMAND STATEMENT.

ARGUMENTS:

MODEL_NO - INT *
MODEL_NAME - CHAR *
KEYWORD_FLAG - CHAR *
ALIAS_FLAG - CHAR *
DESC_FLAG - CHAR *

ROUTINES CALLED:

COSQLS - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.
COFETCH - ORACLE ROUTINE
STRNCMP
ELECLST - THIS ROUTINE CREATES A ROW IN THE UNBOUNDED
EC_LIST
GENENT - GENERATE A CREATE ENTITY..OWNED
ATTRIBUTE..KEYWORD COMMAND
STRNCPY
GENDESC - GENERATED DDL DESCRIBE COMMANDS FOR A GIVEN
OBJECT TYPE AND NO
GENALI - GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT
TYPE
COEXEC - ORACLE ROUTINE
COBINDN - ORACLE ROUTINE
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

CPYMOD - CONTROLS THE PROCESSING LOGIC FOR THE COPY
MODEL COMMAND.
MRGMOD1 - COPY MODEL-1 INTO A NEW MODEL (MODEL-3)

USED IN MAIN PROGRAM(S):

PS 080141100
1 November 1988

NDL/MAN - MAIN PROGRAM FOR THE NDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ALKEY
PURPOSE: GENERATES KEY CLASS FOR AN ENTITY
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ALKEY
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS THE SHELL ROUTINE TO GENERATE THE KEY CLASS
FOR AN ENTITY.

ARGUMENTS:

DEP-EC-NO = DSPLY [S9(9)]
DEP-EC-NAME = DSPLY [X(30)]
RC-NO = DSPLY [S9(9)]
RC-DEPKC-LIST = RECRD

INCLUDE FILES:

KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS
ECLIST - CONTAINS A LIST OF ENTITY CLASS NUMBERS
RCDEPKC - LIST OF KEYS MIGRATED VIA A RELATION
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

CHKOWN - checks each of the OWN-ec's key classes against
the RC-DEPKC LIST
GENALT1 - GENERATE AN ALTER ENTITY..ADD KEY CLASS COMMAND

PS 620141100
1 November 1985

CHKINH - checks the RC-DEPEC table and d. terminates when a
key class can be
ERRPRO

CALLED DIRECTLY BY:

ALLREL - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL
GENERATE
DEPREL - FOR EACH LEVEL OF RELATIONS IN STRUCTURE
GENERATE
MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING
MODEL_2

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ALLREL
PURPOSE: FOR EACH LEVEL OF RELATIONS IN FROM-MODEL
GENERATE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ALLREL
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          --  ALLREL(%MODEL_NO, %MODEL_NAME,  
              %KEYWORD_FLAG,  
              %DESC_FLAG,%STATUS) ;  
  
COBOL     --  CALL "ALLREL"  USING  
              MODEL-NO,  
              MODEL-NAME,  
              KEYWORD-FLAG,  
              DESC-FLAG  
              STATUS.  
  
FORTRAN   --  CALL ALLREL(MODELNO, MODELNAME,  
              KEYWORDFLAG, DESCFLAG,  
              STATUS)
```

INPUT:

INT *MODEL_NO ;
CHAR *MODEL_NAME ;
INT *KEYWORD_FLAG ;
INT *DESC_FLAG ;
INT *STATUS ;

DESCRIPTION:

ARGUMENTS:

MODEL_NO - INT *
MODEL_NAME - CHAR *
KEYWORD_FLAG - CHAR *
DESC_FLAG - CHAR *
RC_DEPKC_LIST - INT *
STATUS - INT *

ROUTINES CALLED:

BLRCKC - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL FIND
COSQLS - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE
COPFETH - ORACLE ROUTINE
GETECNM - THIS ROUTINE SEARCHES THE UNBOUNDED EC LIST DATA
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
COBINDM - ORACLE ROUTINE
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

CPYMOD - CONTROLS THE PROCESSING LOGIC FOR THE COPY
MODEL COMMAND.
MRGMOD1 - COPY MODEL-1 INTO A NEW MODEL (MODEL-3)

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDML COMMAND PROCESSOR Module Documentation

NAME: ALLVIEW
PURPOSE: CREATE AN ES-CS-MAPPING FOR AUC TO DATA
ITEM
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ALLVIEW
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

ALLVIEW CREATES THE ES TO CS MAPPING
WHEN THE USER WISHES TO MAP EACH
ATTRIBUTE USE CLASS OF
AN ENTITY CLASS TO DATA ITEMS IN THE VIEW. DATA
ITEM ARE STORED USING THE SAME NAME AS THE TAG
(ATTRIBUTE USE CLASS).

EXECUTE AN NDML SELECT TO RETRIEVE ALL
THE ATTRIBUTE USE CLASSES FOR AN ENTITY
CLASS. FOR EACH AUC RETRIEVED, GET
THE DATA-TYPE-NAME FOR THE DOMAIN, THEN
CREATE A NEW DATA_ITEM AND PROJECT_DATA_ITEM
OCCURRENCE.

ARGUMENTS:

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

INCLUDE FILES:

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

ROUTINES CALLED:

PS 620141100
1 November 1985

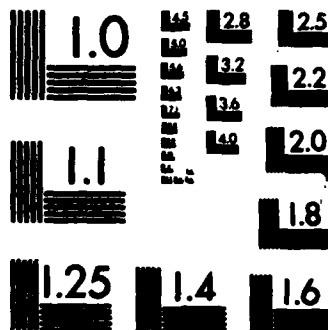
VEREDT - For a given domain number, return its standard
data type name.
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
INSDI - INSERT A RECORD INTO DATA_ITEM
INSPDI - INSERT A RECORD INTO PROJECT_DATA_ITEM
ERRPRO

CALLED DIRECTLY BY:

PIFROM - CREATE A VIEW USING A SINGLE ENTITY CLASS(
ES-CS-MAPPING)

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NDDL COMMAND PROCESSOR Module Documentation

NAME: ALTALI
PURPOSE: CONTROLS ALTER ALIAS PROCESSING (ALIAS TO
PRIM OR VICE VERS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ALTALI
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CHANGES THE NAME TYPE FROM 'ALIAS' TO
'PRIMARY' AND FROM 'PRIMARY' TO ALIAS
FOR AN ENTITY OR ATTRIBUTE.

DEPENDING ON WHETHER THE OBJECT
TYPE IS AN ENTITY OR ATTRIBUTE,
THIS VERIFIES THE EXISTENCE OF
THE OBJECT.

THE PRIM-NAME-TYPE
FOR THE OBJECT-ID WILL BECOME
'ALIAS'. THE ALIAS-NAME-TYPE
FOR THE ALIAS-ID WILL BECOME 'PRIMARY'.

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

PS 620141100
1 November 1985

VERNME - verify the existence of an entity class in a
model.
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
VERNMA - verify the existence of an attribute class in a
model.
CPPNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.
UPDECAL - UPDATE MODEL_CLASS SET EC_NAME_TYPE = : 1
UPDACAL - UPDATE MODEL_CLASS SET AC_NAME_TYPE = : 1

CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: ALTATT
PURPOSE: THE ALTER ATTRIBUTE COMMAND PROCESSOR
ALTERS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ALTATT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

USER SPECIFIED ATTRIBUTES IN THE CDM.

SYNOPSIS

C	--	ALTATT(%MODEL_NO, %STATUS)	
COBOL	--	CALL "ALTATT"	USING MODEL-NO STATUS.
FORTTRAN	--	CALL ALTATT (MODELNO, STATUS)	

INPUT:

INT *MODEL_NO

OUTPUT:

INT *STATUS

DESCRIPTION

ALTATT EXTRACTS THE ATTRIBUTE CLASS TO BE ALTERED FROM
THE LIST,
VERIFIES THAT IT EXISTS, AND MAKES CALLS TO CHANGE THE
DOMAIN,

DROP KEYWORD AND ADD KEYWORD.

ARGUMENTS:

MODEL NO - INT *
STATUS - INT *

INCLUDE FILES:

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

ROUTINES CALLED:

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A
MODEL
CHGDOM - CHANGE THE DOMAIN FOR AN ALTERED ATTRIBUTE
SPRINTF
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD
KEYWORD"(OPTIONAL)
DRPKW - DROP A KEYWORD ASSOCIATION FROM EITHER AN
ATTRIBUTE, ENTITY OR REL

CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: ALTCARD
PURPOSE: PROCESS CARDINALITY FO USER SPECIFIED
RELATION
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ALTCARD
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

ALTCARD OBTAINS THE USER-SPECIFIED CARDINALTY
(IF ANY) FOR THE RELATION FROM THE LISTS.
VERIFIES IT AND TRANSFERS IT TO NUMERIC
VARIABLES WHICH ARE RETURNED TO THE CALLING
ROUTINE.

ARGUMENTS:

NO-IND-ENT = DSPLY [S9(9)]
MIN-NO-DEP-ENT = DSPLY [S9(9)]
MAX-NO-DEP-ENT = DSPLY [S9(9)]
RTM-STATUS = 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
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

PS 620141100
1 November 1985

CALLED DIRECTLY BY:

**ALTREL - CONTROLS PROCESSING LOGIC FOR THE ALTER
RELATION COMMAND**

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

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

DESCRIPTION:

THE ALTER DOMAIN COMMAND OBTAINS A DOMAIN NAME,
THEN CALLS THE CLAUSE PROCESSORS TO DROP DATA
TYPES ADD DATA TYPES AND ALTER DATA TYPES.

FIRST OBTAIN DOMAIN NAME AND VERIFY THAT
IT DOES EXIST. USING THE DOMAIN NUMBER
DROP, ADD AND ALTER DATA TYPE.

INCLUDE FILES:

LISTNOS - VALID LIST NUMBERS

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
PROCDT - PROGRAM NAME

CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

PS 620141100
1 November 1985

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

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

DESCRIPTION:

THIS CLAUSE WILL ALTER THE DATA TYPE GIVEN THE
DOMAIN NUMBER

GIVEN THE DOMAIN NUMBER, EXTRACT THE
DATA TYPE NAME, ID, TYPE-SIZE FOR A DATA
TYPE. VERIFY IF IT IS STANDARD OR USER
DEFINED TYPE AND THAT THE TYPE EXISTS.
IF IT DOES, UPDATE THE DATA TYPE, AND
FETCH THE NEXT RECORD TO BE UPDATED.

ARGUMENTS:

DOM-NO, DATA-TYPE-NAME, TYPE-ID =
MAX-SIZE, NO-OF-DEC, STD-USER =

ROUTINES CALLED:

VERDTD - VERIFY THE EXISTENCE OF A DATA TYPE IN A GIVEN
DOMAIN
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
VERSDT - For a given domain number, return its standard
data type name.
UPDIND - UPDATE USER_DEF_DATA_TYPE SET DATA_TYPE_IND = :
1
UPDTDT - UPDATE USER_DEF_DATA_TYPE SET TYPE_ID = : 1

CALLED DIRECTLY BY:

PS 620141100
1 November 1985

PROCDT - PROGRAM NAME

USED IN MAIN PROGRAM(S):

MDDL/MAIN - MAIN PROGRAM FOR THE MDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: ALTEXT
PURPOSE: CONTROL PROCESSING FOR ALTER ENTITY CLASS
COMMAND.
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ALTEXT
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CONTROLS THE PROCESSING LOGIC FOR THE ALTER
ENTITY CLASS COMMAND. THE COMMAND CONSISTS OF
AN ENTITY CLASS NAME FOLLOWED BY ONE OR MORE
OPTIONAL CLAUSES. THE CLAUSES ARE ADD KEY CLASS,
ADD OWNED ATTRIBUTE LIST, ADD KEYWORD, DROP KEY
CLASS, DROP OWNED ATTRIBUTE, AND/OR DROP KEYWORD.

AN ERROR VERIFYING ENTITY CLASS CAUSES
PROCESSING TO STOP. OTHERWISE, EACH
CLAUSE IS PROCESSED, REGARDLESS OF ERRORS
WHICH MAY HAVE OCCURRED WHILE PROCESSING
A PREVIOUS CLAUSE.

ARGUMENTS:

MODEL-NO = DSPLY [S9(9)]

INCLUDE FILES:

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
DRPAC - DELETE OWNED ATTRIBUTES ASSOCIATED WITH ENTITY
DRPKC - CONTROLS THE PROCESSING LOGIC FOR THE "DROP KEY
CLASS".
ADDATT - ASSOCIATES EXISTING ATT WITH ENTITY IN GREAT
ENTITY COMMAND
ADDKC - CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR
CREATE/ALTER ENTITY
DRPKW - DROP A KEYWORD ASSOCIATION FROM EITHER AN
ATTRIBUTE, ENTITY OR REL
ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD
KEYWORD"(OPTIONAL)
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER

CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

C -- ALTMAP
COBOL -- CALL "ALTMAP".

FORTRAN -- CALL ALTMAP

INPUT:

OUTPUT:

DESCRIPTION

ALTMAP IS THE COMMAND PROCESSOR FOR THE ALTER MAP
COMMAND. IT
VALIDATES CERTAIN COMMON PARAMETERS AND THEN CALLS
SUBORDINATE
ROUTINES DEPENDING ON WHETHER THE ALTER ADD, ALTER
DROP, OR
ALTER ALTER FUNCTIONS ARE TO BE PERFORMED.

INCLUDE FILES:

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

ROUTINES CALLED:

CPPONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A
MODEL

SPRINTF
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.
STRNCMP
ALTSMAP - ALTER A SINGLE MAP
DRPSMAP - DROP A SINGLE MAPPING
ADDMAP - ADD A CS-IS MAPPING
STRNCPY
TOLOWER
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: ALTHOD
PURPOSE: CONTROLS PROCESSING FOR ALTER MODEL
COMMANDS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ALTHOD
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THE ALTER MODEL COMMAND PROCESSOR WILL CONTROL
THE PROCESSING LOGIC OF ALTERING A MODEL WITH-
IN 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

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
UPDMOD - UPDATE MODEL CLASS SET MODEL_STATUS = : 1
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A

CALLED DIRECTLY BY:

PS 620141100
1 November 1985

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

WDDL COMMAND PROCESSOR Module Documentation

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

DESCRIPTION:

CONTROLS THE PROCESSING LOGIC FOR THE ALTER
RELATION COMMAND.

ARGUMENTS:

MODEL-NO = DSPLY [S9(9)]

INCLUDE FILES:

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
ALTCARD - PROCESS CARDINALITY FO USER SPECIFIED RELATION
UPDTRC - UPDATE RELATION_CLASS SET NO_IND_ENT = : 1,

PS 620141100
1 November 1985

ADDMIG - PROCESS THE ADD MIGRATES...SET..CLAUSE
ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD
KEYWORD"(OPTIONAL)
DRPMIG - CONTROLS THE PROCESSING LOGIC FOR THE "DROP
MIGRATES" CLAUSE.
DRPKW - DROP A KEYWORD ASSOCIATION FROM EITHER AN
ATTRIBUTE, ENTITY OR REL
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER

CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

DDDL COMMAND PROCESSOR Module Documentation

NAME: ALTSMAP
PURPOSE: ALTER A SINGLE MAP
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ALTSMAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

```
      C          --  ALTSMAP (%TAG_NO, %AC_NO)

      COBOL      --  CALL  "ALTSMAP"      USING
                                     TAG-NO
                                     AC-NO.

      FORTRAN   --  CALL  ALTSMAP (TAGNO, ACNO)
```

INPUT:

```
      INT *TAG_NO
      INT *AC_NO
```

OUTPUT:

DESCRIPTION

ALTSMAP MODIFIES, FOR AUC TO DATA FIELD MAPS, THE
PRIMARY/SECONDARY
INDICATOR FROM SECONDARY TO PRIMARY AND THE DATA TYPE
NAME. FOR AUC
TO SET MAPS, ALTSMAP MODIFIES THE AUC VALUE.

ARGUMENTS:

TAG_NO - INT *
AC_NO - INT *

INCLUDE FILES:

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

ROUTINES CALLED:

CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.
VERPDF - verify the existence of a project_data_field
occurrence.
CPFCOR - ACCESS A TOKEN IN A CORRESPONDING NAMED LIST
FINDDOM - RETRIEVE A DOMAIN NUMBER FOR A GIVEN DOMAIN NAME
VERDTD - VERIFY THE EXISTENCE OF A DATA TYPE IN A GIVEN
DOMAIN
SPRINTF
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
COSQL3 - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.
COEXEC - ORACLE ROUTINE
COBINDN - ORACLE ROUTINE
FNDASH - DETERMINES IF AN AUC TO SET TYPE MAPPING EXISTS
CHKAUCV - CHECK EXISTENCE OF AUC TO SET MAPPING
TOLOWER
STRNCMP
VERDB - VERIFY THE EXISTENCE OF A DATA BASE IN THE CDM

CALLED DIRECTLY BY:

ALTMAP - ALTER MAP COMMAND PROCESSOR

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

WDDL COMMAND PROCESSOR Module Documentation

NAME: ATTKV
PURPOSE: CONTROLS PROCESSING TO POPULATE KEYWORD
TABLE FOR AUC KEYWO
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ATTKV
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE CONTROLS THE PROCESSING LOGIC TO POPULATE A
KEYWORD TABLE WITH ATTRIBUTE CLASS KEYWORDS OF ONE MODEL,
THEN SEARCHES THIS TABLE FOR MATCHES FROM THE SECOND MODEL.

ARGUMENTS:

MOD-NO1 = DSPLY [S9(9)]
MOD-NO2 = DSPLY [S9(9)]
MOD-NAME1 = DSPLY [X(30)]
MOD-NAME2 = DSPLY [X(30)]

INCLUDE FILES:

KWDTBL - KEYWORD TABLE
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

RETACKW - POPULATES KEYWORD TABLE FOR ATTRIBUTE CLASS
KEYWORDS
RACKW2 - COMPARES ATTRIBUTE CLASS KEYWORDS
ERRPRO

PS 620141100
1 November 1985

CALLED DIRECTLY BY:

**CMPMOD - CONTROLS THE PROCESSING LOGIC TO COMPARE TWO
MODELS.**

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: BLDATT
PURPOSE: CREATE ATT CLASS AND ATT NAME FOR A
MODEL(CREATE/COPY ATT)
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: BLDATT
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

BLDATT CREATES AN ATTRIBUTE CLASS AND ATTRIBUTE
CLASS NAME OCCURENCE FOR A MODEL. BLDATT IS
USED BY BOTH THE CREATE ATTRIBUTE AND THE COPY
ATTRIBUTE COMMAND PROCESSORS

ARGUMENTS:

AC-NAME = DSPLY [X(30)]
MODEL-NO = DSPLY [S9(9)]
DOM-NO = DSPLY [S9(9)]
NEW-AC-NO = DSPLY [S9(9)]

INCLUDE FILES:

UNIQENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED:

VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A
MODEL
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
GETNNUM -
INSAC - INSERT A RECORD INTO ATTRIBUTE_CLASS

PS 620141100
1 November 1985

CALLED DIRECTLY BY:

CRTATT - CONTROLS THE PROCESSING LOGIC FOR CREATING AN
ATTRIBUTE.
ICOPATT - INTERACTIVE COPY ATTRIBUTE, WITH
KEYWORDS, ALIAS, DESCRS.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: BLDATT1
PURPOSE: CREATES ATT CLASS AND ATT NAME FOR A
MODEL(CREATE/COPY ATT)
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: BLDATT1
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

BLDATT CREATES AN ATTRIBUTE CLASS AND ATTRIBUTE
CLASS NAME OCCURENCE FOR A MODEL. BLDATT1 IS
USED BY BOTH THE CREATE ATTRIBUTE AND THE COPY
ATTRIBUTE COMMAND PROCESSORS

ARGUMENTS:

AC-NAME = DSPLY [X(30)]
MODEL-NO = DSPLY [S9(9)]
OLD-AC-NO = DSPLY [S9(9)]
DOM-NO = DSPLY [S9(9)]
NEW-AC-NO = DSPLY [S9(9)]
KW-FLAG = DSPLY [9]
ALIAS-FLAG = DSPLY [9]
DESC-FLAG = DSPLY [9]

INCLUDE FILES:

UNIQUENO - UNIQUE NUMBER ASSIGNMENTS FOR CDM OBJECTS

ROUTINES CALLED:

VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A
MODEL
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A

PS 620141100
1 November 1985

GETNUM -
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
INSAC - INSERT A RECORD INTO ATTRIBUTE_CLASS
WRTANAM - RETRIEVE AND COPY ALL THE NAMES (PRIMARY AND
ALIAS) OF AN ATTRIBU
WRTACKV - RETREIVE AND WRITE ALL THE KEYWORDS ASSOCIATD
WITH AN ATTRIBUTE C
INSACNM - INSERT A RECORD INTO ATTRIBUTE_NAME
WRTDESC - SELECT A RECORD FROM DESC_TEXT ENTITY

CALLED DIRECTLY BY:

COPYAC - CREATE AN ATTRIBUTE, ASSOCIATE WITH ENTITY, ADD
KEY CLASSES

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: BLECLST
PURPOSE: THIS ROUTINE CREATES A ROW IN THE
UNBOUNDED_EC_LIST
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: BLECLST
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

DATA STRUCTURE.

SYNOPSIS

C -- BLECLST(SEC_NO, SEC_NAME, STATUS)
COBOL -- CALL "BLECLST" USING
EC-NO,
EC-NAME,
STATUS.
FORTRAN -- CALL BLECLST(ECNO, ECNAME, STATUS)

INPUT:

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

OUTPUT:

DESCRIPTION:

ARGUMENTS:

EC_NO = INT *
EC_NAME = CHAR *

INCLUDE FILES:

KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

ROUTINES CALLED:

ADDECNM - ADD THE EC_NAME AND EC_NO INTO KEYLIST

CALLED DIRECTLY BY:

ALLENM - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND
GENERATE
DEPENT - SELECT ALL THE DEPENDANT ENTITY CLASSES

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: BLKCL1
PURPOSE: STORE ALL KEY CLASS INFO FOR AN ENTITY IN
A LIST
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: BLKCL1
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE CREATES A KEY-CLASS-LIST TABLE CONTAINING THE
KEY CLASS NUMBERS. KEY CLASS NAMES, KEY CLASS MEMBER TAG
NUMBER AND TAG NAMES FOR A GIVEN ENTITY.

ARGUMENTS:

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

INCLUDE FILES:

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

ROUTINES CALLED:

AKCROW - THIS ROUTINE ADDS A ROW TO THE UNBOUNDED
KEY_CLASS_LIST
ERRPRO

CALLED DIRECTLY BY:

PS 620141100
1 November 1985

CERELS - GENERATES NDDL COMMANDS ON A FILE FOR ALL
ENTITIES IN A RELATION.
GESTRUC - GENERATES NDDL COMMANDS ON A FILE FOR ALL
ENTITIES FOR THE STRUCT
CMBENT - CONROLS THE PROCESSING LOGIC FOR THE COMBINE
ENTITY COMMAND.
DEPFROM - GENERATE CREATE RELATION, DESCRIBE COMMANDS IN
THE TO-MODEL

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: BLKCLST
PURPOSE: SELECT AND STORE KEY CLASS INFO FOR A
GIVEN ENTITY
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: BLKCLST
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SELECTS THE KEY CLASSES AND KEY CLASS MEMBERS FOR EACH ENTITY
IN THE FROM-MODEL AND CALLS A ROUTINE TO BUILD THE UNBOUNDED
KEY-CLASS-LIST STRUCTURE.

ARGUMENTS:

FROM-MOD-NO = DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

AKCROW - THIS ROUTINE ADDS A ROW TO THE UNBOUNDED
KEY_CLASS_LIST
ERRPRO

CALLED DIRECTLY BY:

PS 620141100
1 November 1985

CPYMOD - CONTROLS THE PROCESSING LOGIC FOR THE COPY
MODEL COMMAND.
MRGMOD - MERGE TWO IDEF MODELS INTO ONE
MRGMOD1 - COPY MODEL-1 INTO A NEW MODEL (MODEL-3)

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: BLOOPCK
PURPOSE: CHECK FOR LOOPS FROM THE ENTITY GIVEN UP
THE HIERARCHY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: BLOOPCK
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C --

BLOOPCK(EC_NO)

COBOL

-- CALL "BLOOPCK"

USING

EC-NO.

FORTRAN --

CALL BLOOPCK (ECNO)

INPUT:

INT EC_NO THE ENTITY CLASS FOR WHICH AN UPWARD
HIERARCHY WILL
BE SEARCHED

OUTPUT:

WARNING MESSAGES WILL BE ISSUED

STATUS

SET TO -1 IF A LOOP IS FOUND
LEFT ALONE OTHERWISE

* DESCRIPTION

USE A SQL TREE SEARCH FOR ALL RELATIONS ABOVE THIS
ENTITY, IF A LOOP

IN THE USER DATA IS ENCOUNTERED, REPORT AN ERROR TO THE
USER.

ARGUMENTS:

PS 620141100
1 November 1985

EC_NO - INT
STATUS - INT *

ROUTINES CALLED:

COSQLS - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE
COFETCH - ORACLE ROUTINE
SPRINTF
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A
COEXEC - ORACLE ROUTINE
CODFINN - ORACLE ROUTINE
COBINDM - ORACLE ROUTINE

CALLED DIRECTLY BY:

CHKLOOP - CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: BLRCKC
PURPOSE: FOR EACH LEVEL OF RELATIONS IN FROM-MODEL
FIND
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: BLRCKC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THE KEY CLASSES FOR THE DEPENDENT ENTITY
WHICH CONTAIN ATTRIBUTES INHERITED VIA
THE RELATION AND CALL COBOL ROUTINES
TO POPULATE THE RC_DEPKC_LIST.

SYNOPSIS

C -- BLRCKC(%MODEL_NO, %MODEL_NAME,
RC_DEPKC_LIST)

COBOL -- CALL "BLRCKC" USING
MODEL-NO,
MODEL-NAME.

FORTRAN -- CALL BLRCKC(MODELNO, MODELNAME, RCDEPKC)

INPUT:

INT *MODEL_NO ;
CHAR *MODEL_NAME ;
INT *RC_DEPKC_LIST ;

DESCRIPTION:

ARGUMENTS:

MODEL_NO = INT *

PS 620141100
1 November 1985

MODEL_NAME = CHAR *
RC_DEPKC_LIST = INT *
RET_STATUS = INT *

ROUTINES CALLED:

INITRDL - INITIALISE THE TABLE WHICH STORES A MODELS
RELATIONS
COSQL3 - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,
COFETCH - ORACLE ROUTINE
ADDRCEC - POPULATES THE RC-DEPKC TABLE FOR ALL RELATIONS
IN THE MODEL
COEXEC - ORACLE ROUTINE
COBINDM - ORACLE ROUTINE
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

ALLREL - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL
GENERATE
MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING
MODEL_2

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: BLRCKC1
PURPOSE: FOR EACH LEVEL OF RELATIONS IN STRUCTURE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: BLRCKC1
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

FIND THE KEY CLASSES FOR FIND THE DEPENDENT
ENTITY
WHICH CONTAIN ATTRIBUTES INHERITED VIA
THE RELATION AND CALL COBOL ROUTINES
TO POPULATE THE RC_DEPKC_LIST.

SYNOPSIS

C -- BLRCKC1(EC_NO,MODEL_NO,MODEL_NAME,
RC_DEPKC_LIST)

COBOL -- CALL "BLRCKC1" USING
EC-NO
MODEL-NO,
MODEL-NAME.
RC-DEPKC-LIST.

FORTRAN -- CALL BLRCKC1(EC-NO,MODELNO,
MODELNAME,RCDEPKC)

INPUT:

INT *MODEL_NO ;
CHAR *MODEL_NAME ;
INT *RC_DEPKC_LIST ;

DESCRIPTION:

ARGUMENTS:

EC_NO - INT *
MODEL_NO - INT *
MODEL_NAME - CHAR *
RC_DEPKC_LIST - INT *

ROUTINES CALLED:

INITRDL - INITIALISE THE TABLE WHICH STORES A MODELS
RELATIONS
COSQL3 - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE,
COFETCH - ORACLE ROUTINE
ADDRCEC - POPULATES THE RC-DEPKC TABLE FOR ALL RELATIONS
IN THE MODEL
COEXEC - ORACLE ROUTINE
COBINDN - ORACLE ROUTINE
CODFINN - ORACLE ROUTINE

CALLED DIRECTLY BY:

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: BLSECRC
PURPOSE: CONTROLS PROCESS TO BUILD SEC-RC
COMPONENTS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: BLSECRC
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE CONTROLS THE PROCESSING LOGIC TO
BUILD THE SEC-RC- COMPONENTS. IT IS CALLED FROM
REVIEW, WHICH IS TRYING TO CREATE A VIEW.

FOR EVERY ENTRY IN THERC LIST, CREATE
AN OCCURRENCE OF SEC-RC ENTRY.

ARGUMENTS:

VIEW-RELATION-CLASS-LIST = RECRD
VIEW-NO = DSPLY [S9(9)]
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

VWRC - LIST OF RELATION CLASSES INHERENT TO A VIEW

ROUTINES CALLED:

INSSECR - INSERT A RECORD INTO SEC_RC COMPONENT
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

CALLED DIRECTLY BY:

REVIEW - POPULATE VIEW DEFINITION AND ES-CS-MAPPING

PS 620141100
1 November 1985

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: BLVWLST
PURPOSE: CREATE BUILD VIEW LISTS FOR THE CREATE VIEW COMMAND
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: BLVWLST
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

BLVWLST EXTRACTS THE INPUT FOR THE CREATE VIEW COMMAND FROM THE PARSER LISTS AND STORES IT IN THE VIEW-FROM-LIST, THE VIEW-DATA-ITEM-LIST, THE VIEW-RETRIEVE-LIST AND THE VIEW-RELATION-CLASS-LIST. TABLE OVERFLOW CHECKING PERFORMED.

ARGUMENTS:

VIEW-FROM-LIST = RECRD
VIEW-DATA-ITEM-LIST = RECRD
VIEW-RETRIEVE-LIST = RECRD
VIEW-RELATION-CLASS-LIST = RECRD
RTN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

LISTNOS - VALID LIST NUMBERS
VWFROM - LIST OF ENTITIES SPECIFIED IN A VIEW
VWDI - LIST OF DATA ITEMS IN A VIEW
VWRETR - LIST OF ENTITIES AND TAGS SPECIFIED IN A VIEW
VWRC - LIST OF RELATION CLASSES INHERENT TO A VIEW

ROUTINES CALLED:

PS 620141100
1 November 1985

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.
CPFCOR - ACCESS A TOKEN IN A CORRESPONDING NAMED LIST

CALLED DIRECTLY BY:

CRTVIEW - CONTROLS THE PROCESSING LOGIC FOR THE CREATE
VIEW COMMAND.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: BRANCHR.
PURPOSE: PERFORMS MULTI-WAY CALL TO THE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: BRANCHR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

INDIVIDUAL COMMAND PROCESSOR.

SYNOPSIS

C -- BRANCHR(COMMAND_NO);
COBOL -- CALL "BRANCHR" USING

COMMAND_NO.

FORTRAN --
CALL BRANCHR(CMDNO)

INPUT:

COMMAND_NO AN INTEGER VARIABLE, A VALUE INDICATING
WHICH COMMAND IS TO BE EXECUTED,
FOR VALUE ASSIGNMENTS SEE THE FILE
"CMDS.H"

OUTPUT:

DESCRIPTION

SIMPLY PERFORM A SWITCH ON COMMAND NO, AT EAC
CASE CALL THE APPROPRIATE COMMAND PROCESSOR.

PLEASE KEEP THE CASES IN ALPHABETICAL ORDER!

ARGUMENTS:

COMMAND_NO - INT

INCLUDE FILES:

CMDS - COMMAND NO. FOR EACH NDDL/NDML COMMAND
NDDL - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

ERRPRO
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ALтали - CONTROLS ALTER ALIAS PROCESSING (ALIAS TO PRIM
OR VICE VERSA)
ALTATT - THE ALTER ATTRIBUTE COMMAND PROCESSOR ALTERS
PRINTF
ALтDOM - PROGRAM NAME
ALTENT - CONTROL PROCESSING FOR ALTER ENTITY CLASS
COMMAND.
ALтMAP - ALTER MAP COMMAND PROCESSOR
ALтMOD - CONTROLS PROCESSING FOR ALTER MODEL COMMANDS
ALтREL - CONTROLS PROCESSING LOGIC FOR THE ALTER
RELATION COMMAND
CHKMOD - DETERMINES WHETHER CERTAIN RULES ARE FULFILLED,
CMBENT - CONROLS THE PROCESSING LOGIC FOR THE COMBINE
ENTITY COMMAND.
CMPMOD - CONTROLS THE PROCESSING LOGIC TO COMPARE TWO
MODELS.
COPATT - CONTROLS THE PROCESSING LOGIC FOR COPYING AN
ATTRIBUTE.
CPYDES - COMMAND PROCESSOR FOR COPY DESCRIPTION
COPENT - CONTROLS THE PROCESSING LOGIC FOR THE COPY
ENTITY COMMAND.
CPYMOD - CONTROLS THE PROCESSING LOGIC FOR THE COPY
MODEL COMMAND.
CRTATT - CONTROLS THE PROCESSING LOGIC FOR CREATING AN
ATTRIBUTE.
CRTALI - CREATES ALIASES FOR AN ENTITY OR ATTRIBUTE.
CRTDOM - PROGRAM NAME CRTDOM
CRTENT - CONTROL THE PROCESSING LOGIC FOR CREATING A NEW
ENTITY CLASS.
CRTMAP - CREATE MAP COMMAND PROCESSOR

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
DEFREC - CONTROLS THE PROCESSING LOGIC FOR DEFINING A
RECORD FOR THE SYSTE
DEFSET - CONTROLS THE PROCESSING LOGIC FOR THE DEFINE
SET COMMAND.
DRPDB - CONTROLS THE PROCESSING LOGIC FOR DELETING THE
DATA BASE.
DRPFLD - CONTROLS THE PROCESSING LOGIC FOR DROPPING A
DATA FIELD.
DRPPREC - CONTROLS THE PROCESSING LOGIC FOR THE DROP
RECORD COMMAND.
DRPSET - CONTROLS THE PROCESSING LOGIC FOR DELETING A
SET FROM THE DATABAS
DESCRB - COMMAND PROCESSOR FOR THE NDDL DESCRIBE COMMAND
DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED
ATTRIBUTE CLASSES FROM TH
DRPDOM - PROGRAM NAME
DRPALI - DROP THE ALIAS FOR AN ENTITY OR ATTRIBUTE.
DRPVIEW - DROP THE VIEW.
DRPKWC - OBTAIN THE USED IDENTIFIED KEYWORD, THEN DROP
THEIR ASSOCIATIONS.
DRPMAP - COMMAND PROCESSOR FOR THE DROP MAP COMMAND
DRPENT - CONTROL THE PROCESSING LOGIC FOR DELETING
ENTITIES.
DRPMOD - CONTROLS THE PROCESSING LOGIC TO DROP A MODEL.
DRPREL - CONTROLS THE PROCESSING LOGIC FOR THE "DROP
RELATION" COMMAND.
MRGMOD - MERGE TWO IDEF MODELS INTO ONE
RENAME - UPDATE EXISTING OBJECT NAME WITH NEW OBJECT NAME
HALT - HALT WITH 'COMMIT' OR 'ROLLBACK'.

CALLED DIRECTLY BY:

PRCCMD - THIS ROUTINE IS CALLED TO HANDLE ALL

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CDP4A
PURPOSE: VERIFY SURROGATE ENTITY CLASS STRUCTURE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CDP4A
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

- THIS ROUTINE WILL PERFORM THE NECESSARY
ALGORITHM TO VERIFY A LEGAL SURROGATE ENTITY
CLASS STRUCTURE:
1. ALL ENTITY CLASSES CONNECTED BY 1-1 SETS MUST
BE A REGULAR HIERARCHY
2. EACH GROUPING OF 1-1 STRUCTURES CAN BE SUBSTITUTED
AS A SINGLE NODE
3. THE RESULTING STRUCTURE OF ALL 1-MANY RELATIONSHIPS
MUST FORM A CONFLUENT HIERARCHY

ALL INPUT TO THIS ROUTINE WILL BE SET UP IN THE
SEC-DECOMP-LIST WHICH CONTAINS ALL ENTITY CLASSES THAT
ARE RELATED AND THE TYPE OF RELATIONSHIP.

-

USING A POPULATED SEC-DECOMP-LIST, THIS ROUTINE
WILL VALIDATE A VALID SURROGATE ENTITY CLASS, I.E.
A REGULAR-CONFLUENT HIERARCHY

ARGUMENTS:

SEC-DECOMP-LIST = RECRD
RET-STATUS = DSPLY [X(5)]

INCLUDE FILES:

PS 620141100
1 November 1985

SDLIST - SEC-DECOMPOSITION-LIST
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

ERRPRO

CALLED DIRECTLY BY:

VERRELS - VERIFY THE RELATION CLASS STRUCTURE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CERELS
PURPOSE: GENERATES NDDL COMMANDS ON A FILE FOR ALL ENTITIES IN A REL
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CERELS
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE GENERATES NDDL COMMANDS ON A FILE FOR ALL ENTITIES IN A RELATION.

ARGUMENTS:

FROM-MODEL-NO = DSPLY [S9(9)]
CUR-MODEL-NO = DSPLY [S9(9)]
CUR-MODEL-NAME = DSPLY [X(30)]
FROM-EC-NO = DSPLY [S9(9)]
TO-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)]

INCLUDE FILES:

RELTLB - LIST OF RELATION CLASSES IN A MODEL
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

OPNFIL
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS
CHECKS/DELIMITER
GEMENT1 - GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND
DESCRS.
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
BLKCL1 - STORE ALL KEY CLASS INFO FOR AN ENTITY IN A LIST
DEPFROM - GENERATE CREATE RELATION, DESCRIBE COMMANDS IN
THE TO-MODEL
GENALTE - GENERATE AN ALTER ENTITY..ADD KEY CLASS COMMAND
INDFROM - RETRIEVES RELATIONS, DETERMINES IND EC AND
GENERATES NDDL
CLSFIL - THIS ROUTINE CLOSSES AN OUTPUT FILE. THE FILE
WILL
ERRPRO

CALLED DIRECTLY BY:

FCOPENT - DETERMINE IF COPY ENTITY WITH STRUCTURE OR
RELATION

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: CESTRUC
PURPOSE: GENERATES NDDL COMMANDS ON A FILE FOR ALL ENTITIES FOR THE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CESTRUC
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE GENERATES NDDL COMMANDS ON A FILE FOR ALL ENTITIES FOR THE STRUCTURE.

ARGUMENTS:

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

INCLUDE FILES:

RCDEPKC - LIST OF KEYS MIGRATED VIA A RELATION
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

PS 620141100
1 November 1985

OPNFIL
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS
CHECKS/DELIMITER
GENENT1 - GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND
DESCRS.
BLKCL1 - STORE ALL KEY CLASS INFO FOR AN ENTITY IN A LIST
GENALTE - GENERATE AN ALTER ENTITY..ADD KEY CLASS COMMAND
DEPATT - SELECT ALL THE ATTRIBUTES IN THE
DEPENT - SELECT ALL THE DEPENDANT ENTITY CLASSES
DPKCLST - CREATE AN KEY CLASS_LIST TABLE CONTAINING ALL
THE ENTITY
DEPREL - FOR EACH LEVEL OF RELATIONS IN STRUCTURE
GENERATE
GLSFIL - THIS ROUTINE CLOSES AN OUTPUT FILE. THE FILE
WILL
ERRPRO

CALLED DIRECTLY BY:

FCOPENT - DETERMINE IF COPY ENTITY WITH STRUCTURE OR
RELATION

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHGDOM
PURPOSE: CHANGE THE DOMAIN FOR AN ALTERED ATTRIBUTE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: CHGDOM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CLASS.

SYNOPSIS

C	--	CHGDOM(%AC_NO, %STATUS)	
COBOL	--	CALL "CHGDOM"	USING AC-NO STATUS.
FORTRAN	--	CALL CHGDOM (ACNO, STATUS)	

INPUT:

INT *AC_NO

OUTPUT:

INT *STATUS

DESCRIPTION

CHGDOM ALLOWS AN ATTRIBUTE CLASS TO CHANGE FROM ONE DOMAIN TO ANOTHER. THE VALUE OF THE FUNCTION IS ZERO IF SUCCESSFUL AND -1 OTHERWISE.

ARGUMENTS:

PS 620141100
1 November 1985

AC_NO - INT *
STATUS - INT *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
LISTID - PROVIDES LIST OF PARSED OBJECTS
UNI QENO - 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
SPRINTF
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
UPDAC - UPDATE ATTRIBUTE_CLASS SET DOMAIN_NO = : 1

CALLED DIRECTLY BY:

ALTATT - THE ALTER ATTRIBUTE COMMAND PROCESSOR ALTERS

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHGLOBL
PURPOSE: CHANGES THE GLOBAL DB PARAMETERS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: CHGLOBL
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C CHGLOBL(DB_ID, DB_NAME, HOST_NAME, DBMS_NAME)

COBOL -- CALL "CHGLOBL" USING DB_ID,
DB_NAME,
HOST_NAME,
DBMS_NAME.

FORTRAN -- CALL CHGLOBL
(DB_ID, DB_NAME, HOST_NAME, DBMS_NAME)

INPUT:

OUTPUT:

DB_ID,
DB_NAME,
HOST_NAME,
DBMS_NAME.

DESCRIPTION

THIS ROUTINE WILL CHANGE THE FOLLOWING 4 GLOBAL VARIABLE
1. CUR_DBID (CURRENT DATABASE ID)

PS 620141100
1 November 1985

2. CUR_DBNAME (CURRENT DATABASE NAME)
3. CUR_HOST (CURRENT HOST NAME)
4. CUR_DBMS (CURRENT DATABASE MANAGEMENT)

ARGUMENTS:

DB_ID = INT *
DB_NAME = CHAR *
HOST_NAME = CHAR *
DBMS_NAME = CHAR *

CALLED DIRECTLY BY:

DEFDB - CONTROLS THE PROCESSING LOGIC FOR DEFINING A
DATABASE TO THE SYST
GETDBST - RETURN INFORMATION ABOUT THE CURRENT SESSIONS'
DATABASE
GETRDH - RETURN WITH CURRENT SESSIONS' DATA BASE INFO

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHKATT
PURPOSE: CHECK IF ATTRIBUTES HAVE BEEN CREATED
ACCORDING TO STANDARD
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CHKATT
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE CHECKS TO SEE IF THE FOLLOWING RULES ARE
FULFILLED:

- 1) EACH ENTITY MUST HAVE AT LEAST ONE ATTRIBUTE USE CLASS
- 2) EACH ATTRIBUTE CLASS MUST HAVE A DOMAIN AND THAT DOMAIN
MUST HAVE A STANDARD DATA TYPE.

ARGUMENTS:

EC-NO = DSPLY [S9(9)]
EC-NAME = DSPLY [X(30)]
NDDL-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ERRPRO

PS 620141100
1 November 1985

CALLED DIRECTLY BY:

GETECS - USING ENTITY CLASS VERIFIES CHECK MODEL RULES

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHKAUCV
PURPOSE: CHECK EXISTENCE OF AUC TO SET MAPPING
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CHKAUCV
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CHKAUCV SEARCHES ALL AUC TO SET MAPPINGS FOR A PARTICULAR AUC TO DETERMINE WHETHER AN AUC VALUE HAS BEEN USED MORE THAN ONCE FOR A SINGLE AUC. IF SO, THE FOUND FLAG IS SET TO ONE, OTHERWISE ZERO.

AN NDML SELECT STATEMENT IS EXECUTED TO OBTAIN ALL OCCURENCES OF AUC ST MAPPING WHOSE DB_ID, TAG_NO, AND AUC_VALUE MATCH. IF ANY ARE FOUND, FOUND IS SET TO ONE, OTHERWISE ZERO.

ARGUMENTS:

DB-NO = DSPLY [S9(9)]
TAG-NO = DSPLY [S9(9)]
AUC-VALUE = DSPLY [X(30)]
FOUND = DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

ERRPRO

PS 620141100
1 November 1985

CALLED DIRECTLY BY:

ALTSMAP - ALTER A SINGLE MAP

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHKCARD
PURPOSE: OBTAINS THE USER-SPECIFIED CARDINALTY (IF ANY) FOR THE RELATION FROM THE LISTS,
VERIFIES IT AND TRANSFERS IT TO NUMERIC VARIABLES WHICH ARE RETURNED TO THE CALLING ROUTINE. IF A CARDINALTY IS OMITTED BY THE USER, IT IS ASSIGNED A DEFAULT VALUE.
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CHKCARD
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CHKCARD OBTAINS THE USER-SPECIFIED CARDINALTY (IF ANY) FOR THE RELATION FROM THE LISTS, VERIFIES IT AND TRANSFERS IT TO NUMERIC VARIABLES WHICH ARE RETURNED TO THE CALLING ROUTINE. IF A CARDINALTY IS OMITTED BY THE USER, IT IS ASSIGNED A DEFAULT VALUE.

ARGUMENTS:

NO-IND-ENT = DSPLY [S9(9)]
MIN-NO-DEP-ENT = DSPLY [S9(9)]
MAX-NO-DEP-ENT = DSPLY [S9(9)]
RTN-STATUS = 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
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.

PS 620141100
1 November 1985

UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

CALLED DIRECTLY BY:

**CRTREL - CONTROLS THE LOGIC FOR VALIDATING AND CREATING
A NEW RELATION CLA**

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

MDDL COMMAND PROCESSOR Module Documentation

NAME: CHKDOMS
PURPOSE: VERIFY THAT THE DATA ITEM AND ATTRIBUTE
USE CLASS.
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CHKDOMS
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CHKDOM WILL VERIFY THAT THE DATA ITEM AND
ATTRIBUTE USE CLASS REFERENCED IN THE VIEW
COMMAND ARE FROM THE SAME DOMAIN CLASS.

ARGUMENTS:

VIEW-NO = DSPLY [S9(9)]
EC-NO = DSPLY [S9(9)]
TAG-NAME = DSPLY [X(30)]
DI-NAME = DSPLY [X(30)]
DATA-TYPE-NAME = DSPLY [X(30)]
RETURN-STATUS = DSPLY [S9(9)]

ROUTINES CALLED:

GETDOM - RETRIEVES DOMAIN NUMBER BASED ON TAG NAME FOR
AUC
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
VERDTD - VERIFY THE EXISTENCE OF A DATA TYPE IN A GIVEN
DOMAIN
INSDI - INSERT A RECORD INTO DATA_ITEM
INSPDI - INSERT A RECORD INTO PROJECT_DATA_ITEM
VERSDT - For a given domain number, return its standard
data type name.

PS 620141100
1 November 1985

CALLED DIRECTLY BY:

PIFROM - CREATE A VIEW USING A SINGLE ENTITY CLASS(
ES-CS-MAPPING)
PNFROM - PROCESS A VIEW COMMAND FOR MULTIPLE ENTITY
CLASSES

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHKINH
PURPOSE: checks the RC-DEPKC table and determines
when a key class c
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CHKINH
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CHKINH checks the RC-DEPKC table and determines
when a key class can be added to a dep-ec.
If a key class for a given rc-no and dep-ec-no does
not appear further down in the RC-DEPKC table
then the key class(es) can be added to the dep-ec-no.

ARGUMENTS:

RC-NO = DSPLY [S9(9)]
DEP-EC-NO = DSPLY [S9(9)]
RC-DEPKC-LIST = RECRD
KC-LIST = RECRD

INCLUDE FILES:

RCDEPKC - LIST OF KEYS MIGRATED VIA A RELATION
KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

CALLED DIRECTLY BY:

ALLKEY - GENERATES KEY CLASS FOR AN ENTITY

PS 620141100
1 November 1985

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHKKEYS
PURPOSE: CHECKS TO SEE IF A KEY CLASS FULFILLED RULES.
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CHKKEYS
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE CHECKS TO SEE IF THE FOLLOWING RULES ARE FULFILLED:
1) A KEY CLASS MUST BE DEFINED FOR EACH ENTITY CLASS
2) MULTIPLE KEY CLASSES OF AN ENTITY CLASS MUST NOT BE SUBSETS OF ONE ANOTHER.

ARGUMENTS:

ORACLE-LDA = RECRD
EC-NO = DSNLY [S9(9)]
EC-NAME = DSNLY [X(30)]
RETURN-STATUS = DSNLY [S9(9)]

INCLUDE FILES:

ORCLEDA - WS DEFINITION FOR THE ORACLE LOGIN AREA
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

OOPEN
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

PS 620141100
1 November 1985

OSQL3
ODFINN
OBINDN
OEXEC
OFETCH
UWARN
ERRPRO

- ISSUE A MESSAGE TO THE USER, CONSIDERED A

CALLED DIRECTLY BY:

GETECS - USING ENTITY CLASS VERIFIES CHECK MODEL RULES

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHKLOOP
PURPOSE: CHECK LOOP DEPENDENCY AND FOR TOPS AND
BOTTOMS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: CHKLOOP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C -- CHKLOOP(MODEL_NAME, &STATUS)

COBOL -- CALL
"CHKLOOP" USING

MODEL-NAME

STATUS.

FORTRAN -- CALL CHKLOOP(MODELNAME, STATUS).

INPUT:

CHAR *MODEL_NAME ;

OUTPUT:

INT *STATUS;

DESCRIPTION

THIS ROUTINE CHECKS FOR LOOP DEPENDENCY. IF A LOOP EXISTS THE MODEL CANNOT BE "CHECKED". THE CHECK CONSISTS OF LOOKING FOR ALL TOP NODES IN THE MODEL, THOSE ENTITIES WHICH ARE DEPENDENT ON NO OTHERS, EVERY MODEL SHOULD HAVE AT LEAST ONE. THEN FOR EACH TOP CALL TLOOPCK

TO CHECK FOR ANY LOOPS IN THE HIERARCHY STARTING AT THAT TOP.
NEXT LOOK FOR ALL BOTTOM ENTITIES, EACH MODEL SHOULD HAVE AT
LEAST ONE BOTTOM. FOR EACH BOTTOM CALL BLOOPCK
TO CHECK FOR ANY LOOPS IN THE HIERARCHY ENDING AT THAT
BOTTOM.

ARGUMENTS:

MODEL NAME - CHAR *
STATUS - INT *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

COSQL3 - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE.
SPRINTF
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A
COFETCH - ORACLE ROUTINE
BLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN UP THE
HIERARCHY
TLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN DOWN THE
HIERARCHY
COEXEC - ORACLE ROUTINE
CODFINN - ORACLE ROUTINE
COBINDN - ORACLE ROUTINE

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: CHKMOD
PURPOSE: DETERMINES WHETHER CERTAIN RULES ARE
FULFILLED,
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CHKMOD
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE DETERMINES WHETHER CERTAIN RULES ARE FULFILLED,
AND IF THEY ARE, THE IDEF MODEL IS MARKED AS "CHECKED".

AFTER EXTRACTING THE MODEL NAME FROM THE PARSER LISTS,
VERIFY THAT THE MODEL EXISTS. THEN CHECK THAT THE RULES
ARE OBSERVED, AND IF THEY ARE, UPDATE THE MODEL AS "CHECKED".

ARGUMENTS:

ORACLE-LDA = RECRD

INCLUDE FILES:

LISTNOS - VALID LIST NUMBERS
ORCLEDA - WS DEFINITION FOR THE ORACLE LOGIN AREA
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.

PS 620141100
1 November 1985

UERROR	- ISSUE A MESSAGE TO THE USER, CONSIDERED A
GETECS	- USING ENTITY CLASS VERIFIES CHECK MODEL RULES
CHKLOOP	- CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS
UPDMOD	- UPDATE MODEL CLASS SET MODEL STATUS = : 1
UWARN	- ISSUE A MESSAGE TO THE USER, CONSIDERED A
ERRPRO	

CALLED DIRECTLY BY:

BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHKOWN
PURPOSE: checks each of the OVN-ec's key classes
against the RC-DEPK
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CHKOWN
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CHKOWN checks each of the OVN-ec's key classes against the RC-DEPKC_LIST. If the key class does not appear anywhere in the table, then the key class is owned by the dep-ec and can be added at this point.

ARGUMENTS:

DEP-EC-NO = DSPLY [S9(9)]
RC-DEPKC-LIST = RECRD
EC-LIST = RECRD
KC-LIST = RECRD
RTN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

RCDEPKC - LIST OF KEYS MIGRATED VIA A RELATION
ECLIST - CONTAINS A LIST OF ENTITY CLASS NUMBERS
KEYLIST - DATA STRUCTURE FOR NDDL MODELLING COMMANDS

ROUTINES CALLED:

NEXTKC - THIS ROUTINE RETURNS A KC_NAME FOR A GIVEN

PS 620141100
1 November 1985

ERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
 EC NO FROM

CALLLED DIRECTLY BY:

ALLKEY - GENERATES KEY CLASS FOR AN ENTITY

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CHKREL
PURPOSE: CHECK IF RELATIONS HAVE BEEN CREATED
ACCORDING TO STANDARDS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CHKREL
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE CHECKS TO SEE IF THE FOLLOWING RULES ARE
FULFILLED:

- 1) NO NON-SPECIFIC RULES
- 2) NO INCOMPLETE RELATION CLASSES
- 3) NO ONE-TO-ONE RELATIONSHIPS.

ARGUMENTS:

EC-NO = DSPLY [S9(9)]
EC-NAME = DSPLY [X(30)]
NDDL-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

RETRECP - RETIEVE AN ENTITY CLASS NAME FOR A GIVEN NO.
AND NAME TYPE
UWARN - ISSUE A MESSAGE TO THE USER, CONSIDERED A

PS 620141100
1 November 1985

VERRCC - verify if the relation class is complete.
ERRPRO

CALLED DIRECTLY BY:

GETECS - USING ENTITY CLASS VERIFIES CHECK MODEL RULES

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: CKDUPEC
PURPOSE: POPULATES A TABLE WITH ENTITY NAMES.
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CKDUPEC
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE IS CALLED FROM MRGMOD2 WHICH IN TURN
IS CALLED BY MRGMOD (MERGE MODEL)
IT POPULATES A TABLE WITH ENTITY NAMES
AFTER VERIFYING THAT THE ENTITY IS NOT ALREADY
PRESENT.

ARGUMENTS:

DEP-EC-NO = DSPLY [S9(9)]
EC-LIST = RECRD
FOUND-FLAG = DSPLY [S9(9)]

INCLUDE FILES:

ECLIST - CONTAINS A LIST OF ENTITY CLASS NUMBERS

CALLED DIRECTLY BY:

MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING
MODEL_2

USED IN MAIN PROGRAM(S):

PS 620141100
1 November 1985

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CKRNLST
PURPOSE: SEARCHES THE TABLE OF RENAME PAIR LOOKING FOR AN OLD-TAG EN
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CKRNLST
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CKRNLST SEARCHES THE TABLE OF RENAME PAIRS LOOKING FOR AN OLD-TAG ENTRY WHICH MATCHES THE TAG-NAME FROM CURRENT KEY MEMBER OCCURRENCE. IF A MATCH IS FOUND, THE RENAME-TAG ENTRY IS MOVED TO NEW-TAG-NAME. IF A MATCH IS NOT FOUND, NEW-TAG-NAME RETAINS ITS DEFAULT VALUE, WHICH IS THE TAG-NAME OF CURRENT KEY MEMBER.

ARGUMENTS:

REN-LIST = RECRD
TAG-NAME = DSPLY [X(30)]
NEW-TAG-NAME = DSPLY [X(30)]

INCLUDE FILES:

RENLIST - LIST OF ATTRIBUTES AND INHERITED TAG PAIRS

CALLED DIRECTLY BY:

ADDMIG - PROCESS THE ADD MIGRATES...SET..CLAUSE

USED IN MAIN PROGRAM(S):

PS 620141100
1 November 1985

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CLSFIL
PURPOSE: THIS ROUTINE CLOSSES AN OUTPUT FILE. THE
FILE WILL
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: CLSFIL
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

CONTAINS THE USER'S NDDL.

SYNOPSIS

C -- CLSFIL(%FILE_NAME, %READ_WRITE, %STATUS)
;
COBOL -- CALL "CLSFIL" USING
FILE-NAME,
READ-WRITE,
STATUS.
FORTRAN -- CALL CLSFIL(FILENAME, READWRITE, STATUS)

INPUT:

OUTPUT:

DESCRIPTION:

INCLUDE FILES:

STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

FCLOSE

CALLED DIRECTLY BY:

CERELS - GENERATES NDDL COMMANDS ON A FILE FOR ALL
ENTITIES IN A RELATION.
CESTRUC - GENERATES NDDL COMMANDS ON A FILE FOR ALL
ENTITIES FOR THE STRUCT
CMBENT - CONROLS THE PROCESSING LOGIC FOR THE COMBINE
ENTITY COMMAND.
CPYMOD - CONTROLS THE PROCESSING LOGIC FOR THE COPY
MODEL COMMAND.
FCOPATT - GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE
COMMAND
MRGMOD - MERGE TWO IDEF MODELS INTO ONE

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CMBACAL
PURPOSE: GENERATE CREATE ALIAS ATTRIBUTE.. AND
ALIAS DESC TEXT COMMA
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CMBACAL
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

This routine selects all the alias for an attribute given the ac-no. Each time an alias name is selected the target model is checked. If the alis does not exist in the target model, NDDL statements to create the alias in the target model are generated. If descriptions are not excepted, NDDL statements to create descriptions of the alias are also generated.

ARGUMENTS:

TO-MOD-NO = DSPLY [S9(9)]
AC-NO = DSPLY [S9(9)]
AC-NAME = DSPLY [X(30)]
DESC-FLAG = DSPLY [9]

INCLUDE FILES:

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

ROUTINES CALLED:

PS 620141100
1 November 1985

VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A
MODEL
GENALI - GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT
TYPE
ERRPRO

CALLED DIRECTLY BY:

CMBOA - GENERATE COMMANDS FOR ATTRIBUTES, ITS
KEYWORDS, ALIAS, DESC

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CMBALI
PURPOSE: GENERATE CREATE ALIAS ENTITY..COMMAND
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CMBALI
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE DETERMINES IF THE FROM-EC-NAME EXISTS IN THE TO-MODEL AS AN ALIAS OF THE TO-EC. IF IT DOESN'T, GENERATE "CREATE ALIAS ENTITY" COMMAND.

ARGUMENTS:

TO-MODEL-NO = DSPLY [S9(9)]
FROM-EC-NO = DSPLY [S9(9)]
TO-EC-NO = DSPLY [S9(9)]
TO-EC-NAME = DSPLY [X(30)]
COMBINE-TYPE = DSPLY [9]

INCLUDE FILES:

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

ROUTINES CALLED:

GENALI - GENERATE A CREATE ALIAS COMMAND FOR AN OBJECT
TYPE
VERALI - VERIFY THE EXISTENCE OF AN ALIAS NAME FOR AN
ENTITY

PS 620141100
1 November 1985

ERRPRO

CALLED DIRECTLY BY:

**GENENT1 - GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND
DESCRS.**

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: CMBEKW
PURPOSE: GENERATE ADD KEYWORD CLAUSE FOR ENTITY
KEYWORDS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CMBEKW
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE GENERATES THE ADD KEYWORD PHRASE IN COMBINE
ENTITY FOR EACH KEYWORD FOUND IN THE FROM MODEL, BUT NOT
FOUND IN THE TO-MODEL.

ARGUMENTS:

FROM-EC-NO - DSPLY [S9(9)]
TO-EC-NO - DSPLY [S9(9)]
CREATE-FLAG - DSPLY [9]
COMBINE-TYPE - DSPLY [9]
KW-COUNT - DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

VERKWE - VERIFY THE EXISTENCE OF ENTITY CLASS KEYWORD
ADPARM1 - CREATES 80 CAHR NDDL COMMANDS WITH PARAMETERS
AND DELIMITERS

PS 620141100
1 November 1985

ERRPRO

CALLED DIRECTLY BY:

**GENENT1 - GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND
DESCRS.**

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CMBENT
PURPOSE: CONROLS THE PROCESSING LOGIC FOR THE
COMBINE ENTITY COMMAND
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CMBENT
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE CONROLS THE PROCESSING LOGIC FOR THE COMBINE ENTITY COMMAND. IT GENERATES THE NECESSARY NDDL COMMANDS TO POPULATE THE TO-MODEL WITH THE ALIASES, DESCRIPTIONS, RELATIONS AND ATTRIBUTES ASSOCIATED WITH THE FROM-MODEL ENTITY. ENTITIES MAY BE COMBINED WITHIN THE SAME MODEL (INTRA MODEL) OR BETWEEN TWO MODELS (INTER-MODEL).

NOTE: 0 - INTRA-MODEL COMBINE
1 - INTER-MODEL COMBINE
0 - USER WANTS KEYWORDS, ALIASES AND/OR DESCRIPTIONS
1 - USER WANTS THE ABOVE EXCEPTED.

ARGUMENTS:

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

INCLUDE FILES:

LISTNOS - VALID LIST NUMBERS
RELTBL - LIST OF RELATION CLASSES IN A MODEL
CHKMODL - DETERMINE IF CURRENT MODEL EXISTS FOR A SESSION
ERRPRO - PROCESS ERROR INCLUDE FILE

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.
EXCFLAG - DETERMINE IF KEYWORDS, ALIASES, DESCR ARE TO BE
EXCLUDED
VERMOD - verify the existence of a model and return the
unique number.
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A
MODEL
OPNFIL
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS
CHECKS/DELIMITER
FRTOREL - DETERMINE IF RELATION EXISTS BETWEEN INTRA-MOD
ENTITIES
BLKCL1 - STORE ALL KEY CLASS INFO FOR AN ENTITY IN A LIST
GEMENT1 - GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND
DESCRS.
DEPFROM - GENERATE CREATE RELATION, DESCRIBE COMMANDS IN
THE TO-MODEL
GENALTE - GENERATE AN ALTER ENTITY..ADD KEY CLASS COMMAND
INDFROM - RETRIEVES RELATIONS, DETERMINES IND EC AND
GENERATES NDDL
CLSFIL - THIS ROUTINE CLOSSES AN OUTPUT FILE. THE FILE
WILL
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

MDDL COMMAND PROCESSOR Module Documentation

NAME: CMBOA
PURPOSE: GENERATE COMMANDS FOR ATTRIBUTES, ITS
KEYWORDS, ALIAS, DESC
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CMBOA
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE GENERATES A CREATE ATTRIBUTE COMMAND FOR EACH OWNED ATTRIBUTE, ALONG WITH THEIR KEYWORDS, ALIASES AND DESCRIPTIONS.

ARGUMENTS:

TO-MOD-NO = DSPLY [S9(9)]
FROM-EC-NO = DSPLY [S9(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]
ATT-COUNT = DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

VERATT - VERIFY THE EXISTENCE OF AN ATTRIBUTE CLASS IN A
MODEL
GENATT - GENERATE A CREATE ATTRIBUTE.. COMMAND
GENDESC - GENERATED NDDL DESCRIBE COMMANDS FOR A GIVEN
OBJECT TYPE AND NO
CMBACAL - GENERATE CREATE ALIAS ATTRIBUTE.. AND ALIAS
DESC TEXT COMMANDS
ADPARM1 - CREATES 80 CAHR NDDL COMMANDS WITH PARAMETERS
AND DELIMITERS
ERRPRO

CALLED DIRECTLY BY:

GENENT1 - GENERATE CREATE/ALTER ENTITY COMMAND, ALIAS, AND
DESCRS.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: CMBRKW
PURPOSE: SELECT AND GENERATE RELATION CLASS
KEYWORDS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CMBRKW
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS ROUTINE SELECTS A RELATION CLASS KEYWORD GIVEN THE
RELATION CLASS NUMBER.

ARGUMENTS:

FROM-RC-NO = DSPLY [S9(9)]
TO-RC-NO = DSPLY [S9(9)]

INCLUDE FILES:

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

ROUTINES CALLED:

VERKWR - VERIFIES THE EXISTANCE OF A RELATION CLASS
KEYWORD.
ADDPARM - WRITES 80 CHAR NDDL COMMAND WITH PARAMENTERS
CHECKS/DELIMITER
ERRPRO

PS 620141100
1 November 1985

CALLED DIRECTLY BY:

DEPFROM - GENERATE CREATE RELATION, DESCRIBE COMMANDS IN
 THE TO-MODEL
INDFROM - RETRIEVES RELATIONS, DETERMINES IND EC AND
 GENERATES NDDL
MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING
 MODEL_2

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: CMPMOD
PURPOSE: CONTROLS THE PROCESSING LOGIC TO COMPARE
TWO MODELS.
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: CMPMOD
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

THIS COMMAND CONTROLS THE PROCESSING LOGIC TO COMPARE TWO
MODELS, AND OUTPUTS THE DIFFERENCES.

AFTER EXTRACTING AND VERIFYING THAT THE 2 MODELS EXIST,
COMPARE THE ENTITIES, ATTRIBUTES AND KEYWORDS WITHIN THE
MODEL.

INCLUDE FILES:

LISTNOS - VALID LIST NUMBERS
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.
VERMOD - verify the existence of a model and return the
unique number.
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
RETRAC1 - COMPARES ENTITY CLASS NAMES FOR TWO MODELS
RETRAC1 - DETERMINES IF ANY ATTRIBUTE CLASS NAMES MATCH
BETWEEN MODELS
ENTKW - SEARCH FOR ENTITY KEYWORD MATCHES WITHIN TWO

PS 620141100
1 November 1985

MODELS
ATTKW - CONTROLS PROCESSING TO POPULATE KEYWORD TABLE
FOR AUC KEYWORD
RELKW - CHECKS RC KW BETWEEN MODELS
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

MDDL COMMAND PROCESSOR Module Documentation

NAME: COBINDM
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORAS4C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

CURSOR = SHORT [32]
SQLVARNUM = INT
PROGVAR = CHAR *
PROGVL = INT
FTYPE = INT

ROUTINES CALLED:

OBINDM

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

AD-A101 954

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS) VOLUME 3
COMMON DATA MODEL S. (U) GENERAL ELECTRIC CO
SCHENECTADY NY PRODUCTION RESOURCES CONSU..

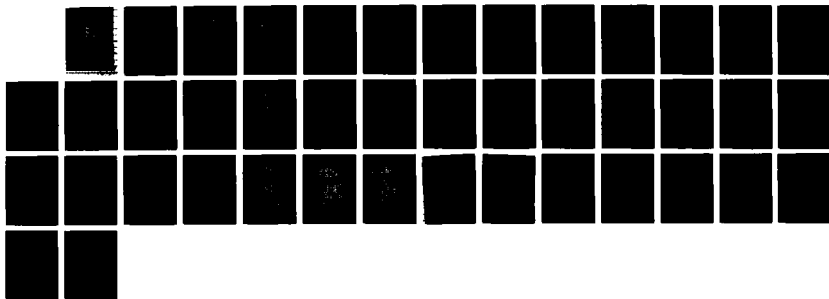
4/4

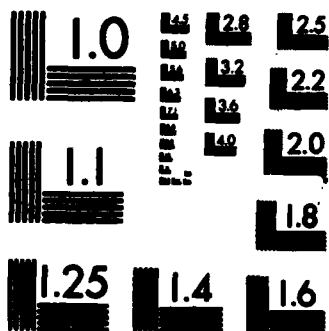
UNCLASSIFIED

S SINGH ET AL. 01 NOV 85 PS-620141100

F/G 12/5

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

BLRCKC1	- FOR EACH LEVEL OF RELATIONS IN STRUCTURE
CHKLOOP	- CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS
DELIPDF	- DELETE A RECORD FROM PROJECT DATA FIELD ENTITY
DELAC	- DELETE A RECORD FROM ATTRIBUTE CLASS
DELACAL	- DELETE RECORD CLASS FROM ATTRIBUTE_NAME
DELACKV	- DELETE A RECORD FROM AC KEYWORD
DELACNM	- DELETE RECORD CLASS FROM ATTRIBUTE_NAME WHERE AC_NO
DELASH	- DELETE FROM ATTRIBUTE USE CLASS MAPPING
DELASH1	- DELETE A RECORD FROM AUC ST MAPPING ENTITY
DELASH2	- DELETE A RECORD FROM AUC ST MAPPING ENTITY
DELAUCL	- DELETE A RECORD FROM ATTRIBUTE USE CL
DELCMPR	- DELETE A RECORD FROM COMPLETE_RELATION
DELCPRC	- DELETE A RECORD FROM COMPLETE_RELATION
DELDA1	- DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDA2	- DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDBA1	- DELETE A RECORD FROM DATA_BASE_AREA ENTITY
DELDBS1	- DELETE A RECORD FROM DATA_BASE_ENTITY
DELDFL1	- DELETE A RECORD FROM DATA_FIELD ENTITY
DELDFL2	- CONTROLS THE DELETING OF DATA FIELDS
DELDFL3	- DELETE A RECORD FROM DATA_FIELD ENTITY
DELDIV	- DELETE FROM DATA_ITEM WHERE
DELDOM	- DELETE RECORD CLASS FROM DOMAIN CLASS
DELDSL1	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELDSL2	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELDSL3	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELDT	- DELETE RECORD CLASS FROM USER_DEF_DATA_TYPE
DELDTD	- DELETE RECORD CLASS FROM USER_DEF_DATA-TYPE
DELEC	- DELETE THE RECORD FROM ENTITY_CLASS
DELECAL	- DELETE RECORD CLASS FROM ENTITY_NAME
DELECKV	- DELETE A RECORD FROM EC KEYWORD
DELECNM	- DELETE THE RECORD FROM ENTITY_NAME
DELIAAM	- DELETE RECORD CLASS FROM AUC ST MAPPING
DELIAUC	- DELETE A REORD FROM INHERITED_ATT_USE
DELIAUK	- DELETE A RECORD FROM INHERITED_ATT_USE ENTITY
DELIPDF	- DELETE RECORD CLASS FROM PROJECT_DATA_FIELD
DELIRCS	- DELETE RECORD CLASS FROM RC_BASED_REC_SET
DELISS1	- DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
DELISS2	- DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
DELKC	- DELETE A RECORD FROM KEY_CLASS
DELKCM	- DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKCMT	- DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKW	- DELETE A RECORD FROM AC_KEYWORD
DELKWAC	- DELETE A RECORD FROM AC_KEYWORD
DELKWEC	- DELETE A RECORD FROM EC_KEYWORD
DELKWRC	- DELETE A RECORD FROM RC_KEYWORD
DELMIGK	- DELETE MIGRATING KEY CLASS

DELMOD - DELETE A RECORD FROM MODEL CLASS ENTITY
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
DELOACE - DELETE A RECORD FROM OWNED_ATTRIBUTE ENTITY
DELOWAC - DELETE A RECORD FROM OWNED_ATTRIBUTE
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
DELRRC - DELETE A RECORD FROM RELATION CLASS
DELRCKW - DELETE A RECORD FROM RC_KEYWORD
DELRCS - DELETE RECORD CLASS FROM RC_BASED_REC_SET
DELREUS -
DELRKM1 - DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRKM2 - DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRKM3 - DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRKY1 - DELETE A RECORD FROM RECORD_KEY ENTITY
DELRKY2 - DELETE A RECORD FROM RECORD_KEY ENTITY
DELRST2 - DELETE A RECORD FROM RECORD_SET ENTITY
DELRST3 - DELETE A RECORD FROM RECORD_SET ENTITY
DELRTY2 - DELETE A RECORD FROM RECORD_TYPE ENTITY
DELSDF1 - DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSDF2 - DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSDF3 - DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSEC - DELETE RECORD CLASS FROM SEC
DELSECR - DELETE RECORD CLASS FROM SEC RC COMPONENT
DELSN1 - DELETE A RECORD FROM SCHEMA_NAMES ENTITY
DELSTM1 - DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSTM2 - DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSTM3 - DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELTEXT - DELETE A RECORD FROM DESC_TEXT ENTITY
DELTXT - DELETE DESCRIPTION TEXT GIVEN THE OBJECT TYPE.
DEPATT - SELECT ALL THE ATTRIBUTES IN THE
DEPENT - SELECT ALL THE DEPENDANT ENTITY CLASSES
DEPREL - FOR EACH LEVEL OF RELATIONS IN STRUCTURE
GENERATE
DLDSL2 - DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DLMIGRC - DELETE MIGRATING KEY CLASS
DPKCLST - CREATE AN KEY_CLASS_LIST TABLE CONTAINING ALL
THE ENTITY
DRPDF - DELETE A RECORD FROM DATA_FIELD ENTITY
GETNXNO -
INSAC - INSERT A RECORD INTO ATTRIBUTE_CLASS
INSACNM - INSERT A RECORD INTO ATTRIBUTE_NAME
INSAREA - INSERT A RECORD INTO THE DATA_BASE AREA ENTITY.
IF

INSAUC - INSERT A RECORD INTO ATTRIBUTE_USE_CL
INSAUCS - INSERT A RECORD INTO AUC_ST_MAPPING
INSCRC - INSERT A RECORD INTO COMPLETE_RELATION
INSDAA - INSERT A RECORD INTO THE DB_AREA_ASSIGNMENT
ENTITY. IF
INSDB - INSERT A RECORD INTO THE DATA_BASE ENTITY. IF
INSDFLD - INSERT A RECORD INTO THE DATA_FIELD ENTITY.
INSDI - INSERT A RECORD INTO DATA_ITEM
INSDOM - INSERT A RECORD INTO DOMAIN_CLASS
INSDSL - INSERT A RECORD INTO THE DF_SET_LINKAGE ENTITY.
IF
INSDT - INSERT A RECORD INTO USER_DEF_DATA_TYPE
INSEC - INSERT A RECORD INTO ENTITY_CLASS
INSECNM - INSERT A RECORD INTO ENTIYT_NAME
INSIAUC - INSERT A RECORD INTO INHERITED_ATT_USE
INSISS - INSERT A RECORD INTO THE IMS_SEGMENT_SIZE
ENTITY. IF
INSKC - INSERT A RECORD INTO KEY_CLASS
INSKCM - INSERT A RECORD INTO KEY_CLASS_MEMBER
INSKW - INSERT A RECORD INTO KEYWORD
INSKWAC - INSERT A RECORD INTO AC_KEYWORD
INSKWEC - INSERT A RECORD INTO EC_KEYWORD
INSKWRC - INSERT A RECORD INTO RC_KEYWORD
INSMOD - INSERT A RECORD INTO MODEL_CLASS
INSOAC - INSERT A RECORD INTO OWNED_ATTRIBUTE
INSPCB - INSERT A RECORD INTO THE PSB_PCB ENTITY. IF
INSPDF - INSERT A RECORD INTO PROJECT_DATA_FIELD
INSPDI - INSERT A RECORD INTO PROJECT_DATA_ITEM
INSPSB - INSERT A RECORD INTO THE PSB ENTITY. IF
INSPWRD - INSERT A RECORD INTO THE DB_PASSWORD. IF
SUCCESSFUL.
INSRC - INSERT A RECORD INTO RELATION_CLASS
INSRCRS - INSERT A RECORD INTO RC_BASED_REC_SET
INSREUS - INSERT A RECORD INTO REUSABLE_NUMBER
INSRKEY - INSERT A RECORD INTO THE RECORD_KEY ENTITY. IF
INSRKM - INSERT A RECORD INTO THE RECORD_KEY_MEMBER
ENTITY. IF
INSRSET - INSERT A RECORD INTO THE RECORD_SET ENTITY. IF
INSRTYP - INSERT A RECORD INTO THE RECORD_TYPE ENTITY. IF
INSSCH - INSERT A RECORD INTO THE SCHEMA_NAMES ENTITY.
IF
INSSDFL - INSERT A RECORD INTO THE SEGMENT_DATA_FIELD
ENTITY. IF
INSSEC - INSERT A RECORD INTO SEC
INSSECR - INSERT A RECORD INTO SEC_RC_COMPONENT
INSSTM - INSERT A RECORD INTO THE SET_TYPE_MEMBER
ENTITY. IF

MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING
MODEL 2
MRGNODE - SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT
ENTITY'S
NRGET -
RDDESC - STORE DESCRIPTION ON THE CDM
SELIAUC - SELECTS ALL THE INHERITED TAG NAMES
TLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN DOWN THE
HIERARCHY
TOPNODE - SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT
ENTITY
UPDAC - UPDATE ATTRIBUTE CLASS SET DOMAIN NO - : 1
UPDACAL - UPDATE MODEL CLASS SET AC NAME TYPE - : 1
UPDACNM - UPDATE ATTRIBUTE NAME SET AC NAME - : 1
UPDECAL - UPDATE MODEL CLASS SET EC NAME TYPE - : 1
UPDECNM - UPDATE MODEL CLASS SET ENTITY NAME - : 1
UPDIND - UPDATE USER_DEF_DATA_TYPE SET DATA_TYPE_IND - :
1
UPDMNAM - UPDATE MODEL CLASS SET MODEL NAME - : 1
UPDMOD - UPDATE MODEL CLASS SET MODEL STATUS - : 1
UPDNXNO -
UPDRCNM - UPDATE RELATION CLASS SET RC NAME - : 1
UPDTDOM - UPDATE DOMAIN SET DOMAIN NAME - : 1
UPDTRC - 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):

DELDFL1 - DELETE A RECORD FROM DATA_FIELD ENTITY
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: COCLOSE
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

CURSOR = SHORT [32]

ROUTINES CALLED:

OCLOSE

CALLED DIRECTLY BY:

TERMSSES - ROUTINE TO TERMINATE AN NDDL

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: COCOF
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

LDA = SHORT [32]

ROUTINES CALLED:

OOCF

CALLED DIRECTLY BY:

INITSES - PERFORM ANY SESSION INITIALIZATION NECESSARY

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: COCOM
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

LDA - SHORT [32]

ROUTINES CALLED:

OCOM

CALLED DIRECTLY BY:

COMMIT - STORE THE REUSEABLE NUMBER TO THE DATA BASE.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: GODFINN
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

CURSOR = SHORT [32]
POS = INT
BUFFER = CHAR *
BUFL = INT
FTYPE = INT

ROUTINES CALLED:

ODFINN

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**
**BLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN UP THE
HIERARCHY**
BLRCKC - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL FIND
BLRCKC1 - FOR EACH LEVEL OF RELATIONS IN STRUCTURE

CHKLOOP	- CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS
DELDL2	- CONTROLS THE DELETING OF DATA FIELDS
DELMIGK	- DELETE MIGRATING KEY CLASS
DELMTKC	- DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
DEPATT	- SELECT ALL THE ATTRIBUTES IN THE
DEPENT	- SELECT ALL THE DEPENDANT ENTITY CLASSES
DEPREL	- FOR EACH LEVEL OF RELATIONS IN STRUCTURE GENERATE
DLMIGRC	- DELETE MIGRATING KEY CLASS
DPKCLST	- CREATE AN KEY CLASS_LIST TABLE CONTAINING ALL THE ENTITY
GETNXNO	-
MRGMOD2	- CONTROL THE LOGIC FOR PROCESSING THE REMAINING MODEL 2
MRGNODE	- SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT ENTITY'S
NRGET	-
SELIAUC	- SELECTS ALL THE INHERITED TAG NAMES
TLOOPCK	- CHECK FOR LOOPS FROM THE ENTITY GIVEN DOWN THE HIERARCHY
TOPNODE	- SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT ENTITY

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: COERMSG
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

RCODE = INT
MSGBUF = CHAR *

ROUTINES CALLED:

OERMSG

CALLED DIRECTLY BY:

ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE

USED IN MAIN PROGRAM(S):

DELDFL1 - DELETE A RECORD FROM DATA_FIELD ENITY
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: COEXEC
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

CURSOR = SHORT [32]

ROUTINES CALLED:

OEXEC

CALLED DIRECTLY BY:

**ALLATT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND
GENERATE**
**ALLENT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND
GENERATE**
**ALLREL - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL
GENERATE**
ALTSMAP - ALTER A SINGLE MAP
**BLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN UP THE
HIERARCHY**
BLRCKC - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL FIND
BLRCKC1 - FOR EACH LEVEL OF RELATIONS IN STRUCTURE
CHKLOOP - CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS
DEL1PDF - DELETE A RECORD FROM PROJECT_DATA FIELD ENITY
DELAC - DELETE A RECORD FROM ATTRIBUTE_CLASS

DELACAL - DELETE RECORD CLASS FROM ATTRIBUTE_NAME
DELACKW - DELETE A RECORD FROM AC_KEYWORD
DELACNM - DELETE RECORD CLASS FROM ATTRIBUTE_NAME WHERE
 AC_NO
DELASM - DELETE FROM ATTRIBUTE USE CLASS MAPPING
DELASM1 - DELETE A RECORD FROM AUC_ST_MAPPING_ENTY
DELASM2 - DELETE A RECORD FROM AUC_ST_MAPPING_ENTY
DELAUCL - DELETE A RECORD FROM ATTRIBUTE_USE_CL
DELCMPR - DELETE A RECORD FROM COMPLETE_RELATION
DELCPRC - DELETE A RECORD FROM COMPLETE_RELATION
DELDA1 - DELETE A RECORD FROM DB_AREA_ASSIGNMENT_ENTY
DELDA2 - DELETE A RECORD FROM DB_AREA_ASSIGNMENT_ENTY
DELDBA1 - DELETE A RECORD FROM DATA_BASE_AREA_ENTY
DELDBS1 - DELETE A RECORD FROM DATA_BASE_ENTY
DELDFL1 - DELETE A RECORD FROM DATA_FIELD_ENTY
DELDFL2 - CONTROLS THE DELETING OF DATA_FIELDS
DELDFL3 - DELETE A RECORD FROM DATA_FIELD_ENTY
DELDIV - DELETE FROM DATA_ITEM WHERE
DELDOM - DELETE RECORD CLASS FROM DOMAIN_CLASS
DELDSL1 - DELETE A RECORD FROM DF_SET_LINKAGE_ENTY
DELDSL2 - DELETE A RECORD FROM DF_SET_LINKAGE_ENTY
DELDSL3 - DELETE A RECORD FROM DF_SET_LINKAGE_ENTY
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
DELEGAL - DELETE RECORD CLASS FROM ENTITY_NAME
DELECKW - DELETE A RECORD FROM EC_KEYWORD
DELECNM - DELETE THE RECORD FROM ENTITY_NAME
DELIASM - DELETE RECORD CLASS FROM AUC_ST_MAPPING
DELIAUC - DELETE A REORD FROM INHERITED_ATT_USE
DELIAUK - DELETE A RECORD FROM INHERITED_ATT_USE_ENTY
DELIPDF - DELETE RECORD CLASS FROM PROJECT_DATA_FIELD
DELIRCS - DELETE RECORD CLASS FROM RC_BASED_REC_SET
DELISS1 - DELETE A RECORD FROM IMS_SEGMENT_SIZE_ENTY
DELISS2 - DELETE A RECORD FROM IMS_SEGMENT_SIZE_ENTY
DELKC - DELETE A RECORD FROM KEY_CLASS
DELKCM - DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKCMT - DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKW - DELETE A RECORD FROM AC_KEYWORD
DELKWAC - DELETE A RECORD FROM AC_KEYWORD
DELKWEC - DELETE A RECORD FROM EC_KEYWORD
DELKWRC - DELETE A RECORD FROM RC_KEYWORD
DELMIGK - DELETE MIGRATING KEY CLASS
DELMOD - DELETE A RECORD FROM MODEL_CLASS_ENTY
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER
DELOACE - DELETE A RECORD FROM OWNED_ATTRIBUTE_ENTY
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
DELRCVK	- DELETE A RECORD FROM RC_KEYWORD
DELRCST	- DELETE RECORD CLASS FROM RC_BASED_REC_SET
DELREUS	-
DELKRM1	- DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELKRM2	- DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELKRM3	- DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELKRY1	- DELETE A RECORD FROM RECORD_KEY ENTITY
DELKRY2	- DELETE A RECORD FROM RECORD_KEY ENTITY
DELRS2	- DELETE A RECORD FROM RECORD_SET ENTITY
DELRS3	- DELETE A RECORD FROM RECORD_SET ENTITY
DELRTY2	- DELETE A RECORD FROM RECORD_TYPE ENTITY
DELSDF1	- DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSDF2	- DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSDF3	- DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSEC	- DELETE RECORD CLASS FROM SEC
DELSECR	- DELETE RECORD CLASS FROM SEC_RC COMPONENT
DELSN1	- DELETE A RECORD FROM SCHEMA_NAMES ENTITY
DELSTM1	- DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSTM2	- DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSTM3	- DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELTEXT	- DELETE A RECORD FROM DESC_TEXT ENTITY
DELTX	- DELETE DESCRIPTION TEXT GIVEN THE OBJECT TYPE.
DEPATT	- SELECT ALL THE ATTRIBUTES IN THE
DEPENT	- SELECT ALL THE DEPENDANT ENTITY CLASSES
DEPREL	- FOR EACH LEVEL OF RELATIONS IN STRUCTURE GENERATE
DLDSL2	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DLMIGRC	- DELETE MIGRATING KEY CLASS
DPKCLST	- CREATE AN KEY_CLASS_LIST TABLE CONTAINING ALL THE ENTITY
DRPDF	- DELETE A RECORD FROM DATA_FIELD ENTITY
GETNXNO	-
INSAC	- INSERT A RECORD INTO ATTRIBUTE_CLASS
INSACNM	- INSERT A RECORD INTO ATTRIBUTE_NAME
INSAREA	- INSERT A RECORD INTO THE DATA_BASE AREA ENTITY. IF
INSAUC	- INSERT A RECORD INTO ATTRIBUTE_USE_CL
INSAUCS	- INSERT A RECORD INTO AUC_ST_MAPPING
INSCRC	- INSERT A RECORD INTO COMPLETE_RELATION
INSDAA	- INSERT A RECORD INTO THE DB_AREA_ASSIGNMENT

ENTITY. IF
INSDB - INSERT A RECORD INTO THE DATA_BASE ENTITY. IF
INSDFLD - INSERT A RECORD INTO THE DATA_FIELD ENTITY.
INSDI - INSERT A RECORD INTO DATA_ITEM
INSDOM - INSERT A RECORD INTO DOMAIN_CLASS
INSDSL - INSERT A RECORD INTO THE DF_SET_LINKAGE ENTITY.
IF
INSDT - INSERT A RECORD INTO USER_DEF_DATA_TYPE
INSEC - INSERT A RECORD INTO ENTITY_CLASS
INSECM - INSERT A RECORD INTO ENTIYT_NAME
INSIAUC - INSERT A RECORD INTO INHERITED_ATT_USE
INSISS - INSERT A RECORD INTO THE IMS_SEGMENT_SIZE
ENTITY. IF
INSKC - INSERT A RECORD INTO KEY_CLASS
INSKCM - INSERT A RECORD INTO KEY_CLASS_MEMBER
INSKW - INSERT A RECORD INTO KEYWORD
INSKWAC - INSERT A RECORD INTO AC_KEYWORD
INSKVEC - INSERT A RECORD INTO EC_KEYWORD
INSKWRC - INSERT A RECORD INTO RC_KEYWORD
INSMOD - INSERT A RECORD INTO MODEL_CLASS
INSOAC - INSERT A RECORD INTO OWNED_ATTRIBUTE
INSPCB - INSERT A RECORD INTO THE PSB_PCB ENTITY. IF
INSPDF - INSERT A RECORD INTO PROJECT_DATA_FIELD
INSPDI - INSERT A RECORD INTO PROJECT_DATA_ITEM
INSPSB - INSERT A RECORD INTO THE PSB_ENTITY. IF
INSPWRD - INSERT A RECORD INTO THE DB_PASSWORD. IF
SUCCESSFUL,
INSRC - INSERT A RECORD INTO RELATION_CLASS
INSRCRS - INSERT A RECORD INTO RC_BASED_REC_SET
INSREUS - INSERT A RECORD INTO REUSABLE_NUMBER
INSRKEY - INSERT A RECORD INTO THE RECORD_KEY ENTITY. IF
INSRKM - INSERT A RECORD INTO THE RECORD_KEY_MEMBER
ENTITY. IF
INSRSET - INSERT A RECORD INTO THE RECORD_SET ENTITY. IF
INSRTYP - INSERT A RECORD INTO THE RECORD_TYPE ENTITY. IF
INSSCH - INSERT A RECORD INTO THE SCHEMA_NAMES ENTITY.
IF
INSSDFL - INSERT A RECORD INTO THE SEGMENT_DATA_FIELD
ENTITY. IF
INSSEC - INSERT A RECORD INTO SEC
INSSECR - INSERT A RECORD INTO SEC_RC_COMPONENT
INSSTM - INSERT A RECORD INTO THE SET_TYPE_MEMBER
ENTITY. IF
MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING
MODEL 2
MRGNODE - SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT
ENTITY'S

NRGET -
SELIAUC - SELECTS ALL THE INHERITED TAG NAMES
TLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN DOWN THE
HIERARCHY
TOPNODE - SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT
ENTITY
UPDAC - UPDATE ATTRIBUTE CLASS SET DOMAIN_NO = : 1
UPDACAL - UPDATE MODEL CLASS SET AC_NAME_TYPE = : 1
UPDACNM - UPDATE ATTRIBUTE NAME SET AC_NAME = : 1
UPDECAL - UPDATE MODEL CLASS SET EC_NAME_TYPE = : 1
UPDECNM - UPDATE MODEL CLASS SET ENTITY_NAME = : 1
UPDIND - UPDATE USER_DEF_DATA_TYPE SET DATA_TYPE_IND = :
1
UPDMNAM - UPDATE MODEL CLASS SET MODEL_NAME = : 1
UPDMOD - UPDATE MODEL CLASS SET MODEL_STATUS = : 1
UPDNXNO -
UPDRGNM - UPDATE RELATION CLASS SET RC_NAME = : 1
UPDTDOM - UPDATE DOMAIN SET DOMAIN_NAME = : 1
UPDTRC - UPDATE USER_DEF_DATA_TYPE SET TYPE_ID = : 1
UPDTKW - UPDATE KEYWORD SET KEYWORD = : 1
UPDTRC - UPDATE RELATION CLASS SET NO_IND_ENT = : 1
UPDVIEW - UPDATE VIEW SET SEC_ID = : 1
WRTDESC - SELECT A RECORD FROM DESC_TEXT ENITY
WRTDSC4 - SELECT A RECORD FROM DESC_TEXT ENITY

USED IN MAIN PROGRAM(S):

DELDFL1 - DELETE A RECORD FROM DATA_FIELD ENITY
NDDL/MAIN - MAIN PROGRAM FOR THE NDDL_COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: COFETCH
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

CURSOR = SHORT [32]

ROUTINES CALLED:

OFETCH

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**
**BLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN UP THE
 HIERARCHY**
BLRCKC - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL FIND
BLRCKC1 - FOR EACH LEVEL OF RELATIONS IN STRUCTURE
CHKLOOP - CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS
DELDL2 - CONTROLS THE DELETING OF DATA FIELDS
DELMIGK - DELETE MIGRATING KEY CLASS
DELMTKC - DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER

PS 620141100
1 November 1985

DEPATT	- SELECT ALL THE ATTRIBUTES IN THE
DEPENT	- SELECT ALL THE DEPENDANT ENTITY CLASSES
DEPREL	- FOR EACH LEVEL OF RELATIONS IN STRUCTURE GENERATE
DLMIGRC	- DELETE MIGRATING KEY CLASS
DPKCLST	- CREATE AN KEY CLASS_LIST TABLE CONTAINING ALL THE ENTITY
GETNXNO	-
MRGMOD2	- CONTROL THE LOGIC FOR PROCESSING THE REMAINING MODEL 2
MRGNODE	- SELECT ALL THE TOP NODE (LEVEL 1) INDEPENDENT ENTITY'S
NRGET	-
SELIAUC	- SELECTS ALL THE INHERITED TAG NAMES
TLOOPCK	- CHECK FOR LOOPS FROM THE ENTITY GIVEN DOWN THE HIERARCHY
TOPNODE	- SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT ENTITY

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: COLOGOF
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORAS4C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

LDA = SHORT [32]

ROUTINES CALLED:

OLOGOF

CALLED DIRECTLY BY:

TERMSSES - ROUTINE TO TERMINATE AN NDDL

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: COLON
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

LDA = SHORT [32]
UID = CHAR *
UIDLEN = INT
PSW = CHAR *
PSWL = INT
AUDIT_FLAG = INT

ROUTINES CALLED:

OLON

CALLED DIRECTLY BY:

INITSES - PERFORM ANY SESSION INITIALIZATION NECESSARY

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

NAME: COMMIT
PURPOSE: STORE THE REUSEABLE NUMBER TO THE DATA
BASE.
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: COMMIT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS

C -- COMMIT(%STATUS) ;
COBOL -- CALL "COMMIT" USING
STATUS.
FORTRAN -- CALL COMMIT (STATUS)

INPUT:
NONE

OUTPUT:
INT *STATUS ;

DESCRIPTION:

THIS ROUTINE USES ORACLE CALLS TO STORE THE REUSEABLE
NUMBERS INTO

THE DATA BASE.

IF SUCCESSFULLY DELETED THE RECORD , THE STATUS AND
RETURN VALUE BOTH

WILL BE 0 , OTHERWISE -1.

ARGUMENTS:

STATUS - INT *

PS 620141100
1 November 1985

ROUTINES CALLED:

INSREUS - INSERT A RECORD INTO REUSABLE_NUMBER
PRINTF
FREE
COCOM - ORACLE ROUTINE
ERRRPT - HANDLE ANY ERROR CODE FROM ORACLE

CALLED DIRECTLY BY:

HALT - HALT WITH 'COMMIT' OR 'ROLLBACK'.
PRCCMD - THIS ROUTINE IS CALLED TO HANDLE ALL

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: COOPEN
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: GDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

CURSOR - SHORT [32]
LDA - SHORT [32]
DBN - INT
DBNLEN - INT
AREASIZ - INT
UID - INT
UIDLEN - INT

ROUTINES CALLED:

OOPEN

CALLED DIRECTLY BY:

INITSES - PERFORM ANY SESSION INITIALIZATION NECESSARY

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

NDDL COMMAND PROCESSOR Module Documentation

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

DESCRIPTION:

THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR COPYING
AN ATTRIBUTE TO A MODEL.

RETRIEVE THE ATTRIBUTE NAME TO BE COPIED. THEN DETERMINE
IF THE COPY WILL BE DONE IMMEDIATELY OR GENERATED ON A FILE.
ALSO DETERMINE WHETHER THE KEYWORDS, ALIASES OR DESCRIPTIONS
ARE TO BE COPIED.

ARGUMENTS:

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

INCLUDE FILES:

LISTNOS - VALID LIST NUMBERS
CHKMODL - DETERMINE IF CURRENT MODEL EXISTS FOR A SESSION
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A

PS 620141100
1 November 1985

- CPFONE - EXTRACT THE FIRST TOKEN FROM THE NAMED LIST
- CPFNXT - ACCESS THE NEXT TOKEN IN A PARSER LIST.
- EXCFLAG - DETERMINE IF KEYWORDS, ALIASES, DESCR ARE TO BE EXCLUDED
- VERMOD - verify the existence of a model and return the unique number.
- VERACNM - RETRIEVE THE DOMAIN FOR AN ATTRIBUTE CLASS
- ICOPATT - INTERACTIVE COPY ATTRIBUTE, WITH KEYWORDS, ALIAS, DESCR.
- FCOPATT - GENERATE NDDL COMMANDS FROM A COPY ATTRIBUTE COMMAND
- ERRPRO

CALLED DIRECTLY BY:

-
- BRANCHR - PERFORMS MULTI-WAY CALL TO THE

USED IN MAIN PROGRAM(S):

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

NDDL COMMAND PROCESSOR Module Documentation

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

DESCRIPTION:

THIS ROUTINE CONTROLS THE PROCESSING LOGIC FOR THE COPY ENTITY COMMAND. FIRST THE INPUTS ARE RETRIEVED AND VALIDATED. THEN ONE OF TWO ROUTINES ARE CALLED DEPENDING ON WHETHER THE COMMAND EXECUTES IMMEDIATELY OR IS TO GENERATE NDDL COMMANDS ON A FILE.

ARGUMENTS:

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

INCLUDE FILES:

LISTNOS - VALID LIST NUMBERS
CHKMODL - DETERMINE IF CURRENT MODEL EXISTS FOR A SESSION
ERRPRO - PROCESS ERROR INCLUDE FILE

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.

PS 620141100
1 November 1985

EXCFLAG - DETERMINE IF KEYWORDS, ALIASES, DESCR ARE TO BE
EXCLUDED
VERENT - VERIFY THE EXISTENCE OF AN ENTITY CLASS IN A
MODEL
ICOPENT - INTERACTIVE COPY ENTITY WITH
ATTRIBUTES, KEYWORDS, ALIAS, DESC
FCOPENT - DETERMINE IF COPY ENTITY WITH STRUCTURE OR
RELATION
VERMOD - verify the existence of a model and return the
unique number.
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: COPYAC
PURPOSE: CREATE AN ATTRIBUTE, ASSOCIATE WITH
ENTITY, ADD KEY CLASSES
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: COPYAC
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

his routine creates an attribute, adds as owned to an entity
and finally adds all key classes for the entity class.

ARGUMENTS:

NEW-EC-NO = DSPLY [S9(9)]
OLD-EC-NO = DSPLY [S9(9)]
MODEL-NO = DSPLY [S9(9)]
KW-FLAG = DSPLY [9]
ALIAS-FLAG = DSPLY [9]
DESC-FLAG = DSPLY [9]

INCLUDE FILES:

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

ROUTINES CALLED:

BLDATT1 - CREATES ATT CLASS AND ATT NAME FOR A
MODEL(CREATE/COPY ATT)
UERROR - ISSUE A MESSAGE TO THE USER, CONSIDERED A
ADDOAC - ADD ATTRIBUTE AS AN OWNED ATTRIBUTE AND AS ATT
USE CLASS

PS 620141100
1 November 1985

KEYLOOK - RETRIVES KEYCLASS NAME AND NUMBER BASED ON TAG
NO
ERRPRO

CALLED DIRECTLY BY:

ICOPENT - INTERACTIVE COPY ENTITY WITH
ATTRIBUTES,KEYWORDS,ALIAS,DESC

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: COROL
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

LDA - SHORT [32]

ROUTINES CALLED:

OROL

CALLED DIRECTLY BY:

ROLBACK - ROLBACK THE TRANSACTIONS.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

DDDL COMMAND PROCESSOR Module Documentation

NAME: COSQL3
PURPOSE: ORACLE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ORA34C
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41100

DESCRIPTION:

SYNOPSIS
SEE ORACLE USER MANUAL

ARGUMENTS:

CURSOR - SHORT [32]
SQLSTATEMENT - CHAR *
SQLLEN - INT

ROUTINES CALLED:

OSQL3

CALLED DIRECTLY BY:

**ALLATT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND
GENERATE**
**ALLENT - SELECT ALL THE ATTRIBUTES IN FROM-MODEL AND
GENERATE**
**ALLREL - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL
GENERATE**
ALTSMAP - ALTER A SINGLE MAP
**BLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN UP THE
HIERARCHY**
BLRCKC - FOR EACH LEVEL OF RELATIONS IN FROM-MODEL FIND
BLRCKC1 - FOR EACH LEVEL OF RELATIONS IN STRUCTURE
CHKLOOP - CHECK LOOP DEPENDENCY AND FOR TOPS AND BOTTOMS

DEL1PDF	- DELETE A RECORD FROM PROJECT_DATA_FIELD ENTITY
DELAC	- DELETE A RECORD FROM ATTRIBUTE_CLASS
DELACAL	- DELETE RECORD CLASS FROM ATTRIBUTE_NAME
DELACKV	- DELETE A RECORD FROM AC_KEYWORD
DELACNM	- DELETE RECORD CLASS FROM ATTRIBUTE_NAME WHERE AC_NO
DELASH	- 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
DELDA1	- DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDA2	- DELETE A RECORD FROM DB_AREA_ASSIGNMENT ENTITY
DELDBA1	- DELETE A RECORD FROM DATA_BASE_AREA ENTITY
DELDBS1	- DELETE A RECORD FROM DATA_BASE_ENTITY
DELDFL1	- DELETE A RECORD FROM DATA_FIELD ENTITY
DELDFL2	- CONTROLS THE DELETING OF DATA FIELDS
DELDFL3	- DELETE A RECORD FROM DATA_FIELD ENTITY
DELDIV	- DELETE FROM DATA_ITEM WHERE
DELDOM	- DELETE RECORD CLASS FROM DOMAIN CLASS
DELDSL1	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELDSL2	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELDSL3	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DELDT	- DELETE RECORD CLASS FROM USER_DEF_DATA_TYPE
DELDTD	- DELETE RECORD CLASS FROM USER_DEF_DATA-TYPE
DELEC	- DELETE THE RECORD FROM ENTITY_CLASS
DELECAL	- DELETE RECORD CLASS FROM ENTITY_NAME
DELECKV	- DELETE A RECORD FROM EC_KEYWORD
DELECNM	- DELETE THE RECORD FROM ENTITY_NAME
DELIA1	- DELETE RECORD CLASS FROM AUC_ST_MAPPING
DELIAUC	- DELETE A REORD FROM INHERITED_ATT_USE
DELIAUK	- DELETE A RECORD FROM INHERITED_ATT_USE ENTITY
DEL1PDF	- DELETE RECORD CLASS FROM PROJECT_DATA_FIELD
DELIRCS	- DELETE RECORD CLASS FROM RC_BASED_REC_SET
DELISS1	- DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
DELISS2	- DELETE A RECORD FROM IMS_SEGMENT_SIZE ENTITY
DELKC	- DELETE A RECORD FROM KEY_CLASS
DELKCM	- DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKCMT	- DELETE A RECORD FROM KEY_CLASS_MEMBER
DELKW	- DELETE A RECORD FROM AC_KEYWORD
DELKWAC	- DELETE A RECORD FROM AC_KEYWORD
DELKVEC	- DELETE A RECORD FROM EC_KEYWORD
DELKWRC	- DELETE A RECORD FROM RC_KEYWORD
DELMIGK	- DELETE MIGRATING KEY CLASS
DELMOD	- DELETE A RECORD FROM MODEL_CLASS ENTITY
DELMTKC	- DELETE EMPTY KEY CLASSES GIVEN THE MODEL NUMBER

DELOACE	- DELETE A RECORD FROM OWNED_ATTRIBUTE ENTITY
DELOWAC	- DELETE A RECORD FROM OWNED_ATTRIBUTE
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
DELRCVK	- DELETE A RECORD FROM RC_KEYWORD
DELRCST	- DELETE RECORD CLASS FROM RC_BASED_REC_SET
DELREUS	-
DELRKM1	- DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRKM2	- DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRKM3	- DELETE A RECORD FROM RECORD_KEY_MEMBER ENTITY
DELRY1	- DELETE A RECORD FROM RECORD_KEY ENTITY
DELRY2	- DELETE A RECORD FROM RECORD_KEY ENTITY
DELRST2	- DELETE A RECORD FROM RECORD_SET ENTITY
DELRST3	- DELETE A RECORD FROM RECORD_SET ENTITY
DELRTY2	- DELETE A RECORD FROM RECORD_TYPE ENTITY
DELSDF1	- DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSDF2	- DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSDF3	- DELETE A RECORD FROM SEGMENT_DATA_FIELD ENTITY
DELSEC	- DELETE RECORD CLASS FROM SEC
DELSECR	- DELETE RECORD CLASS FROM SEC_RC_COMPONENT
DELSN1	- DELETE A RECORD FROM SCHEMA_NAMES ENTITY
DELSTM1	- DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSTM2	- DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELSTM3	- DELETE A RECORD FROM SET_TYPE_MEMBER ENTITY
DELTEXT	- DELETE A RECORD FROM DESC_TEXT ENTITY
DELTXT	- DELETE DESCRIPTION TEXT GIVEN THE OBJECT TYPE.
DEPATT	- SELECT ALL THE ATTRIBUTES IN THE
DEPENT	- SELECT ALL THE DEPENDANT ENTITY CLASSES
DEPREL	- FOR EACH LEVEL OF RELATIONS IN STRUCTURE GENERATE
DLDSL2	- DELETE A RECORD FROM DF_SET_LINKAGE ENTITY
DLMIGRC	- DELETE MIGRATING KEY CLASS
DPKCLST	- CREATE AN KEY_CLASS_LIST TABLE CONTAINING ALL THE ENTITY
DRPDF	- DELETE A RECORD FROM DATA_FIELD ENTITY
GETXNO	-
INSAC	- INSERT A RECORD INTO ATTRIBUTE_CLASS
INSACNM	- INSERT A RECORD INTO ATTRIBUTE_NAME
INSAREA	- INSERT A RECORD INTO THE DATA_BASE AREA ENTITY. IF
INSAUC	- INSERT A RECORD INTO ATTRIBUTE_USE_CL
INSAUCS	- INSERT A RECORD INTO AUC_ST_MAPPING

INSCRC - INSERT A RECORD INTO COMPLETE_RELATION
INSDAA - INSERT A RECORD INTO THE DB_AREA_ASSIGNMENT
ENTITY. IF
INSDB - INSERT A RECORD INTO THE DATA_BASE ENTITY. IF
INSDFLD - INSERT A RECORD INTO THE DATA_FIELD ENTITY.
INSDI - INSERT A RECORD INTO DATA_ITEM
INSDOM - INSERT A RECORD INTO DOMAIN_CLASS
INSDSL - INSERT A RECORD INTO THE DF_SET_LINKAGE ENTITY.
IF
INSDT - INSERT A RECORD INTO USER_DEF_DATA_TYPE
INSEC - INSERT A RECORD INTO ENTITY_CLASS
INSECNM - INSERT A RECORD INTO ENTIYT_NAME
INSIAUC - INSERT A RECORD INTO INHERITED_ATT_USE
INSISS - INSERT A RECORD INTO THE IMS_SEGMENT_SIZE
ENTITY. IF
INSKC - INSERT A RECORD INTO KEY_CLASS
INSKCM - INSERT A RECORD INTO KEY_CLASS_MEMBER
INSKV - INSERT A RECORD INTO KEYWORD
INSKVAC - INSERT A RECORD INTO AC_KEYWORD
INSKVEC - INSERT A RECORD INTO EC_KEYWORD
INSKWORD - INSERT A RECORD INTO RC_KEYWORD
INSMOD - INSERT A RECORD INTO MODEL_CLASS
INSOAC - INSERT A RECORD INTO OWNED_ATTRIBUTE
INSPCB - INSERT A RECORD INTO THE PSB_PCB ENTITY. IF
INSPDF - INSERT A RECORD INTO PROJECT_DATA_FIELD
INSPDI - INSERT A RECORD INTO PROJECT_DATA_ITEM
INSPSB - INSERT A RECORD INTO THE PSB ENTITY. IF
INSPWRD - INSERT A RECORD INTO THE DB_PASSWORD. IF
SUCCESSFUL.
INSRC - INSERT A RECORD INTO RELATION_CLASS
INSRCRS - INSERT A RECORD INTO RC_BASED_REC_SET
INSREUS - INSERT A RECORD INTO REUSABLE_NUMBER
INSRKEY - INSERT A RECORD INTO THE RECORD_KEY ENTITY. IF
INSRKM - INSERT A RECORD INTO THE RECORD_KEY_MEMBER
ENTITY. IF
INSRSET - INSERT A RECORD INTO THE RECORD_SET ENTITY. IF
INSRTYP - INSERT A RECORD INTO THE RECORD_TYPE ENTITY. IF
INSSCH - INSERT A RECORD INTO THE SCHEMA_NAMES ENTITY.
IF
INSSDFL - INSERT A RECORD INTO THE SEGMENT_DATA_FIELD
ENTITY. IF
INSSEC - INSERT A RECORD INTO SEC
INSSECR - INSERT A RECORD INTO SEC_RC_COMPONENT
INSSTM - INSERT A RECORD INTO THE SET_TYPE_MEMBER
ENTITY. IF
MRGMOD2 - CONTROL THE LOGIC FOR PROCESSING THE REMAINING
MODEL_2

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

NRGET -

RDDDESC - STORE DESCRIPTION ON THE CDM

SELIAUC - SELECTS ALL THE INHERITED TAG NAMES

TLOOPCK - CHECK FOR LOOPS FROM THE ENTITY GIVEN DOWN THE
HIERARCHY

TOPNODE - SELECT ALL THE LEVEL 1 TOP NODE INDEPENDENT
ENTITY

UPDAC - UPDATE ATTRIBUTE CLASS SET DOMAIN_NO = : 1

UPDACAL - UPDATE MODEL CLASS SET AC_NAME_TYPE = : 1

UPDACNM - UPDATE ATTRIBUTE NAME SET AC_NAME = : 1

UPDECAL - UPDATE MODEL CLASS SET EC_NAME_TYPE = : 1

UPDECNM - UPDATE MODEL CLASS SET ENTITY_NAME = : 1

UPDIND - UPDATE USER_DEF_DATA_TYPE SET DATA_TYPE_IND = :
1

UPDMNAM - UPDATE MODEL CLASS SET MODEL_NAME = : 1

UPDMOD - UPDATE MODEL CLASS SET MODEL_STATUS = : 1

UPDNXNO -

UPDRCNM - UPDATE RELATION CLASS SET RC_NAME = : 1

UPDTDOM - UPDATE DOMAIN SET DOMAIN_NAME = : 1

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):

DELDFL1 - DELETE A RECORD FROM DATA_FIELD ENTITY

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

MDDL COMMAND PROCESSOR Module Documentation

NAME: CPFCOR
PURPOSE: ACCESS A TOKEN IN A CORRESPONDING 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 -- CPFCOR(LST1, LST2, STRING, RC)

COBOL

-- CALL "CPFCOR" USING
LIST1,
LIST2,
TOKEN,
RC.

FORTRAN --

CALL CPFCOR (LIST1, LIST2, TOKEN, RC)

INPUT:

LST1 -- AN INTEGER IDENTIFYING THE THE CORRESPONDING LIST

LST2 -- THE LIST FROM WHICH THE TOKEN IS DESIRED.

OUTPUT:

STRING --

THE CHARACTER STRING TOKEN FROM LIST2,
BLANK PADDED TO ITS FULL LENGTH.

RC

--
AN INTEGER RETURN CODE, A VALUE OF -1

PS 620141100
1 November 1985

INDICATES A BAD ROW INDEX OR LENGTH
WAS FOUND.

DESCRIPTION

SET UP THE INDEX INTO LIST 2 BY USING THE CURRENT
ENTRY INTO LIST1. BLANK OUT THE STRING OUTPUT AND
COPY IN THE TOKEN FROM LIST2. FINALLY, SET UP THE
ROW ACCESSED AS CURRENT FOR LIST2.

ARGUMENTS:

LST1 - INT *
LST2 - INT *
STRING - CHAR *
RC - INT *

INCLUDE FILES:

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

ROUTINES CALLED:

STRNCPY

CALLED DIRECTLY BY:

BLVWLST - CREATE BUILD VIEW LISTS FOR THE CREATE VIEW
COMMAND
DEFFLD - PROCESSES THE FIELD INFORMATION FOR THE DEFINE
RECORD COMMAND.
DEFKEY - PROCESSES THE KEY INFORMATION FOR THE DEFINE
RECORD COMMAND.
DEFSET - CONTROLS THE PROCESSING LOGIC FOR THE DEFINE
SET COMMAND.
PROCDT - PROGRAM NAME
ADDMAP - ADD A CS-IS MAPPING
ALTSMP - ALTER A SINGLE MAP
DRPSMAP - DROP A SINGLE MAPPING

USED IN MAIN PROGRAM(S):

PS 620141100
1 November 1985

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

PS 620141100
1 November 1985

NDDL COMMAND PROCESSOR Module Documentation

NAME: CPFNXT
PURPOSE: ACCESS THE NEXT TOKEN IN A PARSER 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 --

CPFNXT(LST, STRING, RC)

COBOL

--

CALL "CPFNXT" USING

LIST-NO,

TOKEN,

RC.

FORTRAN

CALL CPFNXT(LIST, TOKEN, RC) --

INPUT:

LST - AN INTEGER IDENTIFYING THE LIST TO BE USED TO
ACCESS THE NEXT TOKEN.

OUTPUT:

STRING - THE TOKEN ACCESSED FROM THE GIVEN LIST.
BLANK PADDED UP TO ITS FULL DEFINED LENGTH.

RC
- THE RETURN CODE WHICH SIGNALS THE END
OF LIST ENCOUNTERED (-1)

DESCRIPTION

USING THE CURRENT LAST ROW RETURNED FROM THE LIST,
ADD 1 AND COMPARE TO THE NUMBER OF ITEMS ON THE
LIST. IF THERE IS ANOTHER, BLANK OUT THE
OUTPUT PARAMETER AND COPY IN THE TOKEN.

ARGUMENTS:

LST = INT *
STRING = CHAR *
RC = INT *

INCLUDE FILES:

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

ROUTINES CALLED:

STRNCPY

CALLED DIRECTLY BY:

ADDATT - ASSOCIATES EXISTING ATT WITH ENTITY IN CREAT
ENTITY COMMAND
ADDKC - CONTROLS PROCESS FOR KEY CLASS CLAUSE FOR
CREATE/ALTER ENTITY
ADDKW - ADDS KEYWORDS FOR COMMANDS USING "ADD
KEYWORD" (OPTIONAL)
ALTAI - CONTROLS ALTER ALIAS PROCESSING (ALIAS TO PRIM
OR VICE VERSA)
ALTCARD - PROCESS CARDINALITY FO USER SPECIFIED RELATION
ALTREL - CONTROLS PROCESSING LOGIC FOR THE ALTER
RELATION COMMAND
BLVWLST - CREATE BUILD VIEW LISTS FOR THE CREATE VIEW
COMMAND

CHKCARD - OBTAINS THE USER-SPECIFIED CARDINALTY (IF ANY)
FOR THE RELATION.

CMBENT - CONROLS THE PROCESSING LOGIC FOR THE COMBINE
ENTITY COMMAND.

CMPMOD - CONTROLS THE PROCESSING LOGIC TO COMPARE TWO
MODELS.

COPATT - CONTROLS THE PROCESSING LOGIC FOR COPYING AN
ATTRIBUTE.

COPENT - CONTROLS THE PROCESSING LOGIC FOR THE COPY
ENTITY COMMAND.

CPYMOD - CONTROLS THE PROCESSING LOGIC FOR THE COPY
MODEL COMMAND.

CRTALI - CREATES ALIASES FOR AN ENTITY OR ATTRIBUTE.
CRTREL - CONTROLS THE LOGIC FOR VALIDATING AND CREATING
A NEW RELATION CLA

DEFAREA - PROCESSES THE AREA INFORMATION IF THE DBMS IS
IDS-II, IDMS AND VA

DEFCODL - PROCESSES THE DBMS TYPES: VAX-11, IDMS, IDS-II.
DEFFLD - PROCESSES THE FIELD INFORMATION FOR THE DEFINE
RECORD COMMAND.

DEFIMS - PROCESSES THE DBMS TYPE: IMS.
DEFKEY - PROCESSES THE KEY INFORMATION FOR THE DEFINE
RECORD COMMAND.

DEFSET - CONTROLS THE PROCESSING LOGIC FOR THE DEFINE
SET COMMAND.

DEFTOT - PROCESSES THE DBMS TYPE: TOTAL.

DRPAC - DELETE OWNED ATTRIBUTES ASSOCIATED WITH ENTITY
DRPATT - CONTROLS THE DROPPING OF USER SPECIFIED
ATTRIBUTE CLASSES FROM TH

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.

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

DRPVIEW - DROP THE VIEW.
GETDBST - RETURN INFORMATION ABOUT THE CURRENT SESSIONS'
DATABASE

MKRNLS - FETCH LIST OF RENAME PAIRS FOR
MIGRATES..SET..CLAUSE

PS 620141100
1 November 1985

MRGMOD - MERGE TWO IDEF MODELS INTO ONE
PROCDT - PROGRAM NAME
RCGHEK - CHECK IF A RELATION CLASS EXISTS FOR A GIVEN
MODEL
RCGHEK1 - CHECK IF A RELATION CLASS EXISTS FOR A GIVEN
MODEL
RENAME - UPDATE EXISTING OBJECT NAME WITH NEW OBJECT NAME
ADDMAP - ADD A CS-IS MAPPING
ALTMAP - ALTER MAP COMMAND PROCESSOR
ALTSMAP - ALTER A SINGLE MAP
CRTMAP - CREATE MAP COMMAND PROCESSOR
DRPMAP - COMMAND PROCESSOR FOR THE DROP MAP COMMAND
DRPSMAP - DROP A SINGLE MAPPING
VEROBJ - VERIFY THAT THE OBJECT EXISTS.

USED IN MAIN PROGRAM(S):

NDDL/MAIN - MAIN PROGRAM FOR THE NDDL COMMAND PROCESSOR

END

7-87

Ditic