# Oracle CDD/Repository™

# Information Model Volume II

Version 5.0

Part No. A24848-1

Information Model Volume II

Version 5.0

Part No. A24848-1

# Contents

## 2 Relation Type Descriptions

## 3 Property Descriptions

## A Type_Hierarchy Structure

## B Implemented Information Model Diagrams

## Index

## Figures



## Tables

# Preface

This manual describes the element types, relation types, and properties that are used directly by specific Digital tool developers. For information on the core of the CDD/Repository element type hierarchy, refer to *CDD /Repository Information Model Volume I*. Volume I and Volume II are the most authoritative sources for predefined repository schema. With this information, users can not only use the existing protocols but also enhance the CDD /Repository information model by extending the Digital-supplied protocols. By sharing these protocols in the CDD/Repository, users optimize software development resources.

This version of the CDD/Repository represents Digital's transition from an entity-relation form of a conceptual schema representation to an object-oriented form. Because of this transition, Volume I is expressed as an object-type hierarchy, and Volume II is expressed as an entity-relation hierarchy. Volume II uses the "supertype" characteristic to map entity descriptions to the object-type hierarchy. See the *CDD/Repository Architecture Manual* for a complete discussion of the CDD/Repositiory Object-Oriented System.

## Intended Audience

This manual is for experienced CDD/Repository users who want to use or enhance the CDD/Repository information model by extending the Digital-supplied protocols. Users must have a working knowledge of the CCD/Repository to interpret the corresponding protocol summary information.

This manual should be used for reference throughout the development process.

## Document Structure

This manual consists of the following chapters and appendixes.

- Chapter 1 contains summary information about each element type.
- Chapter 2 contains summary information about each relation type.
- Chapter 3 contains summary information about each property.

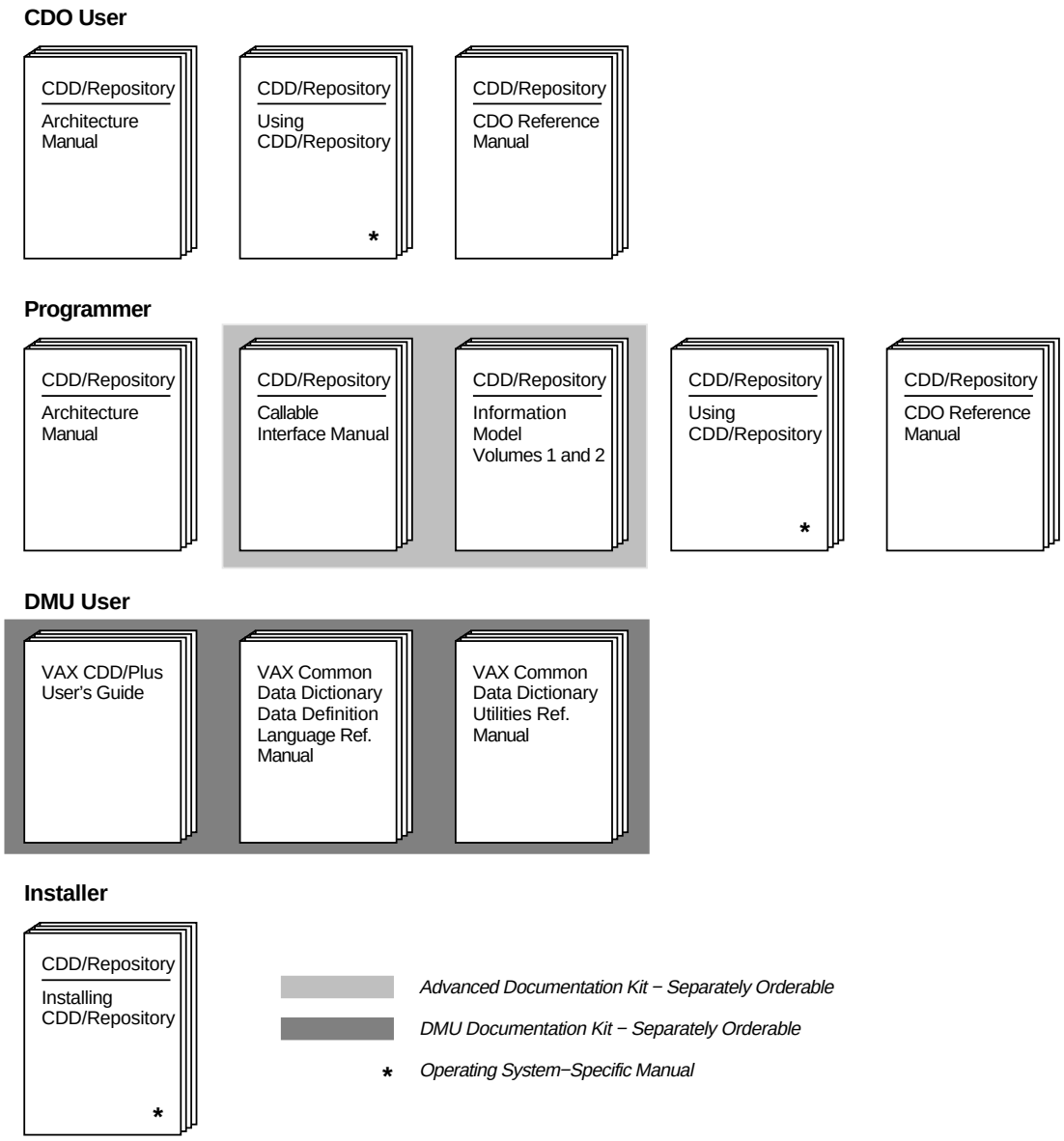
- Appendix A contains the A Tool Integration Standard (ATIS) extension type hierarchy structure.
- Appendix B contains implemented information model (IIM) diagrams for ATIS extension protocols in the CDD data model.

## Associated Documents

See the CDD/Repository Documentation Chart for more information on associated documents and reading paths.

See online help for a glossary of defined terms.

# CDD/Repository Documentation Chart




ZK−3796A−GE

## Conventions

Table 1 shows the conventions used in this manual.

**Table 1 Documentation Conventions**

| Convention | Meaning |
|---|---|
| *italics* | Indicates complete titles of manuals. |
| Ctrl/x | Ctrl/x indicates that you hold down the Ctrl key while you press another key or mouse button (indicated here by x). |
| UPPERCASE | Words in uppercase indicate an element type, relation type, property, command, the name of a file, the name of a file protection code, or an abbreviation for a system privilege. |
| **bold** | The names of messages are set in bold type. |

# 1
# Element Type Descriptions

This chapter contains a description of each element type provided with CDD/Repository, arranged alphabetically. A complete definition of each element type includes properties inherited and methods inherited from the element type's supertype. Inherited properties have been omitted to moderate the size of this document. However, the completed definition can be constructed by adding all defined properties and methods from all supertypes to the list of properties defined for each element type. The descriptions, organized alphabetically, contain the following information in the Summary section:

**Title**—Includes the generic name of the element type and a short phase that describes the type.

**Tag**—Gives the symbolic constant for the value of the **tag** property for this element definition.

**Supertype**—Gives the element type from which another type inherits characteristics. Supertype names with an MCS prefix indicate the boundary between generic and less universally applicable element types.

See Appendix A for the complete element type hierarchy. See Appendix B for the implemented information model (IIM) diagrams.

**Instances Allowed**—Specifies whether or not an instance of this element class exists. Some element classes are defined exclusively to act as "supertypes" to other classes, that is, to describe characteristics intended to be inherited by multiple element subtypes.

**Defined Properties**—Lists properties defined by this element type. Refer to Chapter 3 for a description of each property.

## ACMS$APPLICATION

Describes the highest unit of composition within an Application Control Management System (ACMS) consisting of a list of task groups and tasks comprising the application and control information pertinent to the complete application.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_APPLICATION |
| Instances Allowed | Yes |
| Supertype | ACMS$GROUPS |
| Defined Properties | ACMS$EXTERNAL_LIST<br>ACMS$INTERNAL_CHARACTERISTICS<br>ACMS$OBJECT_SIZE<br>ACMS$PATHNAME<br>ACMS$SOURCE_TEXT |

## ACMS$APPL_SERVER_ITEM

Describes a server item for an ACMS application.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_APPL_SRV_I |
| Instances Allowed | Yes |
| Supertype | ACMS$OBJECTS |
| Defined Properties | ACMS$IDENTIFIER |

### Additional Information

This element type is not currently used.

## ACMS$APPL_TASK_ITEM

Describes a task item for an ACMS application.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_APPL_TSK_I |
| Instances Allowed | Yes |
| Supertype | ACMS$OBJECTS |
| Defined Properties | ACMS$IDENTIFIER |

### Additional Information

This element type is not currently used.

## ACMS$GROUPS

Represents a generic ACMS group.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_GROUPS |
| Instances Allowed | Yes |
| Supertype | MCS_COMPOSITE |
| Defined Properties | None |

### Additional Information

This element type is not currently used.

## ACMS$GROUP_REL

Represents a grouping of ACMS element types.

### Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_GROUP_REL |
| Instances Allowed | Yes |
| Supertype | MCS_COMPOSITE |
| Defined Properties | None |

### Additional Information

This element type is not currently used.

## ACMS$MENU_TASK_ITEM

Represents a task item on an ACMS menu.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_MENU_TSK_I |
| Instances Allowed | Yes |
| Supertype | ACMS$OBJECTS |
| Defined Properties | ACMS$IDENTIFIER |

### Additional Information

This element type is not currently used.

## ACMS$OBJECTS

Provides the supertype for grouping ACMS product components.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_OBJECTS |
| Instances Allowed | Yes |
| Supertype | MCS_VERSION |
| Defined Properties | None |

### Additional Information

This element type is not currently used.

## ACMS$OBJECT_REL

Groups together a set of common relationships for ACMS product components.

### Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_OBJECT_REL |
| Instances Allowed | Yes |
| Supertype | MCS_DEPENDS_ON |
| Defined Properties | None |

### Additional Information

This element type is not currently used.

## ACMS$PROCEDURE

Describes the source code of an ACMS procedure. ACMS procedures are components of ACMS tasks or servers that are written in a conventional programming language to do a specific unit of work.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_PROCEDURE |
| Instances Allowed | Yes |
| Supertype | ACMS$OBJECTS |
| Defined Properties | ACMS$IDENTIFIER |

### Additional Information

This element type is not currently used.

## ACMS$SERVER

Describes the server characteristics as defined in the ACMS task group.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_SERVER |
| Instances Allowed | Yes |
| Supertype | ACMS$OBJECTS |
| Defined Properties | ACMS$EXTERNAL_LIST<br>ACMS$IDENTIFIER<br>ACMS$INTERNAL_CHARACTERISTICS<br>ACMS$OBJECT_SIZE<br>ACMS$PATHNAME<br>ACMS$SOURCE_TEXT |

### Additional Information

This element type is not currently used.

## ACMS$TASK

Describes the complete unit of work to be executed when a user selects that task.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_TASK |
| Instances Allowed | Yes |
| Supertype | ACMS$GROUPS |
| Defined Properties | ACMS$EXTERNAL_LIST<br>ACMS$IDENTIFIER<br>ACMS$INTERNAL_CHARACTERISTICS<br>ACMS$OBJECT_SIZE<br>ACMS$PATHNAME<br>ACMS$SOURCE_TEXT |

## ACMS$TASK_GROUP

Describes an ACMS task group that includes tasks within the group and a set of characteristics relevant to all members of the group. Task groups in ACMS are sets of tasks that share common resources, such as data files and work spaces.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_TASK_GROUP |
| Instances Allowed | Yes |
| Supertype | ACMS$GROUPS |
| Defined Properties | ACMS$EXTERNAL_LIST<br>ACMS$IDENTIFIER<br>ACMS$INTERNAL_CHARACTERISTICS<br>ACMS$OBJECT_SIZE<br>ACMS$PATHNAME<br>ACMS$SOURCE_TEXT |

### Additional Information

Task groups within ACMS are sets of tasks that share common resources such as data files, work spaces, and server processes.

## ACMS$TASK_PROCEDURE_ITEM

Represents a procedure item in an ACMS task.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_PROCEDURE_I |
| Instances Allowed | Yes |
| Supertype | ACMS$OBJECTS |
| Defined Properties | None |

### Additional Information

This element type is not currently used.

## ACMS$TASK_TASK_ITEM

Represents a task item on an ACMS task.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_ACMS_TASK_I |
| Instances Allowed | Yes |
| Supertype | ACMS_OBJECTS |
| Defined Properties | ACMS$IDENTIFIER |

### Additional Information

This element type is not currently used.

## CDD$4GL

Provides the supertype for a subset of element types representing display components.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_4GL |
| Instances Allowed | No |
| Supertype | MCS_COMPOSITE |
| Defined Properties | None |

### Additional Information

This element type should not be instantiated.

## CDD$CDD_DATABASE

Provides the supertype for a subset of element types representing data file structures.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATABASE |
| Instances Allowed | Yes |
| Supertype | MCS_COMPOSITE |
| Defined Properties | CDD$ELEMENT_HAS_ASSOC_KWD |

## CDD$COLLATING_SEQUENCE

Establishes the collating sequence for CDD$DATA_ELEMENT element types by specifying the rule used.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_COL_SEQ |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_DESCRIPTION |
| Defined Properties | CDD$RDB_NCS_NAME |

## CDD$COMPILED_MODULE

Describes a compiled module that represents the output of a compiler operation—the object module as created by a language such as VAX BASIC or VAX COBOL.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_CMP_MODULE |
| Instances Allowed | Yes |
| Supertype | CDD$PROGRAMS |
| Defined Properties | None |

### Additional Information

A CDD$COMPILED_MODULE element type normally owns one or more instances of the CDD$COMPILED_DEPENDS_ON relationship to connect the compiled module to the things it needs at compile time.

The CDD$COMPILED_MODULE element type owns one instance of the CDD$COMPILED_DERIVED_FROM relationship for each source module included in the object module. The member of the relationship is a CDD$SOURCE_MODULE element type.

# CDD$CONSTRAINT

Describes a consistency or integrity constraint applied to related element types that, if not true, cause repository software to disallow storage of the element type.

## Summary

| | |
|---|---|
| Tag | NAD$K_ENT_CONSTRAINT |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_DESCRIPTION |
| Defined Properties | CDD$EVALUATION_TIME |

## Additional Information

The CDD$CONSTRAINT element type must be kept true for all stored data by the underlying data storage mechanism.

# CDD$DATABASE

Describes the linkage between a logical structure and the physical storage of the database. The logical structure can be a record structure, a Digital Standard Relational Interface (DSRI) database structure, a Record Management Services (RMS) file, or a Database Management System (DBMS) database structure.

## Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATABASE |
| Instances Allowed | Yes |
| Supertype | CDD$CDD_DATABASE |
| Defined Properties | CDD$ACCESS_CONTROL_LIST<br>CDD$CREATED_TIME<br>DAF$DRIVER_NAME |

# CDD$DATA_AGGREGATE

Defines a collection of data elements that a program treats as a unit. Typically, data aggregates describe record definitions or Rdb/VMS relation definitions.

## Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATA_AGGREGATE |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_GROUP |
| Defined Properties | CDD$PROCESS_NAME_BAS<br>CDD$DATA_AGG_RDB_CHECK_OPTION<br>CDD$DATA_AGGREGATE_INPUT_PROMPT<br>CDD$DATA_ARRAY_MAJOR_ORDER<br>CDD$PROCESS_NAME_COB<br>CDD$PROCESS_NAME_EBCDIC<br>CDD$PROCESS_NAME_PAS<br>CDD$PROCESS_NAME_PLI<br>CDD$PROCESS_NAME_RPG<br>DTR$FIELD_TREE<br>DTR$SOURCE_TEXT |

## Additional Information

You can use a data aggregate to represent any logical grouping of CDD$ELEMENT (fields), including:

- Record descriptions
- Relation descriptions
- Group fields
- Logical views of fields spanning many physical records

## CDD$DATA_DESCRIPTION

Provides the supertype for a subset of element types representing detailed data structure components.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATA_DESCRIPTION |
| Instances Allowed | No |
| Supertype | MCS_VERSION |
| Defined Properties | None |

### Additional Information

This element type should not be instantiated.

## CDD$DATA_DIMENSION

Describes the repeating characteristics of a single dimension (or subscript) of an array.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATA_DIMENSION |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_DESCRIPTION |
| Defined Properties | CDD$DATA_DIM_HIGH_BOUND_HI_VAL<br>CDD$DATA_DIM_HIGH_BOUND_LOW_VAL<br>CDD$DATA_DIMENSION_HIGH_BOUND<br>CDD$DATA_DIMENSION_LOW_BOUND |

### Additional Information

The CDD$DATA_DIMENSION element type describes the characteristics of a single subscript of an array. Each CDD$DATA_DIMENSION element type is a member of a CDD$DATA_ARRAY_HAS_DIMENSION relationship, which can be owned by either a CDD$DATA_ELEMENT element type or a CDD$DATA_AGGREGATE element type. There is one CDD$DATA_DIMENSION element type for each dimension of the array.

### CDD$DATA_DIMENSION

Each CDD$DATA_DIMENSION element type owns attributes that describe the subscript's upper and lower bounds. If either bound requires a complex expression, the CDD$DATA_DIMENSION element type may own relationships that link it to a CDD$DATA_VALUE element type describing the complex expression. You cannot use both an attribute and a relationship to describe the same bound. For more information, refer to the CDD$DATA_DIM*_REL relationships.

The CDD$DATA_DIMENSION element type describes only a single dimension of an array. To describe arrays with more than one dimension, use a CDD$DATA_ARRAY_HAS_DIMENSION relationship to link a CDD$DATA_ELEMENT element type with a number of CDD$DATA_DIMENSION element types.

## CDD$DATA_ELEMENT

Describes a field, also known as domain, that includes a collection of element types, relationships, and properties describing a field.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATA_ELEMENT |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_GROUP |

Defined Properties

CDD$DATA_ARRAY_MAJOR_ORDER
CDD$DATA_ELEMENT_ALPHA_CASE
CDD$DATA_ELEMENT_CURRENCY_SIGN
CDD$DATA_ELEMENT_DATATYPE
CDD$DATA_ELEMENT_DECIMAL_POINT
CDD$DATA_ELEMENT_DIGITS
CDD$DATA_ELEMENT_DISPLAY_SCALE
CDD$DATA_ELEMENT_COLLATING_SEQ
CDD$DATA_ELEMENT_EDIT_STRING
CDD$DATA_ELEMENT_HELP_TEXT
CDD$DATA_ELEMENT_INITIAL_VALUE
CDD$DATA_ELEMENT_INPUT_REQUIRED
CDD$DATA_ELEMENT_INPUT_PROMPT
CDD$DATA_ELEMENT_JUSTIFICATION
CDD$DATA_ELEMENT_LENGTH
CDD$DATA_ELEMENT_MISSING_VALUE
CDD$DATA_ELEMENT_OUTPUT_HEADER
CDD$DATA_ELEMENT_READ_ONLY
CDD$DATA_ELEMENT_SCALE
CDD$DATA_ELEMENT_SEG_SUBTYPE
CDD$DATA_ELEMENT_SEGMENT_LENGTH
CDD$EDIT_STRING_COBOL
CDD$EDIT_STRING_DTR
CDD$EDIT_STRING_FORMS
CDD$EDIT_STRING_PLI
CDD$EDIT_STRING_RPG
CDD$INPUT_EDIT_STRING
CDD$INPUT_EDIT_STRING_DTR
CDD$INPUT_EDIT_STRING_FORMS
CDD$PROCESS_NAME_BAS
CDD$PROCESS_NAME_COB
CDD$PROCESS_NAME_EBCDIC
CDD$PROCESS_NAME_PAS
CDD$PROCESS_NAME_PLI
CDD$PROCESS_NAME_RPG

## CDD$DATA_GROUP

Provides the supertype for a subset of element types describing components of physical fields and records.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATA_GROUP |
| Instances Allowed | No |
| Supertype | MCS_COMPOSITE |
| Defined Properties | None |

### Additional Information

This element type should not be instantiated.

## CDD$DATA_INSTANCE

Describes a unique instance of a field for use in a complex expression.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATA_INSTANCE |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_DESCRIPTION |
| Defined Properties | None |

### Additional Information

The CDD$DATA_INSTANCE element type is a mechanism for linking a CDD$DATA_VALUE element type to the fully qualified fields on which it depends.

The CDD$DATA_INSTANCE element type can own the CDD$DATA_INSTANCE_PATH relationship and the CDD$DATA_INSTANCE_ROOT relationship.

## CDD$DATA_OVERLAY

Describes an overlay or variant field description.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATA_OVERLAY |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_GROUP |
| Defined Properties | None |

### Additional Information

The CDD$DATA_OVERLAY element type represents an individual overlay. It is the member of a CDD$DATA_OVERLAY_AGG_CONTAINS relationship that links the individual overlay to the CDD$DATA_OVERLAY_AGGREGATE element type representing the collection of overlays.

The CDD$DATA_OVERLAY element type owns one or more instances of CDD$DATA_OVERLAY_CONTAINS or CDD$DATA_OVERLAY_IDENTIFICATION relationships that describe the structure of this overlay.

## CDD$DATA_OVERLAY_AGGREGATE

Describes a set of overlays or variant field descriptions.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DO_AGGREGATE |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_GROUP |
| Defined Properties | None |

### Additional Information

The CDD$DATA_OVERLAY_AGGREGATE element type identifies a set of variant structures that describe alternate interpretations of all or part of a record.

The CDD$DATA_OVERLAY_AGGREGATE element type owns a series of CDD$DATA_OVERLAY_AGG_CONTAINS relationships linking it to the CDD$DATA_OVERLAY element types that describe the individual variants. It is a member of a CDD$DATA_AGGREGATE_CONTAINS relationship linking it to the CDD$DATA_AGGREGATE element type that owns the variant structures.

## CDD$DATA_VALUE

Describes a complex expression.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DATA_VALUE |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_DESCRIPTION |
| Defined Properties | CDD$DATA_VALUE_EXPRESSION |

### Additional Information

The CDD$DATA_VALUE element type provides the mechanism for describing complex characteristics of expressions.

The CDD$DATA_VALUE element type must own the CDD$DATA_VALUE_EXPRESSION property, whose value is the expression being represented.

The CDD$DATA_VALUE element type requires an instance of the CDD$DATA_VALUE_DEPENDS_ON relationship for each field reference in the CDD$DATA_VALUE_EXPRESSION property's value.

## CDD$EXECUTABLE_IMAGE

Describes an executable image that represents the output of linking together one or more CDD$COMPILED_MODULE element types.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_EXE_IMAGE |
| Instances Allowed | Yes |
| Supertype | CDD$PROGRAMS |
| Defined Properties | None |

## CDD$FILE

Describes a physical file.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_BINARY |
| Instances Allowed | Yes |
| Supertype | MCS_BINARY |
| Defined Properties | CDD$QUALIFIED_NAME |

### Additional Information

The CDD$FILE element type contains both the user-supplied file name and the file name that is translated down to a concealed logical name.

The user-supplied name or processing name of the file must be stored. The translation of the logical name is optional and can be stored and used by products that need this information.

All files that are part of an application for storage of control information, programs, and data should be represented in CDD/Repository by this protocol.

## CDD$FILE_ACCESS

Describes the detailed record access block (RAB) for an RMS file.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_FILE_ACCESS |
| Instances Allowed | Yes |
| Supertype | CDD$FILE_ATTS |
| Defined Properties | CDD$RMS_RAB_RAC<br>CDD$RMS_RAB_ROP_RAH<br>CDD$RMS_RAB_ROP_REA<br>CDD$RMS_RAB_ROP_RLK<br>CDD$RMS_RAB_ROP_RNE<br>CDD$RMS_RAB_ROP_RNF<br>CDD$RMS_RAB_ROP_RRL<br>CDD$RMS_RAB_ROP_TMO<br>CDD$RMS_RAB_ROP_TPT<br>CDD$RMS_RAB_ROP_UIF<br>CDD$RMS_RAB_ROP_ULK<br>CDD$RMS_RAB_ROP_WAT<br>CDD$RMS_RAB_ROP_WBH<br>CDD$RMS_RAB_TMO |

## CDD$FILE_ALLOCATION

Describes the RMS extended control of file disk space allocation.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_FILE_ALLOCATION |
| Instances Allowed | Yes |
| Supertype | CDD$FILE_ATTS |
| Defined Properties | CDD$RMS_XABALL_AID<br>CDD$RMS_XABALL_ALN<br>CDD$RMS_XABALL_ALQ<br>CDD$RMS_XABALL_AOP_CBT<br>CDD$RMS_XABALL_AOP_CTG<br>CDD$RMS_XABALL_AOP_HRD<br>CDD$RMS_XABALL_AOP_ONC<br>CDD$RMS_XABALL_BKZ<br>CDD$RMS_XABALL_DEQ<br>CDD$RMS_XABALL_LOC<br>CDD$RMS_XABALL_RFI<br>CDD$RMS_XABALL_VOL |

## CDD$FILE_ATTS

Provides the supertype for a subset of element types describing file management characteristics.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_FILE_ATTS |
| Instances Allowed | To be supplied. No |
| Supertype | MCS_VERSION |
| Defined Properties | None |

### Additional Information

This element type should not be instantiated.

# CDD$FILE_DEFINITION

Describes the logical structure of an RMS file.

## Summary

| | |
|---|---|
| Tag | NAD$K_ENT_FILE_DEFINITION |
| Instances Allowed | Yes |
| Supertype | CDD$FILE_ATTS |
| Defined Properties | CDD$RMS_FAB_ALQ<br>CDD$RMS_FAB_BKS<br>CDD$RMS_FAB_BLS<br>CDD$RMS_FAB_CHAN_MODE<br>CDD$RMS_FAB_FAC_BIO<br>CDD$RMS_FAB_FAC_BRO<br>CDD$RMS_FAB_FAC_DEL<br>CDD$RMS_FAB_FAC_GET<br>CDD$RMS_FAB_FAC_PUT<br>CDD$RMS_FAB_FNS<br>CDD$RMS_FAB_FOP_CBT<br>CDD$RMS_FAB_FOP_CIF<br>CDD$RMS_FAB_FOP_DLT<br>CDD$RMS_FAB_FOP_DFW<br>CDD$RMS_FAB_FOP_MXV<br>CDD$RMS_FAB_FOP_NEF<br>CDD$RMS_FAB_FOP_NFS<br>CDD$RMS_FAB_FOP_POS<br>CDD$RMS_FAB_FOP_RCK<br>CDD$RMS_FAB_FOP_RWC<br>CDD$RMS_FAB_FOP_RWO<br>CDD$RMS_FAB_FOP_SCF<br>CDD$RMS_FAB_FOP_SQO<br>CDD$RMS_FAB_FOP_TEF |

CDD$RMS_FAB_FOP_TMD
CDD$RMS_FAB_FOP_TWD
CDD$RMS_FAB_FOR_UFO
CDD$RMS_FAB_FOP_WCK
CDD$RMS_FAB_DEQ
CDD$RMS_FAB_DNA
CDD$RMS_FAB_DNS
CDD$RMS_FAB_FAC_TRN
CDD$RMS_FAB_FAC_UPD
CDD$RMS_FAB_FNA
CDD$RMS_FAB_GBC
CDD$RMS_FAB_LNM_MODE
CDD$RMS_FAB_MBF
CDD$RMS_FAB_MRN
CDD$RMS_FAB_MRS
CDD$RMS_FAB_ORG
CDD$RMS_FAB_RAT
CDD$RMS_FAB_RAT_BLK
CDD$RMS_FAB_RFM
CDD$RMS_FAB_SHR_DEL
CDD$RMS_FAB_SHR_GET
CDD$RMS_FAB_SHR_PUT
CDD$RMS_FAB_SHR_UPD
CDD$RMS_FAB_BKS
CDD$RMS_FAB_FAC_BIO
CDD$RMS_FAB_FAC_BRO
CDD$RMS_FAB_LNM_MODE
CDD$RMS_FAB_ORG
CDD$RMS_FAB_RAC
CDD$RMS_FAB_RAT
CDD$RMS_FAB_RAT_BLK
CDD$RMS_FAB_RFM
CDD$RMS_FAB_MRN
CDD$RMS_FAB_RTV
CDD$RMS_FAB_SHR_NIL
CDD$RMS_FAB_SHR_MSE
CDD$RMS_FAB_SHR_UPI
CDD$RMS_RAB_BKT
CDD$RMS_RAB_MBC
CDD$RMS_RAB_PBF
CDD$RMS_RAB_PSZ

CDD$RMS_RAB_RAC
CDD$RMS_RAB_RFA
CDD$RMS_RAB_ROB_ASY
CDD$RMS_RAB_ROB_BIO
CDD$RMS_RAB_ROB_CCO
CDD$RMS_RAB_ROB_CVT
CDD$RMS_RAB_ROB_EOF
CDD$RMS_RAB_ROB_ETO
CDD$RMS_RAB_ROB_FDL
CDD$RMS_RAB_ROB_KGE
CDD$RMS_RAB_ROB_KGT
CDD$RMS_RAB_ROB_LIM
CDD$RMS_RAB_ROB_LOA
CDD$RMS_RAB_ROB_LOC
CDD$RMS_RAB_ROB_NLK
CDD$RMS_RAB_ROB_NXR
CDD$RMS_RAB_ROB_PMT
CDD$RMS_RAB_ROB_PTA
CDD$RMS_RAB_ROB_RAH
CDD$RMS_RAB_ROB_REA
CDD$RMS_RAB_ROB_RLK
CDD$RMS_RAB_ROB_RNE
CDD$RMS_RAB_ROB_RNF
CDD$RMS_RAB_ROB_RRL
CDD$RMS_RAB_ROB_TMO
CDD$RMS_RAB_ROB_TPT
CDD$RMS_RAB_ROB_UIF
CDD$RMS_RAB_ROB_ULK
CDD$RMS_RAB_ROB_WAT
CDD$RMS_RAB_ROB_WBH
CDD$RMS_RAB_TMO
CDD$RMS_RAB_MBF
CDD$RMS_XABPRO_GRP_NODEL
CDD$RMS_XABPRO_GRP_NOEXE
CDD$RMS_XABPRO_GRP_NOREAD
CDD$RMS_XABPRO_GRP_NOWRITE
CDD$RMS_XABPRO_MTACC
CDD$RMS_XABPRO_OWN_NODEL
CDD$RMS_XABPRO_OWN_NOEXE
CDD$RMS_XABPRO_OWN_NOREAD
CDD$RMS_XABPRO_OWN_NOWRITE
CDD$RMS_XABPRO_PROPAGATE

CDD$RMS_XABPRO_SYS_NODEL
CDD$RMS_XABPRO_SYS_NOEXE
CDD$RMS_XABPRO_SYS_NOREAD
CDD$RMS_XABPRO_SYS_NOWRITE
CDD$RMS_XABPRO_WLD_NODEL
CDD$RMS_XABPRO_WLD_NOEXE
CDD$RMS_XABPRO_WLD_NOREAD
CDD$RMS_XABPRO_WLD_NOWRITE

### Additional Information

The properties of the CDD$FILE_DEFINITION element type correspond to properties that may be specified through the CDO utility's DEFINE RMS_DATABASE command. Refer to the *VMS Record Management Services Reference Manual* for information about how to use these properties to describe RMS databases in CDD/Repository.

## CDD$FILE_INDEX

Describes the logical structure of the indexes in an RMS file.

### CDD$FILE_INDEX

#### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_FILE_INDEX |
| Instances Allowed | Yes |
| Supertype | CDD$FILE_ATTS |
| Defined Properties | CDD$RMS_XABKEY_CHG<br>CDD$RMS_XABKEY_DAT_NCMPR<br>CDD$RMS_XABKEY_DLF<br>CDD$RMS_XABKEY_DUP<br>CDD$RMS_XABKEY_DTP<br>CDD$RMS_XABKEY_IDX_NCMPR<br>CDD$RMS_XABKEY_IFL<br>CDD$RMS_XABKEY_KEY_NCMPR<br>CDD$RMS_XABKEY_KNM<br>CDD$RMS_XABKEY_NUL<br>CDD$RMS_XABKEY_NULL_VALUE<br>CDD$RMS_XABKEY_PROLOG |

#### Additional Information

This element type is not identified in the reference manual but is readable by using the SHOW GENERIC command and can be updated by using the call interface.

---

## CDD$INDEX

Describes an index access mechanism for a DSRI compliant database system.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_INDEX |
| Instances Allowed | Yes |
| Supertype | CDD$DATA_DESCRIPTION |
| Defined Properties | CDD$UNIQUE_INDEX |

## CDD$LINK_TYPE

Provides the link between the owner relation type and the member relation type.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_LINK_TYPE |
| Instances Allowed | Yes |
| Supertype | MCS_TYPE |
| Defined Properties | CDD$LINK_ORDER<br>CDD$LINK_INDEX_FIELDNAME<br>CDD$LINK_TYPE_FIELDNAME |

### Additional Information

This element type is intended for internal use only.

## CDD$MENU

Represents a generic menu.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_MENU |
| Instances Allowed | Yes |
| Supertype | CDD$4GL |
| Defined Properties | ACMS$EXTERNAL_LIST<br>ACMS$INTERNAL_CHARACTERISTICS<br>ACMS$OBJECT_SIZE<br>ACMS$PATHNAME<br>ACMS$SOURCE_TEXT |

### Additional Information

It is not currently used.

## CDD$PROCEDURE

Represents a generic procedure.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_PROCEDURE |
| Instances Allowed | Yes |
| Supertype | CDD$PROGRAMS |
| Defined Properties | None |

### Additional Information

It is not currently used.

## CDD$PROGRAMS

Provides the supertype for a subset of element types representing source, object, and executable code units.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_PROGRAMS |
| Instances Allowed | To be supplied. No |
| Supertype | MCS_VERSION |
| Defined Properties | None |

### Additional Information

This element type should not be instantiated.

## CDD$RDB_DATABASE

Describes a logical structure of a DSRI database that may be thought of as an aggregate of objects that are record-oriented.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_RDB_DATABASE |
| Instances Allowed | Yes |
| Supertype | CDD$CDD_DATABASE |
| Defined Properties | CDD$ACCESS_CONTROL_LIST<br>CDD$CREATED_TIME<br>CDD$DATABASE_PARAMETERS |

## CDD$REPORT

Represents a generic report.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_REPORT |
| Instances Allowed | Yes |
| Supertype | CDD$4GL |
| Defined Properties | None |

### Additional Information

It is not currently used.

## CDD$RMS_DATABASE

Describes the logical structure of an RMS file that may be thought of as an aggregate of objects.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_RMS_DATABASE |
| Instances Allowed | Yes |
| Supertype | CDD$CDD_DATABASE |
| Defined Properties | CDD$ACCESS_CONTROL_LIST<br>CDD$CREATED_TIME |

## CDD$SOURCE_MODULE

Describes a module of source text.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_SRC_MODULE |
| Instances Allowed | Yes |
| Supertype | MCS_TEXT |
| Defined Properties | None |

## CDD$VIDEO_DISPLAY

Represents a DECforms user-visible display.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_VIDEO_DISP |
| Instances Allowed | Yes |
| Supertype | CDD$4GL |
| Defined Properties | None |

## DBM$AREA

Represents a named database storage area file.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DBM_AREA |
| Instances Allowed | Yes |
| Supertype | DBM$OBJECT |
| Defined Properties | None |

## DBM$OBJECT

Provides a supertype for a set of element types that represent components of DBMS and other hierarchic database products.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DBM_OBJECT |
| Instances Allowed | No |
| Supertype | MCS_VERSION |
| Defined Properties | None |

### Additional Information

This element type should not be instantiated.

## DBM$REALM

Represents a logical subschema mapping to physical schema storage files, and maps directly to one or more database schema area element types.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DBM_REALM |
| Instances Allowed | Yes |
| Supertype | DBM$OBJECT |
| Defined Properties | None |

## DBM$SCHEMA

Contains the internal representation of a DBMS schema definition and maintains dictionary links to all the following associated database component definitions:

Subschemas
Storage schemas
Security schemas
Storage areas

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DBM_SCHEMA |
| Instances Allowed | Yes |
| Supertype | CDD$CDD_DATABASE |
| Defined Properties | DBM$DDBLK<br>DBM$DDNAME<br>DBM$DDSET |

## DBM$SECURITY_SCHEMA

Contains the internal representation for the DBMS security schema definition.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DBM_SEC_SCHEMA |
| Instances Allowed | Yes |
| Supertype | CDD$CDD_DATABASE |
| Defined Properties | DBM$DDBLK<br>DBM$DDNAME<br>DBM$DDSET<br>DBM$RELSCH_STAMP |

## DBM$SET

Represents a named database subschema set structure and maps to record definitions, such as owners and members of the defined subschema set structure.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DBM_SET |
| Instances Allowed | Yes |
| Supertype | CDD$CDD_DATABASE |
| Defined Properties | None |

## DBM$STORAGE_SCHEMA

Contains the internal representation for the DBMS storage schema definition.

### DBM$STORAGE_SCHEMA

#### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DBM_STO_SCHEMA |
| Instances Allowed | Yes |
| Supertype | CDD$CDD_DATABASE |
| Defined Properties | None |

## DBM$SUBSCHEMA

Contains the internal representation for the DBMS subschema definition as well as maintains dictionary links to all the following associated database component definitions:

Realms
Sets
Records

#### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DBM_SUBSCHEMA |
| Instances Allowed | Yes |
| Supertype | CDD$CDD_DATABASE |
| Defined Properties | DBM$DDBLK<br>DBM$DDNAME<br>DBM$DDSET<br>DBM$RELSCH_STAMP |

## DTR$DATABASE

Describes the pointer to the Rdb/VMS or DBMS database.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DTR_DATABASE |
| Instances Allowed | Yes |
| Supertype | CDD$4GL |
| Defined Properties | DTR$DTR_DB_FOR_PROCEDURE<br>DTR$DTR_DB_HAS_SCHEMA<br>DTR$DTR_DB_HAS_SUBSCHEMA<br>DTR$FILE<br>DTR$SCHEMA<br>DTR$SUBSCHEMA<br>DTR$SOURCE_FOR_PROCEDURE<br>DTR$SOURCE_TEXT |

## DTR$DOMAIN

Describes a domain definition that is used within VAX Datatrieve.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DTR_DOMAIN |
| Instances Allowed | Yes |
| Supertype | MCS_COMPOSITE |
| Defined Properties | DTR$DATABASE_PATHNAME<br>DTR$DOMAIN_TYPE<br>DTR$FILE<br>DTR$FORM_LIBRARY<br>DTR$FORM_NAME<br>DTR$NODE<br>DTR$RECORD_PATHNAME<br>DTR$SOURCE_TEXT |

## DTR$PLOT

Describes the procedure for producing Remote Graphics Instruction Set (ReGIS) graphics.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DTR_PLOT |
| Instances Allowed | Yes |
| Supertype | CDD$4GL |
| Defined Properties | DTR$PLOT_ARGS<br>DTR$PLOT_CODE<br>DTR$SOURCE_TEXT |

## DTR$PROCEDURE

Describes the source code of a procedure written in VAX Datatrieve.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DTR_PROCEDURE |
| Instances Allowed | Yes |
| Supertype | CDD$4GL |
| Defined Properties | DTR$PROCEDURE_HAS_CDD_DB<br>DTR$PROCEDURE_HAS_DTR_DB<br>DTR$PROCEDURE_HAS_FIELD<br>DTR$PROCEDURE_HAS_SOURCE<br>DTR$PROCEDURE_HAS_TABLE DTR$SOURCE_<br>TEXT |

## DTR$TABLE

Describes a translation table used within VAX Datatrieve to associate descriptions to be used when printing data to encoded values that are stored in files.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_DTR_TABLE |
| Instances Allowed | Yes |
| Supertype | CDD$4GL |
| Defined Properties | DTR$CODE_FIELD<br>DTR$CODES<br>DTR$DATA_ELEMENT_OUTPUT_HEADER<br>DTR$DESCRIPTION_FIELD<br>DTR$DESCRIPTIONS<br>DTR$DOMAIN_PATH<br>DTR$EDIT_STRING_DTR<br>DTR$SOURCE_TEXT<br>DTR$TABLE_FOR_PROCEDURE<br>DTR$TABLE_HAS_SOURCE |

## GEN$PROGRAM

Represents a proxy element type for a VAX COBOL generator session file.

### Summary

| | |
|---|---|
| Tag | GEN$K_ENT_PROGRAM |
| Instances Allowed | Yes |
| Supertype | CDD$PROGRAMS |
| Defined Properties | None |

## RALLY$APPLICATION

Represents an element type for a single RALLY application.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_RALLY_APP |
| Instances Allowed | Yes |
| Supertype | MCS_COMPOSITE |
| Defined Properties | None |

### Additional Information

You can create this element type by using by the DCL command RALLY CREATE/DATA_DEPENDENCY or by using the RALLY INTEGRATE utility.

The RALLY$APPLICATION element type owns one RALLY$APPLICATION_CONTAINS relationship per data source defintion within the RALLY application and a single CDD$IN_FILE relationship. The CDD$IN_FILE relationship owns a CDD$FILE element type which specifies the name of the .RGA file that contains the RALLY application.

The RALLY$APPLICATION is a proxy element type, that is, there is not enough information stored in the repository to recreate the application, only enough to hold the history and pieces tracking relationships.

The RALLY$APPLICATION owns the RALLY$APPLICATION_CONTAINS relation and the CDD$IN_FILE relation.

## RALLY$DATA_SOURCE_DEFINITION

Contains information about a RALLY data source definition.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_RALLY_DSD |
| Instances Allowed | Yes |
| Supertype | MCS_COMPOSITE |
| Defined Properties | None |

### Additional Information

For Rdb/VMS and RMS data source defintion, this element type has a relationship to the CDD$DATABASE which is referenced by the RALLY data source definition.

The RALLY$DATA_SOURCE_DEFINITION element type is a proxy element type, that is, there is not enough information stored in the repository to recreate the data source definition, only enough to hold the history and pieces tracking relationships.

The RALLY$DATA_SOURCE_DEFINITION element type owns the RALLY$DSD_SOURCE relation and is owned by the RALLY$APPLICATION_CONTAINS relation.

## RALLY$OBJECTS

Provides the supertype for grouping RALLY product components in the CDD/Repository type hierarchy.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_RALLY_OBJS |
| Instances Allowed | Yes |
| Supertype | MCS_VERSION |
| Defined Properties | None |

## RALLY$PACKET

Represents a RALLY communication object.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_RALLY_PACKET |
| Instances Allowed | Yes |
| Supertype | RALLY$OBJECTS |
| Defined Properties | None |

### Additional Information

This element type is not currently used.

## RALLY$PROCEDURE

Represents a RALLY procedure.

### Summary

| | |
|---|---|
| Tag | NAD$K_ENT_RALLY_ADL |
| Instances Allowed | Yes |
| Supertype | RALLY$OBJECTS |
| Defined Properties | None |

### Additional Information

This element type is not currently used.

# RALLY$TASK

Represents a RALLY task.

## Summary

| | |
|---|---|
| Tag | NAD$K_ENT_RALLY_TASK |
| Instances Allowed | Yes |
| Supertype | RALLY$OBJECTS |
| Defined Properties | None |

## Additional Information

This element type is not currently used.

# 2

# Relation Type Descriptions

This chapter contains a description of the relation types provided with CDD/Repository, arranged alphabetically. The descriptions contain the following information in the Summary section:

**Title**—Includes the generic name of the relation type and a short phase describing the type.

**Defined Properties**—Lists properties defined by this relation type. Refer to Chapter 3 for a description of each property.

**Notification Dependency**—Provides notification of change messages to object types "owning" relations based on change to "member" object types. This subsystem operates by recording a change notification selectively by type of relation. The notification dependency characteristics are as follows:

- **Blocked**—passed through but not recorded.
- **Success**—passed through successfully. If a new version of the element is created, CDD/Repository interprets the notification dependency characteristic as though it were Signal and sends a notice to the owner of the relationship. If the element is changed without creating a new version, Success indicates that the owner of the relationship does not receive a notice. However, CDD/Repository tries to send the notice to other elements that own the owner.
- **Signal**—appended to the relation owner and passed to owners of the immediate owner.

Refer to the chapter on using notices in the *CDD/Repository Callable Interface Manual* for further information.

**Owning Element Types**—Specifies the owning element type.

**Member Object Types**—Specifies the member element type.

# ACMS$APPLICATION_ADB_FILE

Establishes the relationship between ACMS$APPLICATION and the representation of the application definition binary (ADB) file.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_APP_ADB |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$APPLICATION |
| Members | CDD$SOURCE_MODULE |

# ACMS$APPLICATION_SERVER

Establishes the relationship between ACMS$APPL_SERVER_ITEM and ACMS$TASK_GROUP.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_APP_SERVER |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$APPLICATION |
| Members | ACMS$APPL_SERVER_ITEM |

# ACMS$APPLICATION_SRV_GRP

Establishes the relationship between ACMS$APPL_SERVER_ITEM and ACMS$APPL_SERVER.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_APP_SRV_GRP |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$APPL_SERVER_ITEM |
| Members | ACMS$TASK_GROUP |

# ACMS$APPLICATION_SRV_SRV

Establishes the relationship between ACMS$APPL_SERVER_ITEM and ACMS$APPL_SERVER.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_APP_SRV_SRV |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$APPL_SERVER_ITEM |
| Members | ACMS$APPL_SERVER |

# ACMS$APPLICATION_TASK

Establishes the relationship between ACMS$APPLICATION and ACMS$APPL_TASK_ITEM.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_APP_TASK |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$APPLICATION |
| Members | ACMS$APPL_TASK_ITEM |

# ACMS$APPLICATION_TASK_GROUP

Establishes the relationship between ACMS$APPLICATION and ACMS$TASK_GROUP.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_APP_TK_GRP |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$APPLICATION |
| Members | ACMS$TASK_GROUP |

# ACMS$APPLICATION_TASK_GRP

Establishes the relationship between ACMS$APPL_TASK_ITEM and ACMS$TASK_GROUP.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_APP_TSK_GRP |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$APPL_TASK_ITEM |
| Members | ACMS$TASK_GROUP |

# ACMS$APPLICATION_TASK_TSK

Establishes the relationship between ACMS$APPL_TASK_ITEM and ACMS$TASK.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_APP_TSK_TSK |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$APPL_TASK_ITEM |
| Members | ACMS$TASK |

# ACMS$MENU_MDB_FILE

Establishes the relationship between CDD$MENU and the representation of the menu definition binary (MDB) file.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_MENU_MDB |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$MENU |
| Members | CDD$SOURCE_MODULE |

# ACMS$MENU_TASK

Establishes the relationship between CDD$MENU and ACMS$MENU_TASK_ITEM.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_MENU_TASK |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$MENU |
| Members | ACMS$MENU_TASK_ITEM |

# ACMS$MENU_TASK_APPL

Establishes the relationship between ACMS$MENU_TASK_ITEM and ACMS$APPLICATION.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_MENU_TSK_APP |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$MENU_TASK_ITEM |
| Members | ACMS$APPLICATION |

# ACMS$MENU_TASK_TASK

Establishes the relationship between ACMS$MENU_TASK_ITEM and ACMS$APPL_TASK_ITEM.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_MENU_TSK_TSK |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$MENU_TASK_ITEM |
| Members | ACMS$APPL_TASK_ITEM |

# ACMS$PROCEDURE_DATA_AGGREGATE

Establishes the relationship between ACMS$TASK_PROCEDURE_ITEM and CDD$DATA_AGGREGATE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_PRC_DA |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_PROCEDURE_ITEM |
| Members | CDD$DATA_AGGREGATE |

# ACMS$PROCEDURE_ENTRY_PT

Establishes the relationship between ACMS$PROCEDURE and CDD$EXECUTABLE_IMAGE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_PRC_ENT |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$PROCEDURE |
| Members | CDD$PROCEDURE<br>CDD$EXECUTABLE_IMAGE |

# ACMS$PROCEDURE_PROCEDURE

Establishes the relationship between ACMS$TASK_PROCEDURE_ITEM and ACMS$PROCEDURE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_PRC_PROCED |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_PROCEDURE_ITEM |
| Members | ACMS$PROCEDURE |

# ACMS$PROCEDURE_SERVER

Establishes the relationship between ACMS$TASK_PROCEDURE_ITEM and ACMS$SERVER.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_PRC_SRV |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_PROCEDURE_ITEM |
| Members | ACMS$SERVER |

# ACMS$SERVER_ABORT_PROCEDURE

Establishes the relationship between ACMS$SERVER and ACMS$PROCEDURE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_SERVER_ABORT |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$SERVER |
| Members | ACMS$PROCEDURE |

# ACMS$SERVER_ACTION_PROCEDURE

Establishes the relationship between ACMS$SERVER and ACMS$PROCEDURE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_SERVER_ACT |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$SERVER |
| Members | ACMS$PROCEDURE |

## ACMS$SERVER_BASED_ON

Establishes the relationship between ACMS$SERVER and ACMS$SERVER.

### Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_SERVER_BO |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>ACMS$SERVER |
| Members | ACMS$SERVER |

# ACMS$SERVER_END_PROCEDURE

Establishes the relationship between ACMS$SERVER and ACMS$PROCEDURE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_SERVER_END |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$SERVER |
| Members | ACMS$PROCEDURE |

# ACMS$SERVER_INIT_PROCEDURE

Establishes the relationship between ACMS$SERVER and ACMS$PROCEDURE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_SERVER_INIT |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$SERVER |
| Members | ACMS$PROCEDURE |

# ACMS$SERVER_MODULE

Establishes the relationship between ACMS$SERVER and other ACMS application pieces.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_SERVER_MOD |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$SERVER |
| Members | CDD$COMPILED_MODULE |

# ACMS$TASK_BASED_ON

Establishes the relationship between ACMS$TASK and ACMS$TASK.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TK_BASED_ON |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK |
| Members | ACMS$TASK |

# ACMS$TASK_DATA_AGG

Establishes the relationship between ACMS$TASK and CDD$DATA_AGGREGATE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TASK_DA |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK |
| Members | CDD$DATA_AGGREGATE |

# ACMS$TASK_GROUP_BASED_ON

Establishes the relationship between ACMS$TASK_GROUP and ACMS$TASK_GROUP.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TK_GRP_BO |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_GROUP |
| Members | ACMS$TASK_GROUP |

# ACMS$TASK_GROUP_DATA_AGG

Establishes the relationship between ACMS$TASK_GROUP and CDD$DATA_AGGREGATE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TK_GRP_DA |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_GROUP |
| Members | CDD$DATA_AGGREGATE |

# ACMS$TASK_GROUP_MSG_FILE

Establishes the relationship between ACMS$TASK_GROUP and CDD$SOURCE_MODULE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TK_GRP_MSG |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_GROUP |
| Members | CDD$SOURCE_MODULE |

# ACMS$TASK_GROUP_RLB_FILE

Establishes the relationship between ACMS$TASK_GROUP and the forms request library binary (RLB) file.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TK_GRP_RLB |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_GROUP |
| Members | CDD$SOURCE_MODULE |

# ACMS$TASK_GROUP_SERVER

Establishes the relationship between ACMS$TASK_GROUP and ACMS$SERVER.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TK_GRP_SRV |
| Defined Properties | |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_GROUP |
| Members | ACMS$SERVER |

# ACMS$TASK_GROUP_TASK

Establishes the relationship between ACMS$TASK_GROUP and ACMS$TASK.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TK_GRP_TK |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_GROUP |
| Members | ACMS$TASK |

# ACMS$TASK_GROUP_TDB_FILE

Establishes the relationship between ACMS$TASK_GROUP and the task group definition binary (TDB) file.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TK_GRP_TDB |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_GROUP |
| Members | CDD$SOURCE_MODULE |

# ACMS$TASK_GROUP_VIDEO_DISPLAY

Establishes the relationship between ACMS$TASK_GROUP and CDD$VIDEO_DISPLAY.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TK_GRP_VD |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_GROUP |
| Members | None |

# ACMS$TASK_ITEM_DATA_AGGREGATE

Establishes the relationship between ACMS$TASK_TASK_ITEM and CDD$DATA_AGGREGATE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TSK_I_DA |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_TASK_ITEM |
| Members | CDD$DATA_AGGREGATE |

# ACMS$TASK_ITEM_TASK

Establishes the relationship between ACMS$TASK_TASK_ITEM and ACMS$TASK.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TSK_I_TSK |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK_TASK_ITEM |
| Members | ACMS$TASK |

# ACMS$TASK_PROCEDURE

Establishes the relationship between ACMS$TASK and ACMS$TASK_PROCEDURE_ITEM.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TASK_PROC |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK |
| Members | ACMS$TASK_PROCEDURE_ITEM |

# ACMS$TASK_TASK

Establishes the relationship between ACMS$TASK and ACMS$TASK_TASK_ITEM.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TASK_TASK |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK |
| Members | ACMS$TASK_TASK_ITEM |

# ACMS$TASK_VIDEO_DISPLAY

Establishes the relationship between ACMS$TASK and CDD$VIDEO_DISPLAY.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ACMS_TASK_VD |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>ACMS$TASK |
| Members | CDD$VIDEO_DISPLAY |

# CDD$4GL_GROUP_REL

Describes a grouping of composite part relationships for a set of types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_4GL_GROUP |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | None |

## Additional Information

This relation type should not be instantiated.

# CDD$4GL_REL

Groups together a set of common relationships for a set of types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_4GL |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | None |

## Additional Information

This relation type should not be instantiated.

# CDD$ASSOCIATED_KEYWORD

Establishes the relationship between any element type in the CDD/Repository and a keyword name, phrase, or acronym.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ASSOC_KWD, |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | All |
| Members | CDD$KEYWORD |

## Additional Information

Instances of CDD$ASSOCIATED_KEYWORD are usually defined by repository users (rather than created by client products) and serve a user-defined keyword classification concept.

Note that this relation type is used to associate keywords with all other element types and is not repeated throughout the document.

# CDD$ATT_VALIDATION

Establishes the relationship between a property and a method for its validation.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_ATT_VAL |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | None |

# CDD$COMPILED_DEPENDS_ON

Relates all element types that were used to create the CDD$COMPILED_MODULE element type that owns the relationship.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_CMP_DEPENDS_ON |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$COMPILED_MODULE |
| Members | CDD$COMPILED_MODULE<br>CDD$DATA_AGGREGATE<br>CDD$DATA_ELEMENT<br>CDD$DATABASE<br>CDD$MENU<br>CDD$REPORT<br>CDD$SOURCE_MODULE<br>CDD$VIDEO_DISPLAY |

# CDD$COMPILED_DERIVED_FROM

Describes the source module that was compiled to create the compiled module described by a CDD$COMPILED_MODULE element type.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_CMP_DERIVED_FROM |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$COMPILED_MODULE |
| Members | CDD$SOURCE_MODULE |

# CDD$CONSTRAINT_EXPRESSION

Relates a specific CDD$CONSTRAINT element type to a specific CDD$DATA_VALUE element type.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_CONSTRAINT_EXP |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$CONSTRAINT |
| Members | CDD$DATA_VALUE |

# CDD$CONTAINS_COPY

Implements external dependencies for other repositories.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_CONTAINED |
| Defined Properties | CDD$EXTERNAL_REF |
| Notification Dependency | Block |
| Owners | All |
| Members | All |

## Additional Information

This relationship is created by the CDD/Repository when it creates relationships or objects that span physical repositories in distribution. It is not useful for non-system usage. Note that this relation type is also used to associate keywords with all other element types.

# CDD$DATABASE_FILE

Links a database storage representation to the physical file or files in which it is stored.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DATABASE_FILE |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATABASE |
| Members | CDD$SOURCE_MODULE |

# CDD$DATABASE_GROUP_REL

Groups together a set of common relationships for CDD$DATABASE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DB_GROUP |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | None |

## Additional Information

This relationship type should not be instantiated.

# CDD$DATABASE_REL

Groups together a set of common relationships for a set of types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DB |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | None |

# CDD$DATABASE_SCHEMA

Links a DSRI database, an RMS database, or DBMS database to its physical description.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DATABASE_SCHEMA |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATABASE |
| Members | CDD$DATA_AGGREGATE<br>CDD$RDB_DATABASE<br>CDD$RMS_DATABASE<br>DBM$SCHEMA |

## Additional Information

This relation type should not be instantiated.

# CDD$DATA_AGGREGATE_BASED_ON

Lets the owner element type assume the field and record attributes defined in the member element type.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DA_BASED_ON |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_AGGREGATE |
| Members | CDD$DATA_AGGREGATE |

## Additional Information

The CDD$DATA_AGGREGATE_BASED_ON relationship creates a record description using some or all of the attributes of another record description contained elsewhere in the repository. The owner of the relationship assumes the values of the member element type's field attributes, but still contains its own instances of the attributes that apply to all protocols.

If you provide an attribute's value in the owner element type, CDD/Repository uses that value rather than the value for the member element type, so you can override certain attributes in the member element type while still storing commonly used field descriptions in a single place. CDD/Repository automatically overrides the value of each global attribute in the member element type.

Each element type can own only one instance of the CDD$DATA_AGGREGATE_BASED_ON relationship.

The CDD$DATA_ELEMENT_BASED_ON relationship defines a new field based on an existing field. See *CDD/Repository Callable Interface Manual* for more information.

# CDD$DATA_AGGREGATE_COMPUTED_VAL

Establishes a relationship between a data aggregate (record) and a computed value of that aggregate.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DA_COMPUTED |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_AGGREGATE |
| Members | CDD$DATA_VALUE |

# CDD$DATA_AGGREGATE_CONTAINS

Relates field and record attributes defined as members to the aggregation named as owner of this relation.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DA_CONTAINS |
| Defined Properties | CDD$DATA_AGGREGATE_ALIGNMENT<br>CDD$PROTOCOL_TAG<br>CDD$DATA_SEQUENCE_NUMBER |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_AGGREGATE |
| Members | CDD$DATA_AGGREGATE<br>CDD$DATA_ELEMENT<br>CDD$DATA_OVERLAY_AGGREGATE |

# CDD$DATA_ARRAY_HAS_DIMENSION

Links the definition of an array field or repeating record with a set of element types that define its repeating characteristics.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DA_HAS_DIMENSION |
| Defined Properties | CDD$DATA_ARRAY_ORDER |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_AGGREGATE<br>CDD$DATA_ELEMENT |
| Members | CDD$DATA_DIMENSION |

## Additional Information

The CDD$DATA_ARRAY_HAS_DIMENSION relationship links a record or field to one or more CDD$DATA_DIMENSION entities.

An array with more than one subscript uses one instance of the CDD$DATA_ARRAY_HAS_DIMENSION relationship for each subscript. For example, a two-dimensional array field has two instances of the CDD$DATA_ARRAY_HAS_DIMENSION relationship. One instance describes the characteristics of each row of the array, the other describes the characteristics of each column. For more information, refer to the CDD$DATA_DIMENSION and CDD$DATA_DIM*_REL relationships.

# CDD$DATA_DESC_GROUP_REL

Provides a supertype for a set of relation types that provide detailed data description characteristcis.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DATA_DESC_GROUP |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | None |

## Additional Information

This relation type should not be instantiated.

# CDD$DATA_DIMENSION_HIGH_BD_REL

Links a dimension element type with a data value describing the upper bound of the dimension's subscript.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DD_HIGH_BOUND |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_DIMENSION |
| Members | CDD$DATA_VALUE |

## Additional Information

The CDD$DATA_DIMENSION_HIGH_BD_REL relation links a CDD$DATA_DIMENSION element type to a complex expression describing the subscript. The member of the CDD$DATA_DIMENSION_HIGH_BD_REL relationship is a data value element type that describes the characteristics of the upper bound.

If the array subscript's upper bound is a simple constant, use the CDD$DATA_DIMENSION_HIGH_BOUND property.

A single CDD$DATA_DIMENSION element type cannot own instances of both the CDD$DATA_DIMENSION_HIGH_BD_REL relationship and the CDD$DATA_DIMENSION_HIGH_BOUND property.

# CDD$DATA_DIMENSION_INDEX

Links a dimension element type with a description of a program variable.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DD_INDEX |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_DIMENSION |
| Members | CDD$DATA_VALUE |

## Additional Information

In a VAX COBOL program, you can declare a program variable as an index into an array structure. The CDD$DATA_DIMENSION_INDEX relationship describes such a program variable in the CDD/Repository.

# CDD$DATA_DIMENSION_LOW_BD_REL

Links a dimension element type to a data value describing the lower bound of the dimension's subscript.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DD_LOW_BOUND |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_DIMENSION |
| Members | CDD$DATA_VALUE |

## Additional Information

The CDD$DATA_DIMENSION_LOW_BD_REL relationship links a CDD$DATA_DIMENSION element type to a CDD$DATA_VALUE element type that defines a complex expression for the subscript value.

If the array subscript's lower bound is a simple constant, use the CDD$DATA_DIMENSION_LOW_BOUND property. A single CDD$DATA_DIMENSION element type cannot own instances of both the CDD$DATA_DIMENSION_LOW_BD_REL relationship and of the CDD$DATA_DIMENSION_LOW_BOUND property.

# CDD$DATA_DIM_HIGH_BOUND_HI_REL

Links a CDD$DATA_DIMENSION element type with a CDD$DATA_VALUE element type describing the maximum legal value of a run-time expression controlling the upper bound of a data dimension.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DD_HB_HIGH_VALUE |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DIMENSION |
| Members | CDD$DATA_VALUE |

## Additional Information

The CDD$DATA_DIM_HIGH_BOUND_HI_REL relationship specifies the maximum size of an array whose bounds are declared at run time.

You use the CDD$DATA_DIM_HIGH_BOUND_HI_REL relationship to describe values other than simple constants. For simple constants, you use the CDD$DATA_DIM_HIGH_BOUND_HI_VAL property.

# CDD$DATA_DIM_HIGH_BOUND_LOW_REL

Links a CDD$DATA_DIMENSION element type with a CDD$DATA_VALUE element type describing the minimum legal value of a run-time variable controlling the upper bound of a data dimension.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DD_HB_LOW_VALUE |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_DIMENSION |
| Members | CDD$DATA_VALUE |

## Additional Information

The CDD$DATA_DIM_HIGH_BOUND_LOW_REL relationship defines the minimum size of an array whose bounds are declared at run time.

The CDD$DATA_DIM_HIGH_BOUND_LOW_REL relationship describes values other than simple constants. If the value is a simple constant, use the CDD$DATA_DIM_HIGH_BOUND_LOW_VAL property to describe it.

# CDD$DATA_ELEMENT_BASED_ON

Lets the owner element assume the field attributes defined in the member element.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DE_BASED_ON |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_ELEMENT |
| Members | CDD$DATA_ELEMENT |

## Additional Information

With the CDD$DATA_ELEMENT_BASED_ON relationship, you can create a field description using some or all of the attributes of another field description contained elsewhere in the CDD/Repository. The owner of the relationship usually assumes the values of the attributes as they exist for the member element, but if you provide an attribute's value in the owner element, CDD/Repository uses that value instead.

The owner element assumes only the values of the member element's field attributes; the owner element contains its own instances of those attributes that apply to all protocols. Each data element can own no more than one instance of the CDD$DATA_ELEMENT_BASED_ON relationship.

To base a new record on record definition, use the CDD$DATA_AGGREGATE_BASED_ON relationship.

# CDD$DATA_ELEMENT_COMPUTED_VALUE

Links the description of a field to the calling programs used to compute the value of the field.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DE_COMPUTED |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_ELEMENT |
| Members | CDD$DATA_VALUE |

## Additional Information

The calling program calculates the value of a computed field using values of other fields in the record. You use the CDD$DATA_ELEMENT_COMPUTED_VALUE relationship to describe a field whose value is calculated at run time.

# CDD$DATA_ELEMENT_INITIAL_DEF

Links the owner element to multiple instances of data value elements that collectively express components of an expression used to calculate an initial value for a data element.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DE_INITIAL |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_ELEMENT |
| Members | CDD$DATA_VALUE |

## Additional Information

With the CDD$DATA_ELEMENT_INITIAL_DEF relationship, you can specify complex expressions for the first value of a field in the database. Such expressions can perform complex calculations or rely on the values of other fields.

To specify a simple constant or literal value as the first value, use the CDD$DATA_ELEMENT_INITIAL_VALUE property. The same data element cannot own both the CDD$DATA_ELEMENT_INITIAL_VALUE property and the CDD$DATA_ELEMENT_INITIAL_DEF relationship.

# CDD$DATA_ELEMENT_INPUT_VALID

Links a CDD$DATA_ELEMENT element type to a CDD$DATA_VALUE element type, where the CDD$DATA_VALUE element type provides validation constraints for the CDD$DATA_ELEMENT element type.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DE_IN_VALID |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_ELEMENT |
| Members | CDD$DATA_VALUE |

## Additional Information

CDD$DATA_VALUE instances may be expressions identifying constraints, tables of values, constants, or derivations.

# CDD$DATA_ELEMENT_MISSING_DEF

Links the owner element to a data value element that expresses a missing value for the field.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DE_MISSING |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_ELEMENT |
| Members | CDD$DATA_VALUE |

## Additional Information

With the CDD$DATA_ELEMENT_MISSING_DEF relationship, you can specify complex expressions as the missing value of the field being defined in your program. Such expressions can involve complex calculations or use values from other fields.

If the missing value is a simple constant or a literal value, use the CDD$DATA_ELEMENT_MISSING_VALUE property instead. The same data
element cannot own both the CDD$DATA_ELEMENT_MISSING_VALUE property and the CDD$DATA_ELEMENT_MISSING_DEF relationship.

# CDD$DATA_ELEMENT_POINTER_REF

Links an element type with an address, which is stored as CDD$DATA_VALUE.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DE_POINTER |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_ELEMENT |
| Members | CDD$DATA_VALUE |

## Additional Information

The CDD$DATA_ELEMENT element type related by this relationship is a component of a larger data structure (database). The role of the element type is to name a field providing linkage to a physical file containing instances of the data structure. The element type is a pointer data type and the CDD$DATA_VALUE element type represents the instance of that pointer.

# CDD$DATA_ELE_VALS

Provides a supertype for element types that represent special purpose data description characteristics such as computed values and reference pointers.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DATA_ELE_VALS |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | None |

## Additional Information

This relation type should not be instantiated.

# CDD$DATA_INSTANCE_PATH

Includes a repository element in its appropriate position within a data instance path.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DI_PATH |
| Defined Properties | CDD$DATA_INSTANCE_PATH_STEP |
| Notification Dependency | Block |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_INSTANCE |
| Members | CDD$DATA_AGGREGATE<br>CDD$DATA_OVERLAY<br>CDD$DATA_OVERLAY_AGGREGATE<br>CDD$DATABASE<br>CDD$RDB_DATABASE<br>CDD$RMS_DATABASE<br>DTR$DOMAIN |

## Additional Information

A data instance is made up of:

- A data element or data aggregate that functions as the data instance root
- A path of repository elements that leads to the data instance root

You use the CDD$DATA_INSTANCE_PATH relationship to include an element in the path that leads to the data instance root.

A data instance element type that owns the CDD$DATA_INSTANCE_PATH relationship must also own the CDD$DATA_INSTANCE_ROOT relationship.

A data instance element type can own any number of CDD$DATA_INSTANCE_PATH relationships. However, each of these relationships must have a unique value of the CDD$DATA_INSTANCE_PATH_STEP property.

# CDD$DATA_INSTANCE_ROOT

Identifies an element type upon which a data value expression depends.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DI_ROOT |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_INSTANCE |
| Members | CDD$DATA_AGGREGATE<br>CDD$DATA_ELEMENT |

## Additional Information

When you use an expression that refers to a field, you must use an instance of the CDD$DATA_VALUE_DEPENDS_ON relationship to identify the field. The relationship does not own the field description, however. Instead, it owns a data instance element type that points to the field description, using the CDD$DATA_INSTANCE_ROOT relationship.

Each CDD$DATA_INSTANCE element type must own one instance of the CDD$DATA_INSTANCE_ROOT relationship.

# CDD$DATA_OVERLAY_AGG_CONTAINS

Links a named declaration of an aggregate overlay with a set of members that comprises the aggregate.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DOA_CONTAINS |
| Defined Properties | CDD$SEQUENCE_NUMBER |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_OVERLAY_AGGREGATE |
| Members | CDD$DATA_OVERLAY |

## Additional Information

The CDD$DATA_OVERLAY_AGG_CONTAINS relationship links an instance of the CDD$DATA_OVERLAY_AGGREGATE element type with each CDD$DATA_OVERLAY element type that the aggregate uses.

Within a record description, you use the CDD$DATA_OVERLAY_AGGREGATE element type to provide alternate sets of field descriptions (overlays) for a part of the record. In describing a set of overlays, the CDD$DATA_OVERLAY_AGGREGATE element type comprises a number of CDD$DATA_OVERLAY entities, each describing one possible interpretation of a part of a record. At run time, your program chooses which overlay to use when interpreting a record.

# CDD$DATA_OVERLAY_CONTAINS

Links an overlay with the individual fields and records that describe it.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DO_CONTAINS |
| Defined Properties | CDD$DATA_AGGREGATE_ALIGNMENT<br>CDD$DATA_SEQUENCE_NUMBER |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_OVERLAY |
| Members | CDD$DATA_AGGREGATE<br>CDD$DATA_ELEMENT<br>CDD$DATA_OVERLAY_AGGREGATE |

## Additional Information

Instances of the CDD$DATA_OVERLAY_CONTAINS relationship own the CDD$DATA_SEQUENCE_NUMBER property to establish a sequential order among the various fields or records in a CDD$DATA_OVERLAY element type.

If a single CDD$DATA_OVERLAY element type owns several instances of the CDD$DATA_OVERLAY_CONTAINS relationship, each instance must have a unique value of the CDD$DATA_SEQUENCE_NUMBER property.

# CDD$DATA_OVERLAY_IDENTIFICATION

Links an overlay with a data value that determines which of several variant structures the program will choose at run time.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DO_ID |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_OVERLAY |
| Members | CDD$DATA_VALUE |

## Additional Information

When you define a variant structure in the CDD/Repository, the CDD$DATA_OVERLAY_IDENTIFICATION relationship links each overlay with a data value.

At run time, your program compares this data value with a special overlay control field. If the values match, your program can use this overlay structure to interpret the data. If the values do not match, your program chooses another structure.

If one variant in a variant structure owns an instance of the CDD$DATA_OVERLAY_IDENTIFICATION relationship, all variants in that variant structure must own an instance of it.

# CDD$DATA_VALUE_DEPENDS_ON

Links a data value to a fully qualified field expression, as represented by a data instance.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DV_DEPENDS_ON |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_VALUE |
| Members | CDD$DATA_INSTANCE |

## Additional Information

A data value element type can include a CDD$DATA_VALUE_EXPRESSION property, a CDD$DATA_VALUE_DEPENDS_ON relationship, or both. The relationship informs the CDD$DATA_VALUE element type of changes to fields referred to in the expression. For each field you refer to in the value expression, you must establish a link between the data value element type and the field in the value expression by using the CDD$DATA_VALUE_DEPENDS_ON relationship.

# CDD$DBM_REL

Provides a supertype for a set of relation types that attach values to schema parameters.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DMB_REL |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS |
| Members | None |

## Additional Information

This relation type should not be instantiated.

# CDD$DISPLAY_ELEMENTS

Connects the video display to a data value.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DISP_ELMNTS |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$VIDEO_DISPLAY |
| Members | CDD$DATA_VALUE |

# CDD$FILE_ACCESS_INDEX

Links an index to use with an RMS access mechanism.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_FILE_ACCESS_INDEX |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$FILE_ACCESS |
| Members | CDD$FILE_INDEX |

# CDD$FILE_AREA_ALLOC

Links the RMS allocation areas to be used with a file or index.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_FILE_AREA_ALLOC |
| Defined Properties | CDD$RMS_XABKEY_AREA<br>CDD$DATA_SEQUENCE_NUMBER |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$FILE_DEFINITION<br>CDD$FILE_INDEX |
| Members | CDD$FILE_ALLOCATION |

# CDD$FILE_INDEXED_BY

Links the indexes to be used with an RMS file.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_FILE_INDEXED_BY |
| Defined Properties | CDD$RMS_XABKEY_REF |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$FILE_DEFINITION |
| Members | CDD$FILE_INDEX |

# CDD$FILE_INDEX_SEG

Provides linkage between a component of an RMS file index and the value held by that segment.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_FILE_INDEX_SEG |
| Defined Properties | CDD$RMS_XABKEY_SEQ |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$FILE_INDEX |
| Members | CDD$DATA_VALUE |

# CDD$FILE_REL

Provides a supertype for a set of relation types that describe an RMS file.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_FILE_REL |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | None |

## Additional Information

This relation type should not be instantiated.

# CDD$HAS_LINK

Provides the relationship between the links which a particular relationship may have.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_HAS_LINK |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | CDD$LINK_TYPE |

## Additional Information

This relation type is intended for internal use only.

# CDD$IMAGE_DERIVED_FROM

Relates each of the compiled modules that were linked together to form an image.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_EXE_DERIVED_FROM |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$EXECUTABLE_IMAGE |
| Members | CDD$COMPILED_MODULE |

# CDD$INDEX_SEGMENT

Links an index defined for a DSRI compliant database to the segments of that index.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_INDEX_SEGMENT |
| Defined Properties | CDD$DATA_SEQUENCE_NUMBER<br>CDD$RDB_DESC_INDEX_SEQ |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$INDEX |
| Members | CDD$DATA_VALUE |

# CDD$IN_FILE

Links an element type to its physical location on a disk as described by a CDD$FILE element type.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_IN_FILE |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$COMPILED_MODULE<br>CDD$EXECUTABLE_IMAGE<br>CDD$PROCEDURE<br>CDD$PROGRAMS<br>CDD$SOURCE_MODULE<br>CDD$VIDEO_DISPLAY<br>GEN$PROGRAM<br>RALLY$APPLICATION |
| Members | CDD$SOURCE_MODULE |

# CDD$KEYWORD_GENERALIZATION

Recognizes a user-defined hierarchy of keywords.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_KWD_GEN |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$KEYWORD |
| Members | CDD$KEYWORD |

## Additional Information

An example for using this type of relationship would be a user-defined keyword scheme in which all (instances of type) element types must be classified as members of a limited set of conceptual data types, such as quantities, tables, or dates. The general class of quantities could be decomposed into large quantities, monetary quantities, and small quantities. So by assigning an instance to the keyword classification small quantities, it is implicitly assigned to the classification quantities as well.

Keyword schemes are designed to aid retreival by creating a subset of instances from the complete inventory of object instances in the repository.

# CDD$MENU_CONTAINS

Represents the relationship between a menu and the items on the menu.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_MENU_CONTAINS |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$MENU |
| Members | CDD$MENU |

## Additional Information

This relation type is not currently used.

# CDD$PREFERRED_TERM

Recognizes a user-defined set of keywords.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_PREFERRED_TERM |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$KEYWORD |
| Members | CDD$KEYWORD |

## Additional Information

Both owners and members of this relation type are instances of the CDD$KEYWORD element type.

This facility may be used as part of a user-defined data standardization program where a subset of keywords are determined to be the preferred terminology.

# CDD$RDB_COL_SEQ

Establishes the linkage between an Rdb/VMS database and the collating sequence charactertistic for that database.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RDB_COL_SEQ |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$RDB_DATABASE |
| Members | CDD$COLLATING_SEQUENCE |

## Additional Information

The CDD$RDB_COL_SEQ relation type is used to sort data. The collating sequence specifies the language to be used for collating.

# CDD$RDB_CONSTRAINT

Links a database to the criteria for maintaining the validity of the database.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RDB_CONSTRAINT |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$RDB_DATABASE |
| Members | CDD$CONSTRAINT |

# CDD$RDB_DATABASE_DERIVED_FROM

Links a DSRI compliant database to an interrupted dispatch block (IDB) database and maps DSRI concepts to block IDB concepts.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RDB_DERIVED_FROM |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$RDB_DATABASE |
| Members | IDB$DATABASE |

# CDD$RDB_DATA_AGGREGATE

Links a database description for a DSRI compliant database to a data aggregate that describes the format for a table in the database.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RDB_DA |
| Defined Properties | CDD$DATABASE_KEY_LENGTH |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$RDB_DATABASE |
| Members | CDD$DATA_AGGREGATE |

# CDD$RDB_DATA_ELEMENT

Links a data element to a database in which it may be used.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DATA_ELEMENT |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$RDB_DATABASE |
| Members | CDD$DATA_ELEMENT |

# CDD$RDB_DATA_REL

Provides a supertype for the primary data structure components of an Rdb/VMS database.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RDB_DATA |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY |
| Members | None |

## Additional Information

This relation type should not be instantiated.

# CDD$RDB_DA_DERIVED_FROM

Links a record in an IDB database description to a data aggregate that describes its structure.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RDB_DA_DRVD_FROM |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATA_AGGREGATE |
| Members | CDD$DATA_AGGREGATE |

# CDD$RDB_INDEX

Associates a named index with the Rdb/VMS database being indexed.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RDB_INDEX |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$RDB_DATABASE |
| Members | CDD$INDEX |

# CDD$REPORT_SOURCE

Represents the source of a report.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RPT_SRC |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$REPORT |
| Members | CDD$DATA_VALUE |

## Additional Information

This relation type should not be instantiated.

# CDD$RMS_DATA_AGGREGATE

Links an RMS database description to a data aggregate that describes the format of each record type in the file.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RMS_DATA_AGG |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$RMS_DATABASE |
| Members | CDD$DATA_AGGREGATE |

# CDD$RMS_FILE_DEFINITION

Links an RMS database description to the physical file name and location within the file management system.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RMS_FILE_DEF |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$RMS_DATABASE |
| Members | CDD$FILE_DEFINITION |

# CDD$SOURCE_DEPENDS_ON

Describes elements that a CDD$SOURCE_MODULE element type uses to build it.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_SRC_DEPENDS_ON |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$SOURCE_MODULE |
| Members | CDD$DATA_AGGREGATE<br>CDD$DATA_ELEMENT<br>CDD$DATABASE<br>CDD$MENU<br>CDD$REPORT<br>CDD$SOURCE_MODULE<br>CDD$VIDEO_DISPLAY |

# CDD$SOURCE_DERIVED_FROM

Describes elements from which a CDD$SOURCE_MODULE element type was created. For example, it might connect preprocessor output to the source module.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_SRC_DERIVED_FROM |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>CDD$SOURCE_MODULE |
| Members | CDD$SOURCE_MODULE<br>GEN$PROGRAM |

# CDD$VIDEO_DISPLAY_BASED_ON

Associates one screen format with a second from which some of its characteristics have been defined.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_VD_BASED_ON |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$VIDEO_DISPLAY |
| Members | CDD$VIDEO_DISPLAY |

## Additional Information

This relation type should not be instantiated.

# DBM$AREAS

Provides the link between database schema element types and physical database storage file element types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_AREA |
| Defined Properties | DBM$ID |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DBM$REALM<br>DBM$SCHEMA |
| Members | DBM$AREA |

# DBM$DATA_AGGREGATES

Provides the link between database subschema element types and each segment where segments are represented by CDD$DATA_AGGREGATE element types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_RECORD |
| Defined Properties | DBM$ID |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DBM$SUBSCHEMA |
| Members | CDD$DATA_AGGREGATE |

# DBM$REALMS

Provides the link between database subschema element types and related realm element types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_REALMS |
| Defined Properties | DBM$ID |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$AGGREGATE<br>DBM$SUBSCHEMA |
| Members | DBM$REALM |

# DBM$SECURITY_SCHEMAS

Provides the link between database schema element types and database security schema element types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_SEC_SCHEMA |
| Defined Properties | None |
| Notification Dependency | Block |
| Owners | CDD$CONTAINS_COPY<br>DBM$SCHEMA |
| Members | DBM$SECURITY_SCHEMA |

# DBM$SEC_SCHEMA_INST

Provides the link between the CDD$DATABASE and the security schema element types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_SEC_SI_MEM |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATABASE |
| Members | DBM$SECURITY_SCHEMA |

# DBM$SETS

Provides the link between database subschema element types and related set element types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_SETS |
| Defined Properties | DBM$ID |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DBM$SUBSCHEMA |
| Members | DBM$SET |

# DBM$SET_MEMBERS

Provides the link between database subschema set element types and related member CDD$DATA_AGGREGATE element types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_SET_MEMBER |
| Defined Properties | DBM$ORDER<br>DBM$INSERTION<br>DBM$RETENTION |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DBM$SET |
| Members | CDD$DATA_AGGREGATE |

# DBM$SET_OWNERS

Provides the link between database subschema set element types and the related owner CDD$DATA_AGGREGATE element type.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_SET_OWNER |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DBM$SET |
| Members | CDD$DATA_AGGREGATE |

# DBM$STORAGE_SCHEMAS

Provides the link between database schema element types and database security schema element types.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_STO_SCHEMA |
| Defined Properties | None |
| Notification Dependency | Block |
| Owners | CDD$CONTAINS_COPY<br>DBM$SCHEMA |
| Members | DBM$STORAGE_SCHEMA |

# DBM$STO_SCHEMA_INST

Provides the relationship between the CDD$DATABASE element type and the storage schema it is using.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_STO_SI |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATABASE |
| Members | DBM$STORAGE_SCHEMA |

# DBM$SUBSCHEMAS

Provides the link between database schema element type and database subschema element type.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_SUBSCHEMA |
| Defined Properties | None |
| Notification Dependency | Block |
| Owners | CDD$CONTAINS_COPY<br>DBM$SCHEMA |
| Members | DBM$SUBSCHEMA |

# DBM$SUBSCHEMAS_INST

Provides the relationship between the CDD$DATABASE element type and the subschemas it is using.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DBM_SUB_SI |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>CDD$DATABASE |
| Members | DBM$SUBSCHEMA |

# DTR$DATABASE_SCHEMA

Links a VAX Datatrieve definition of a DBMS database to the database schema.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DTR_DB_SCHEMA |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DTR$DATABASE |
| Members | DBM$SCHEMA |

# DTR$DATABASE_SUBSCHEMA

Links a VAX Datatrieve definition of a DBMS database to the subschema being accessed from Datatrieve.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DTR_DB_SUBSCHEMA |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DTR$DATABASE |
| Members | DBM$SUBSCHEMA |

# DTR$DOMAIN_SOURCE

Links a VAX Datatrieve domain to a CDD$DATABASE element type on which the domain is based.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DTR_DOMAIN_SOURCE |
| Defined Properties | None |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>DTR$DOMAIN |
| Members | CDD$DATA_AGGREGATE<br>CDD$DATA_VALUE<br>CDD$DATABASE<br>DTR$DOMAIN |

# DTR$PROCEDURE_CDD_DB

Links a VAX Datatrieve procedure to a CDD$DATABASE element type that is accessed by the procedure itself.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DTR_PROC_CDD_DB |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DTR$PROCEDURE |
| Members | DTR$DATABASE |

# DTR$PROCEDURE_DTR_DB

Links a VAX Datatrieve procedure to a DTR$DATABASE element type that is accessed by the procedure itself.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DTR_PROC_DTR_DB |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DTR$PROCEDURE |
| Members | DTR$DATABASE |

# DTR$PROCEDURE_FIELD

Links a VAX Datatrieve procedure to a Common Dictionary Operator (CDO) field used to declare variables within the procedure.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DTR_PROC_FIELD |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DTR$PROCEDURE |
| Members | CDD$DATA_ELEMENT |

# DTR$PROCEDURE_SOURCE

Links a VAX Datatrieve procedure to a domain that is accessed by the procedure itself.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DTR_PROC_SOURCE |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DTR$PROCEDURE |
| Members | CDD$DATABASE<br>DTR$DATABASE<br>DTR$DOMAIN |

# DTR$PROCEDURE_TABLE

Links a VAX Datatrieve procedure to a table used within the procedure itself.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DTR_PROC_TABLE |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DTR$PROCEDURE |
| Members | DTR$TABLE |

# DTR$TABLE_SOURCE

Links a VAX Datatrieve domain table where the values of the table are stored.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_DTR_TABLE_SOURCE |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>DTR$TABLE |
| Members | DTR$DOMAIN |

# RALLY$APPLICATION_CONTAINS

Provides the relationship between the RALLY$APPLICATION element type and the RALLY$DATA_SOURCE_DEFINITION element type.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RALLY_APP_CONT |
| Defined Properties | RALLY$AFILE_POINTER |
| Notification Dependency | Signal |
| Owners | CDD$CONTAINS_COPY<br>RALLY$APPLICATION |
| Members | CDD$MENU<br>CDD$REPORT<br>CDD$VIDEO_DISPLAY<br>RALLY$DATA_SOURCE_DEFINITION<br>RALLY$PACKET<br>RALLY$PROCEDURE<br>RALLY$TASK |

# RALLY$DSD_SOURCE

Provides the RALLY$DSD_SOURCE relationship between the RALLY$DATA_SOURCE_DEFINITION element type and its underlying metadata source.

## Summary

| | |
|---|---|
| Tag | NAD$K_REL_RALLY_DSD |
| Defined Properties | None |
| Notification Dependency | Success |
| Owners | CDD$CONTAINS_COPY<br>RALLY$DATA_SOURCE_DEFINITION |
| Members | CDD$DATA_VALUE<br>CDD$DATABASE |

# 3

# Property Descriptions

This chapter contains descriptions of all the CDD/Repository properties, arranged alphabetically. Each property description contains the following information in the Summary section:

**Title**—Includes the generic name of the property, a short phrase describing the property.

**Tag**—For normal properties, gives the symbolic constant for the value of the **tag** property for this property definition.

**Defined By**—Gives the name of the element type that uses the property. Subtypes of that element type inherit the property's definition.

If more than one element type is listed, the element types are on separate branches of the type hierarchy.

**Required With new**—Specifies whether you must specify an initial value for this property when you send the **new** message to create an instance of its defining type or a subtype.

**Type**—Gives the property's implementation type, as follows:

- Normal—The data is stored with the element.
- Relation—The property's value (an element ID or scan of element IDs) is formed by traversing relationships, and modified by adding or removing relationships.
- Closure—Similar to relation, except that the traversal is recursive.
- Computed—The property's value is computed by a method.

**Data Type**—Gives the name of the property's data type.

**Access**—Specifies the allowable access to the property using **getProp** and **setProp**. The values are as follows:

- Read-only—The property may only be read.
- Read/Write—The property may be read and its value modified.

- Write-once—The property may be read and its value may be set once.
- Write-once-at-creation—The property may be read and its value may be set when the element that possesses it is created.

The access described here is for all users and is in addition to the access control provided for elements on a user-by-user basis. Note that the value of a property that cannot be changed with **setProp** may still be modified as the side effect of some other operation.

**Relation Traversed**—Applies only to relation and closure properties. Names the relation that implements the property.

**Traversal Direction**—Applies only to relation and closure properties. Specifies the direction in which the implementing relation is traversed, either To owner or To member. For closure properties, a value of To all owners or To all members indicates the recursive nature of the traversal.

**Inverse Property**—Applies only to relation and closure properties. Names the property that traverses the same relation in the opposite direction.

## ACMS$APPLICATION_FOR_ADB_FILE

Represents the inverse property for ACMS$APPLICATION_HAS_ADB_FILE.

### Summary

| | |
|---|---|
| Defined By | MCS_BINARY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_ADB_FILE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$APPLICATION_HAS_ADB_FILE |

## ACMS$APPLICATION_FOR_SERVER

Represents the inverse property for ACMS$APPLICATION_HAS_SERVER.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION_SERVER_ITEM |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_SERVER |
| Traversal Direction | To owner |
| Inverse Property | ACMS$APPLICATION_HAS_SERVER |

## ACMS$APPLICATION_FOR_TASK

Represents the inverse property for ACMS$APPLICATION_HAS_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION_TASK_ITEM |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_TASK |
| Traversal Direction | To owner |
| Inverse Property | ACMS$APPLICATION_HAS_TASK |

## ACMS$APPLICATION_FOR_TASK_GROUP

Represents the inverse property for ACMS$APPLICATION_HAS_TASK_GROUP.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_TASK_GROUP |
| Traversal Direction | To owner |
| Inverse Property | ACMS$APPLICATION_HAS_TASK_GROUP |

## ACMS$APPLICATION_HAS_ADB_FILE

Represents the inverse property for ACMS$APPLICATION_FOR_ADB_FILE.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_ADB_FILE |
| Traversal Direction | To member |
| Inverse Property | ACMS$APPLICATION_FOR_ADB_FILE |

## ACMS$APPLICATION_HAS_SERVER

Represents the inverse property for ACMS$APPLICATION_FOR_SERVER.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_SERVER |
| Traversal Direction | To owner |
| Inverse Property | ACMS$APPLICATION_FOR_SERVER |

## ACMS$APPLICATION_HAS_TASK

Represents the inverse property for ACMS$APPLICATION_FOR_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_TASK |
| Traversal Direction | To member |
| Inverse Property | ACMS$APPLICATION_FOR_TASK |

## ACMS$APPLICATION_HAS_TASK_GROUP

Represents the inverse property for ACMS$APPLICATION_FOR_TASK_GROUP.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_TASK_GROUP |
| Traversal Direction | To member |
| Inverse Property | ACMS$APPLICATION_FOR_TASK_GROUP |

## ACMS$APPLICATION_SRV_FOR_GRP

Represents the inverse property for ACMS$APPLICATION_SRV_HAS_GRP.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_SRV_GRP |
| Traversal Direction | To owner |
| Inverse Property | ACMS$APPLICATION_SRV_HAS_GRP |

## ACMS$APPLICATION_SRV_FOR_SRV

Represents the inverse property for ACMS$APPLICATION_SRV_HAS_SRV.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_SRV_SRV |
| Traversal Direction | To owner |
| Inverse Property | ACMS$APPLICATION_SRV_HAS_SRV |

## ACMS$APPLICATION_SRV_HAS_GRP

Represents the inverse property for ACMS$APPLICATION_SRV_FOR_GRP.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION_SERVER_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_SRV_GRP |
| Traversal Direction | To member |
| Inverse Property | ACMS$APPLICATION_SRV_FOR_GRP |

## ACMS$APPLICATION_SRV_HAS_SRV

Represents the inverse property for ACMS$APPLICATION_SRV_FOR_SRV.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION_SERVER_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_SRV_SRV |
| Traversal Direction | To member |
| Inverse Property | ACMS$APPLIATION_SRV_SRV |

## ACMS$APP_TASK_FOR_TASK_GRP

Represents the inverse property for ACMS$APP_TASK_HAS_TASK_GRP.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_TASK_GRP |
| Traversal Direction | To owner |
| Inverse Property | ACMS$APPLIATION_HAS_TASK_GRP |

## ACMS$APP_TASK_HAS_TASK_GRP

Represents the inverse property for ACMS$APP_TASK_FOR_TASK_GRP.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION_TASK_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLIATION_TASK_GRP |
| Traversal Direction | To member |
| Inverse Property | ACMS$APP_TASK_FOR_TASK_GRP |

## ACMS$APP_TASK_HAS_TASK_ITEM

Represents the inverse property for ACMS$APP_TASK_FOR_TASK_ITEM.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION_TASK_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_TASK_TSK |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_ITEM_FOR_APP_TASK |

## ACMS$EXTERNAL_LIST

Represents an ACMS external list. This property can be used as internal buffer in ACMS.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_ACMS_EXTERN_LIST |
| Defined By | ACMS$APPLICATION<br>ACMS$SERVER<br>ACMS$TASK<br>ACMS$TASK_GROUP<br>CDD$MENU |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## ACMS$IDENTIFIER

Represents an ACMS identifier.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_ACMS_IDENTIFIER |
| Defined By | ACMS$APPLICATION<br>ACMS$APPLICATION_SERVER_ITEM<br>ACMS$APPLICATION_TASK_ITEM<br>ACMS$MENU_TASK_ITEM<br>ACMS$PROCEDURE<br>ACMS$SERVER<br>ACMS$TASK<br>ACMS$TASK_ITEM<br>ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## ACMS$INTERNAL_CHARACTERISTICS

Represents ACMS internal characteristics. This property can be used as an internal buffer in ACMS.

### ACMS$INTERNAL_CHARACTERISTICS

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_ACMS_INTERN_CHAR |
| Defined By | ACMS$APPLICATION<br>ACMS$SERVER<br>ACMS$TASK<br>ACMS$TASK_GROUP<br>CDD$MENU |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## ACMS$MENU_FOR_APPL

Represents the inverse property for ACMS$MENU_HAS_APPL.

#### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$MENU_TASK_APPL |
| Traversal Direction | To owner |
| Inverse Property | ACMS$MENU_TASK_HAS_APPL |

## ACMS$MENU_FOR_MDB_FILE

Represents the inverse property for ACMS$MENU_HAS_MDB_FILE.

### Summary

| | |
|---|---|
| Defined By | CDD$MENU<br>MCS_BINARY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$MENU_MDG_FILE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$MENU_HAS_MDB_FILE |

## ACMS$MENU_FOR_TASK

Represents the inverse property for ACMS$MENU_HAS_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$MENU_TASK_ITEM |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$MENU_TASK |
| Traversal Direction | To owner |
| Inverse Property | ACMS$MENU_HAS_TASK |

## ACMS$MENU_HAS_MDB_FILE

Represents the inverse property for ACMS$MENU_FOR_MDB_FILE.

### Summary

| | |
|---|---|
| Defined By | CDD$MENU |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$MENU_MDB_FILE |
| Traversal Direction | To member |
| Inverse Property | ACMS$MENU_FOR_MDB_FILE |

## ACMS$MENU_HAS_TASK

Represents the inverse property for ACMS$MENU_FOR_TASK.

### Summary

| | |
|---|---|
| Defined By | CDD$MENU |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$MENU_TASK |
| Traversal Direction | To member |
| Inverse Property | ACMS$MENU_FOR_TASK |

## ACMS$MENU_TASK_FOR_APPL

Represents the inverse property for ACMS$MENU_TASK_HAS_APPL.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION_TASK_ITEM |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$MENU_TASK_APPL |
| Traversal Direction | To owner |
| Inverse Property | ACMS$MENU_HAS_TASK_APPL |

## ACMS$MENU_TASK_FOR_TASK

Represents the inverse property for ACMS$MENU_TASK_HAS_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$APPLICATION_TASK_ITEM |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$MENU_TASK |
| Traversal Direction | To owner |
| Inverse Property | ACMS$MENU_TASK_HAS_TASK |

## ACMS$MENU_TASK_HAS_APPL

Represents the inverse property for ACMS$MENU_TASK_FOR_APPL.

### Summary

| | |
|---|---|
| Defined By | ACMS$MENU_TASK_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$MENU_TASK_APPL |
| Traversal Direction | To member |
| Inverse Property | ACMS$MENU_FOR_APPL |

## ACMS$MENU_TASK_HAS_TASK

Represents the inverse property for ACMS$MENU_TASK_FOR_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$MENU_TASK_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$MENU_TASK |
| Traversal Direction | To member |
| Inverse Property | ACMS$MENU_TASK_FOR_TASK |

## ACMS$OBJECT_SIZE

Represents an ACMS object size.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_ACMS_SIZE |
| Defined By | ACMS$TASK |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## ACMS$PATHNAME

Represents an ACMS pathname.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_ACMS_PATHNAME |
| Defined By | ACMS$APPLICATION<br>ACMS$SERVER<br>ACMS$TASK<br>ACMS$TASK_GROUP<br>CDD$MENU |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## ACMS$PROCEDURE_FOR_DATA_AGGREGATE

Represents the inverse property for ACMS$PROCEDURE_HAS_DATA_AGGREGATE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$PROCEDURE_DATA_AGGREGATE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$PROCEDURE_HAS_DATA_AGGREGATE |

## ACMS$PROCEDURE_FOR_ENTRY_PT

Represents the inverse property for ACMS$PROCEDURE_HAS_ENTRY_PT.

### Summary

| | |
|---|---|
| Defined By | CDD$PROGRAMS |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$PROCEDURE_ENTRY_PT |
| Traversal Direction | To owner |
| Inverse Property | ACMS$PROCEDURE_HAS_ENTRY_PT |

## ACMS$PROCEDURE_FOR_PROCEDURE

Represents the inverse property for ACMS$PROCEDURE_HAS_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$PROCEDURE_PROCEDURE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$PROCEDURE_HAS_PROCEDURE |

## ACMS$PROCEDURE_FOR_SERVER

Represents the inverse property for ACMS$PROCEDURE_HAS_SERVER.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$PROCEDURE_SERVER |
| Traversal Direction | To owner |
| Inverse Property | ACMS$PROCEDURE_HAS_SERVER |

## ACMS$PROCEDURE_HAS_DATA_AGGREGATE

Represents the inverse property for ACMS$PROCEDURE_FOR_DATA_AGGREGATE.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$PROCEDURE_DATA_AGGREGATE |
| Traversal Direction | To member |
| Inverse Property | ACMS$PROCEDURE_FOR_DATA_AGGREGATE |

## ACMS$PROCEDURE_HAS_ENTRY_PT

Represents the inverse property for ACMS$PROCEDURE_FOR_ENTRY_PT.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$PROCEDURE_ENTRY_PT |
| Traversal Direction | To member |
| Inverse Property | ACMS$PROCEDURE_FOR_ENTRY_PT |

## ACMS$PROCEDURE_HAS_PROCEDURE

Represents the inverse property for ACMS$PROCEDURE_FOR_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$PROCEDURE_SERVER |
| Traversal Direction | To member |
| Inverse Property | ACMS$PROCEDURE_FOR_PROCEDURE |

## ACMS$PROCEDURE_HAS_SERVER

Represents the inverse property for ACMS$PROCEDURE_FOR_SERVER.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$PROCEDURE_SERVER |
| Traversal Direction | To member |
| Inverse Property | ACMS$PROCEDURE_FOR_SERVER |

## ACMS$SERVER_FOR_ABORT_PROCEDURE

Represents the inverse property for ACMS$SERVER_HAS_ABORT_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_ABORT_PROCEDURE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$SERVER_HAS_ABORT_PROCEDURE |

## ACMS$SERVER_FOR_ACTION_PROCEDURE

Represents the inverse property for ACMS$SERVER_HAS_ACTION_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_ACTION_PROCEDURE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$SERVER_HAS_ACTION_PROCEDURE |

## ACMS$SERVER_FOR_BASED_ON

Represents the inverse property for ACMS$SERVER_HAS_BASED_ON.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_BASED_ON |
| Traversal Direction | To owner |
| Inverse Property | ACMS$SERVER_IS_BASED_ON |

## ACMS$SERVER_FOR_END_PROCEDURE

Represents the inverse property for ACMS$SERVER_HAS_END_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_END_PROCEDURE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$SERVER_HAS_END_PROCEDURE |

## ACMS$SERVER_FOR_INIT_PROCEDURE

Represents the inverse property for ACMS$SERVER_HAS_INIT_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_INIT_PROCEDURE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$SERVER_HAS_INIT_PROCEDURE |

## ACMS$SERVER_FOR_MODULE

Represents the inverse property for ACMS$SERVER_HAS_MODULE.

### Summary

| | |
|---|---|
| Defined By | CDD$COMPILED_MODULE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_MODULE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$SERVER_HAS_MODULE |

## ACMS$SERVER_HAS_ABORT_PROCEDURE

Represents the inverse property for ACMS$SERVER_FOR_ABORT_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_ABORT_PROCEDURE |
| Traversal Direction | To member |
| Inverse Property | ACMS$SERVER_FOR_ABORT_PROCEDURE |

## ACMS$SERVER_HAS_ACTION_PROCEDURE

Represents the inverse property for ACMS$SERVER_FOR_ACTION_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_ACTION_PROCEDURE |
| Traversal Direction | To member |
| Inverse Property | ACMS$SERVER_FOR_ACTION_PROCEDURE |

## ACMS$SERVER_HAS_END_PROCEDURE

Represents the inverse property for ACMS$SERVER_FOR_END_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_END_PROCEDURE |
| Traversal Direction | To member |
| Inverse Property | ACMS$SERVER_FOR_END_PROCEDURE |

## ACMS$SERVER_HAS_INIT_PROCEDURE

Represents the inverse property ACMS$SERVER_FOR_INIT_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_INIT_PROCEDURE |
| Traversal Direction | To member |
| Inverse Property | ACMS$SERVER_FOR_INIT_PROCEDURE |

## ACMS$SERVER_HAS_MODULE

Represents the inverse property for ACMS$SERVER_FOR_MODULE.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_MODULE |
| Traversal Direction | To member |
| Inverse Property | ACMS$SERVER_FOR_MODULE |

## ACMS$SERVER_IS_BASED_ON

Represents the inverse property for ACMS$SERVER_FOR_BASED_ON.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$SERVER_BASED_ON |
| Traversal Direction | To member |
| Inverse Property | ACMS$SERVER_FOR_BASED_ON |

## ACMS$SIZE

Represents an ACMS size.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_ACMS_SIZE |
| Defined By | ACMS$APPLICATION<br>ACMS$SERVER<br>ACMS$TASK<br>ACMS$TASK_GROUP<br>CDD$MENU |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## ACMS$SOURCE_TEXT

Represents the source text of the definition of any ACMS element type.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_ACMS_SRC_TXT |
| Defined By | ACMS$APPLICATION<br>ACMS$SERVER<br>ACMS$TASK<br>ACMS$TASK_GROUP<br>CDD$MENU |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## ACMS$TASK_FOR_BASED_ON

Represents the inverse property for ACMS$TASK_HAS_BASED_ON.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_BASED_ON |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_IS_BASED_ON |

## ACMS$TASK_FOR_DATA_AGG

Represents the inverse property for ACMS$TASK_HAS_DATA_AGG.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_DATA_AGG |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_HAS_DATA_AGG |

## ACMS$TASK_FOR_PROCEDURE

Represents the inverse property for ACMS$TASK_HAS_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$PROCEDURE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_PROCEDURE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_HAS_PROCEDURE |

## ACMS$TASK_FOR_TASK

Represents the inverse property for ACMS$TASK_HAS_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_ITEM |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_TASK |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_HAS_TASK |

## ACMS$TASK_FOR_VIDEO_DISPLAY

Represents the inverse property for ACMS$TASK_HAS_VIDEO_DISPLAY.

### Summary

| | |
|---|---|
| Defined By | CDD$VIDEO_DISPLAY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_VIDEO_DISPLAY |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_HAS_VIDEO_DISPLAY |

## ACMS$TASK_GROUP_FOR_BASED_ON

Represents the inverse property for ACMS$TASK_GROUP_HAS_BASED_ON.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_BASED_ON |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_GROUP_IS_BASED_ON |

## ACMS$TASK_GROUP_FOR_DATA_AGG

Represents the inverse property for ACMS$TASK_GROUP_HAS_DATA_AGG.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_DATA_AGG |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_GROUP_HAS_DATA_AGG |

## ACMS$TASK_GROUP_FOR_MSG_FILE

Represents the inverse property for ACMS$TASK_GROUP_HAS_MSG_FILE.

### Summary

| | |
|---|---|
| Defined By | MCS_BINARY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_MSG_FILE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_GROUP_HAS_MSG_FILE |

## ACMS$TASK_GROUP_FOR_RLB_FILE

Represents the inverse property for ACMS$TASK_GROUP_HAS_RLB_FILE.

### Summary

| | |
|---|---|
| Defined By | MCS_BINARY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_RLB_FILE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_GROUP_HAS_RLB_FILE |

## ACMS$TASK_GROUP_FOR_SERVER

Represents the inverse property for ACMS$TASK_GROUP_HAS_SERVER.

### Summary

| | |
|---|---|
| Defined By | ACMS$SERVER |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_SERVER |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_GROUP_HAS_SERVER |

## ACMS$TASK_GROUP_FOR_TASK

Represents the inverse property for ACMS$TASK_GROUP_HAS_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_TASK |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_GROUP_HAS_TASK |

## ACMS$TASK_GROUP_FOR_TDB_FILE

Represents the inverse property for ACMS$TASK_GROUP_HAS_TDB_FILE.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP<br>MCS_BINARY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_TDB_FILE |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_GROUP_HAS_TDB_FILE |

## ACMS$TASK_GROUP_FOR_VIDEO_DISPLAY

Represents the inverse property for ACMS$TASK_GROUP_HAS_VIDEO_DISPLAY.

### Summary

| | |
|---|---|
| Defined By | CDD$VIDEO_DISPLAY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_VIDEO_DISPLAY |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_GROUP_HAS_VIDEO_DISPLAY |

## ACMS$TASK_GROUP_HAS_DATA_AGG

Represents the inverse property for ACMS$TASK_GROUP_FOR_DATA_AGG.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_DATA_AGG |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_GROUP_FOR_DATA_AGG |

## ACMS$TASK_GROUP_HAS_MSG_FILE

Represents the inverse property for ACMS$TASK_GROUP_FOR_MSG_FILE.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_MSG_FILE |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_GROUP_FOR_MSG_FILE |

## ACMS$TASK_GROUP_HAS_RLB_FILE

Represents the inverse property for ACMS$TASK_GROUP_FOR_RLB_FILE.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_RLB_FILE |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_GROUP_FOR_RLB_FILE |

## ACMS$TASK_GROUP_HAS_SERVER

Represents the inverse property for ACMS$TASK_GROUP_FOR_SERVER.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_SERVER |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_GROUP_FOR_SERVER |

## ACMS$TASK_GROUP_HAS_TASK

Represents the inverse property for ACMS$TASK_GROUP_FOR_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_TASK |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_GROUP_FOR_TASK |

## ACMS$TASK_GROUP_HAS_TDB_FILE

Represents the inverse property for ACMS$TASK_GROUP_FOR_TDB_FILE.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_TDB_FILE |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_GROUP_FOR_TDB_FILE |

## ACMS$TASK_GROUP_HAS_VIDEO_DISPLAY

Represents the inverse property for ACMS$TASK_GROUP_FOR_VIDEO_DISPLAY.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_VIDEO_DISPLAY |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_GROUP_FOR_VIDEO_DISPLAY |

## ACMS$TASK_GROUP_IS_BASED_ON

Represents the inverse property for ACMS$TASK_FOR_BASED_ON.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_GROUP_BASED_ON |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_GROUP_FOR_BASED_ON |

## ACMS$TASK_HAS_DATA_AGG

Represents the inverse property for ACMS$TASK_FOR_DATA_AGG.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_GROUP |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_DATA_AGG |
| Direction | To member |
| Inverse Property | ACMS$TASK_FOR_DATA_AGG |

## ACMS$TASK_HAS_PROCEDURE

Represents the inverse property for ACMS$TASK_FOR_PROCEDURE.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_PROCEDURE |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_FOR_PROCEDURE |

## ACMS$TASK_HAS_TASK

Represents the inverse property for ACMS$TASK_FOR_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_TASK |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_FOR_TASK |

## ACMS$TASK_HAS_VIDEO_DISPLAY

Represents the inverse property for ACMS$TASK_FOR_VIDEO_DISPLAY.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_VIDEO_DISPLAY |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_FOR_VIDEO_DISPLAY |

## ACMS$TASK_ITEM_FOR_APP_TASK

Represents the inverse property for ACMS$TASK_ITEM_HAS_APP_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK<br>ACMS$APPLICATION_TASK_ITEM |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$APPLICATION_TASK_TSK |
| Traversal Direction | To owner |
| Inverse Property | ACMS$APP_TASK_HAS_TASK_ITEM |

## ACMS$TASK_ITEM_FOR_DATA_AGGREGATE

Represents the inverse property for ACMS$TASK_ITEM_HAS_DATA_AGGREGATE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGTE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_DATA_AGG |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_ITEM_HAS_DATA_AGGREGATE |

## ACMS$TASK_ITEM_FOR_TASK

Represents the inverse property for ACMS$TASK_ITEM_HAS_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_ITEM_TASK |
| Traversal Direction | To owner |
| Inverse Property | ACMS$TASK_ITEM_HAS_TASK |

## ACMS$TASK_ITEM_HAS_DATA_AGGREGATE

Represents the inverse property for ACMS$TASK_ITEM_FOR_DATA_AGGREGATE.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_DATA_AGG |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_ITEM_FOR_DATA_AGGREGATE |

## ACMS$TASK_ITEM_HAS_TASK

Represents the inverse property for ACMS$TASK_ITEM_FOR_TASK.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK_ITEM |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_ITEM_TASK |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_ITEM_FOR_TASK |

## ACMS$TASK_IS_BASED_ON

Represents the inverse property for ACMS$TASK_FOR_BASED_ON.

### Summary

| | |
|---|---|
| Defined By | ACMS$TASK |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | ACMS$TASK_BASED_ON |
| Traversal Direction | To member |
| Inverse Property | ACMS$TASK_FOR_BASED_ON |

## CDD$4GL_FOR_DATA_DESC

Represents the inverse property for CDD$4GL_HAS_DATA_DESC.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$4GL_REL |
| Traversal Direction | To owner |
| Inverse Property | CDD$4GL_HAS_DATA_DESC |

## CDD$4GL_FOR_VALUES

Represents the inverse property for CDD$4GL_HAS_VALUES.

### Summary

| | |
|---|---|
| Defined By | CDD$4GL<br>CDD$DATABASE<br>CDD$DATA_AGGREGATE<br>CDD$DATA_VALUE<br>DTR$DOMAIN<br>RALLY$DSD<br>RALLY$OBJECTS |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$4GL_GROUP_REL |
| Traversal Direction | To owner |
| Inverse Property | CDD$4GL_HAS_VALUES |

## CDD$4GL_HAS_DATA_DESC

Represents the inverse property for the CDD$4GL_FOR_DATA_DESC.

### CDD$4GL_HAS_DATA_DESC

#### Summary

| | |
|---|---|
| Defined By | CDD$REPORT |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$4GL_REL |
| Traversal Direction | To member |
| Inverse Property | CDD$4GL_FOR_DATA_DESC |

## CDD$4GL_HAS_VALUES

Represents the inverse property for the CDD$4GL_FOR_VALUES.

#### Summary

| | |
|---|---|
| Defined By | CDD$4GL<br>DTR$DOMAIN<br>RALLY$APPLICATION<br>RALLY$DSD |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$4GL_GROUP_REL |
| Traversal Direction | To member |
| Inverse Property | CDD$4GL_FOR_VALUES |

## CDD$BASED_ON_DATA_AGGREGATE

Describes a relation from member to owner of the CDD$DATA_AGGREGATE_BASED_ON relationship. This property is the scan of all element IDs representing CDD$DATA_AGGREGATE based upon the CDD$DATA_AGGREGATE element type including this property.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_AGGREGATE_BASED_ON |
| Direction | To owner |
| Inverse Property | CDD$DATA_AGGREGATE_IS_BASED_ON |

## CDD$BASED_ON_DATA_ELEMENT

Describes a relation from member to owner of the CDD$DATA_ELEMENT_BASED_ON relationship. This property is the scan of all element IDs representing CDD$DATA_ELEMENTS based upon the CDD$DATA_ELEMENT element type which includes this property.

### CDD$BASED_ON_DATA_ELEMENT

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ELEMENT_BASED_ON |
| Direction | To owner |
| Inverse Property | CDD$DATA_ELEMENT_IS_BASED_ON |

## CDD$COMPILED_HAS_DEPENDS_ON

Describes the scan of all elements IDs representing source modules, object modules, data values or data structures used to create a CDD$COMPILED_MODULE element type.

#### Summary

| | |
|---|---|
| Defined By | CDD$COMPILED_MODULE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$COMPILED_DEPENDS_ON |
| Direction | To member |
| Inverse Property | CDD$DEPENDED_ON_BY_COMPILED |

## CDD$COMPILED_IS_DERIVED_FROM

Describes the scan valued property comprising one or more source files used in compilation of the CDD$COMPILED_MODULE element type in which this property is included. This property is the ATIS implementation of the CDD$COMPILED_DERIVED_FROM relationship.

### Summary

| | |
|---|---|
| Defined By | CDD$COMPILED_MODULE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$COMPILED_DERIVED_FROM |
| Direction | To member |
| Inverse Property | CDD$IS_DERIVED_FROM_COMPILED |

## CDD$CONSTRAINT_EXP_FOR_DATA_VALUE

Represents the inverse property for CDD$CONSTRAINT_EXP_HAS_DATA_VALUE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$CONSTRAINT_EXPRESSION |
| Traversal Direction | To owner |
| Inverse Property | CDD$CONSTRAINT_EXP_HAS_DATA-VALUE |

## CDD$CONSTRAINT_EXP_HAS_DATA_VALUE

Represents the inverse property for the CDD$CONSTRAINT_EXP_FOR_DATA_VALUE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$CONSTRAINT_EXPRESSION |
| Traversal Direction | To member |
| Inverse Property | CDD$CONSTRAINT_EXP_FOR_DATA_VALUE |

## CDD$CONTAINED_BY_DATA_AGG

Describes the scan of element IDs that identify other instances of CDD$DATA_AGGREGATE which are contained within the DATA AGGREGATE being described. This property is the ATIS implementation of the relationship CDD$DATA_AGGREGATE_CONTAINS in the implemented information model of CDD/Repository. The complementary property is CDD$DATA_AGG_CONTAINS.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE<br>CDD$DATA_OVERLAY_AGGREGATE<br>CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_AGGREGATE_CONTAINS |
| Direction | To owner |
| Inverse Property | CDD$DATA_AGG_CONTAINS |

## CDD$CONTAINED_BY_DATA_OVER_AGG

Describes the scan of element IDs which identify instances of CDD$DATA_ AGGREGATE included within this instance of CDD$DATA_AGGREGATE. This property implements in ATIS the relationship CDD$DATA_OVERLAY_ CONTAINS and has a complementary property CDD$CONTAINS_ DATA_ OVERLAY.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_DIMENSION |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_OVERLAY_AGG_CONTAINS |
| Direction | To owner |
| Inverse Property | CDD$DATA_OVER_AGG_CONTAINS |

## CDD$CONTAINS_DATA_OVERLAY

Describes the complement of CDD$CONTAINED_BY_DATA_OVER_AGG and is the ATIS implementation of the CDD$DATA_OVERLAY_CONTAINS relationship.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_ELEMENT<br>CDD$DATA_OVERLAY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_OVERLAY_CONTAINS |
| Direction | To owner |
| Inverse Property | CDD$DATA_OVERLAY_CONTAINS |

## CDD$CREATED_TIME

Tells the date and time CDD/Repository created a repository element.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_CREATED_TIME |
| Defined By | MCS_NAMED_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_VMSTIME |
| Access | Read-only |

## CDD$DB_FOR_GROUP

Represents the inverse property for CDD$DB_HAS_GROUP.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE<br>CDD$RDB_DATABASE<br>DBM$REALM<br>DBM$SECURITY_SCHEMA<br>DBM$SET<br>DBM$STORAGE_SCHEMA<br>DBM$SUBSCHEMA |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATABASE_GROUP_REL |
| Traversal Direction | To owner |
| Inverse Property | CDD$DB_HAS_GROUP |

## CDD$DB_HAS_GROUP

Represents the inverse property for the CDD$DB_FOR_GROUP.

### CDD$DB_HAS_GROUP

#### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATABASE_GROUP_REL |
| Traversal Direction | To member |
| Inverse Property | CDD$DB_FOR_GROUP |

## CDD$DATABASE_FOR_FILE

Represents the inverse property for CDD$DATABASE_HAS_FILE.

#### Summary

| | |
|---|---|
| Defined By | MCS_BINARY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATABASE_FILE |
| Direction | To owner |
| Inverse Property | CDD$DATABASE_HAS_FILE |

## CDD$DATABASE_HAS_FILE

Describes the property that is the ATIS implementation of the CDD$DATABASE_FILE relationship. A logical CDD$DATABASE is physically stored in a CDD$DATABASE_FILE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATABASE_FILE |
| Direction | To member |
| Inverse Property | CDD$DATABASE_FOR_FILE |

## CDD$DATABASE_HAS_SCHEMA

Describes the property in the ATIS implementation of the relationship CDD$DATABASE_SCHEMA, where a CDD$DATABASE_SCHEMA describes the structure associated with a CDD$DATABASE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATABASE_SCHEMA |
| Direction | To member |
| Inverse Property | CDD$RDB_FOR_SCHEMA |

## CDD$DATABASE_KEY_LENGTH

Gives the length of the database key when creating a database.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DB_KEY_LEN |
| Defined By | CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATABASE_PARAMETERS

Names the parameters that were specified when creating a database. This property can be used as a database parameter buffer. Refer to the chapter on the relational call interface (RCI) to DSRI databases in the *Guide to DSRI Programming with Rdb/VMS*.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DB_PARAMETERS |
| Defined By | CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$DATATYPE

Represents a VAX standard datatype.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DATATYPE |
| Defined By | MCS_TYPE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_AGGREGATE_ALIGNMENT

Aligns fields within a data aggregate on specific boundaries within physical memory.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DA_ALIGNMENT |
| Defined By | CDD$DATA_AGGREGATE_CONTAINS<br>CDD$DATA_OVERLAY_CONTAINS |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_AGGREGATE_DB_KEY_LEN

The length in bytes of a database key Rdb/VMS uses to process relations.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DA_DB_KEY_LEN |
| Defined By | Unowned |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$DATA_AGGREGATE_FOR_COMPUTED

Represents the inverse property for CDD$DATA_AGGREGATE_HAS_COMPUTED.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-Write |
| Relation Traversed | CDD$DATA_AGGREGATE_COMPUTED_VAL |
| Direction | To owner |
| Inverse Property | CDD$DATA_AGG_HAS_COMPUTED_VAL |

## CDD$DATA_AGGREGATE_INPUT_PROMPT

Represents a text string used on entry screens to identify and prompt for entry of a set of values.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_IN_PROMPT |
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$DATA_AGGREGATE_IS_BASED_ON

Describes a scan valued property identifying other CDD$DATA_AGGREGATE instances upon which the element containing this property is based. This property is the ATIS implementation of CDD$DATA_AGGREGATE_BASED_ ON relationship and is the complement of CDD$ BASED_ON_DATA_ AGGREGATE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_AGGREGATE_BASED_ON |
| Direction | To member |
| Inverse Property | CDD$BASED_ON_DATA_AGGREGATE |

## CDD$DATA_AGG_CONTAINS

Describes the scan of element IDs that identify other instances of CDD$DATA_AGGREGATE which contain the DATA AGGREGATE being described. This property is the ATIS implementation of the relationship CDD$DATA_AGGREGATE_CONTAINS in the Entity relation model of CDD/Repository. The complementary property is CDD$CONTAINED_BY_DATA_AGG.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_AGGREGATE_CONTAINS |
| Direction | To member |
| Inverse Property | CDD$CONTAINED_BY_DATA_AGG |

## CDD$DATA_AGG_HAS_COMPUTED_VALUE

Describes the property in the ATIS implementatin for the CDD$DATA_AGGREGATE_COMPUTED_VAL relationship. This relationship associates a CDD$DATA_VALUE instance with a CDD$DATA_AGGREGATE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE<br>CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_AGGREGATE_COMPUTED_VAL |
| Direction | To member |
| Inverse Property | CDD$DATA_AGGREGATE_FOR_COMPUTED |

## CDD$DATA_AGG_RDB_CHECK_OPTION

Represents a Rdb/VMS check option.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DA_RDB_CHECK_OPT |
| Defined By | CDD$DATA_AGGEGATE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$DATA_ARRAY_DIMENSION

Describes the property in the ATIS implementatin for the CDD$DATA_ARRAY_HAS_DIMENSION relationship. This relationship associates a CDD$DATA_DIMENSION instance with either a CDD$DATA_ELEMENT or CDD$DATA_AGGREGATE that which is described by the array.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ARRAY_DIMENSION |
| Direction | To owner |
| Inverse Property | CDD$DATA_ARRAY_FOR_DIMENSION |

## CDD$DATA_ARRAY_FOR_DIMENSION

Represents the inverse property for CDD$DATA_ARRAY_HAS_DIMENSION.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_DIMENSION |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ARRAY_DIMENSION |
| Direction | To owner |
| Inverse Property | CDD$DATA_ARRAY_HAS_DIMENSION |

## CDD$DATA_ARRAY_MAJOR_ORDER

Specifies whether the column or row is treated as the primary ordering characteristic in a two-dimensional array.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DA_MAJOR_ORDER |
| Defined By | CDD$DATA_AGGREGATE<br>CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_ARRAY_ORDER

Describes the relative position of each subscript of a multidimensional array relative to the other subscripts.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DA_ORDER |
| Defined By | CDD$DATA_AGGREGATE_HAS_DIMENSION |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_DIMENSION_FOR_INDEX

Represents the inverse property for CDD$DATA_DIMENSION_HAS_INDEX.

### CDD$DATA_DIMENSION_FOR_INDEX

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_DIMENSION_INDEX |
| Direction | To owner |
| Inverse Property | CDD$DATA_DIMENSION_HAS_INDEX |

## CDD$DATA_DIMENSION_HAS_INDEX

Represents the inverse property for the CDD$DATA_DIMENSION_FOR_INDEX.

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$INDEX_SEGMENT |
| Direction | To member |
| Inverse Property | CDD$INDEX_SEGMENT_FOR_DATA_VALUE |

## CDD$DATA_DIMENSION_HIGH_BOUND

Defines the upper bound for an array subscript at run time. This property can be used as an expression buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DD_HIGH_BOUND |
| Defined By | CDD$DATA_DIMENSION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$DATA_DIM_HIGH_BOUND_HI_VAL

Defines the maximum legal size of the upper bound of an array dimension of variable size. This property can be used as an expression buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_REL_DD_HB_HIGH_VALUE |
| Defined By | CDD$DATA_DIMENSION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$DATA_DIM_HIGH_BOUND_LOW_VAL

Defines the minimum legal size of the upper bound of an array dimension of variable size. This property can be used as an expression buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DD_HB_LOW_VALUE |
| Defined By | CDD$DATA_DIMENSION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$DATA_ELEMENT_ALPHA_CASE

Indicates that a string valued property should be stored and presented as uppercase only, lowercase only, or case insensitive.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_ALPHA_CASE |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$DATA_ELEMENT_COLLATING_SEQ

Represents a collating sequence used by Rdb/VMS for sorting.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_COL_SEQ |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$DATA_ELEMENT_CURRENCY_SIGN

Represents a single character string used by CDD client products to display currency sign.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_CURRENCY_SIGN |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$DATA_ELEMENT_DATATYPE

Identifies the datatype of this element.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_DATATYPE |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_ELEMENT_DECIMAL_POINT

Represents a single character string used by CDD client products to specify the symbol representing a decimal point on display of a CDD$DATA_ELEMENT.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_DECIMAL_PT |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$DATA_ELEMENT_DIGITS

Specifies the number of digits that you can store in a numeric datatype.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_DIGITS |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_ELEMENT_DISPLAY_SCALE

Represents a signed integer value indicating the number of decimal places to be indicated in presentation of the CDD$DATA_ELEMENT value.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_DISPLAY_SCALE |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_ELEMENT_EDIT_STRING

Provides the output format when retrieving the field. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD /Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_EDT_STR_IN |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$DATA_ELEMENT_HELP_TEXT

Represents the character string presented to an entry screen in response to request for help information. Content is user defined.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_HELP_TXT |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$DATA_ELEMENT_INITIAL_VALUE

Provides an initial value to be stored for this field if no value is specified when doing data entry. This property can be used as an expression buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_INITIAL |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$DATA_ELEMENT_INPUT_PROMPT

Provides a prompt to be used when doing data entry on this field.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_IN_PROMPT |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$DATA_ELEMENT_INPUT_REQUIRED

Indicates that a value must be input when an instance of the CDD$DATA_ELEMENT is initially defined. Values permitted represent "Required" or "Optional."

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_INPUT_REQ |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_ELEMENT_IS_BASED_ON

Represents the inverse property of CDD$BASED_ON_DATA_ELEMENT.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ELEMENT_BASED_ON |
| Direction | To member |
| Inverse Property | CDD$BASED_ON_DATA_ELEMENT |

## CDD$DATA_ELEMENT_JUSTIFICATION

Defines the justification of a text string in a field.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_JUSTIFICATION |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_ELEMENT_LENGTH

Defines the length of the field.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_LENGTH |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$DATA_ELEMENT_OUTPUT_HEADER

Provides text to be used as a heading on reports. This property can be used as a text buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### CDD$DATA_ELEMENT_OUTPUT_HEADER

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_OUT_HEADER |
| Defined By | CDD$DATA_ELEMENT<br>DTR$TABLE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$DATA_ELEMENT_READ_ONLY

Represents instances that are read only.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_READ_ONLY |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_ELEMENT_SCALE

Defines the scale factor for the field. Negative scale is used for decimal numbers.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_SCALE |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_ELEMENT_SEGMENT_LENGTH

Defines the maximum length of the longest segment in a segmented string. This length is the optimal retrieval length. The segment length is specified in bytes. If you do not specify this property, the default segment length is 512 bytes.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_SEG_LENGTH |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$DATA_ELEMENT_SEG_SUBTYPE

Defines the specific type of string data a semented string contains.

### CDD$DATA_ELEMENT_SEG_SUBTYPE

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_SEG_SUBTYPE |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_ELE_FOR_COMPUTED_VALUE

Describes the scan valued property including the element IDs of elements used to compute the value of the CDD$DATA_ELEMENT entity type that includes this property. This property is the ATIS implementation of the CDD$DATA_ELEMENT_COMPUTED_VALUE relationship.

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ELEMENT_COMPUTED_VALUE |
| Direction | To owner |
| Inverse Property | CDD$DATA_ELE_HAS_COMPUTED_VALUE |

## CDD$DATA_ELE_FOR_INPUT_VALID

Represents the inverse property for CDD$DATA_ELE_HAS_INPUT_VALID.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ELEMENT_INPUT_VALID |
| Direction | To owner |
| Inverse Property | CDD$DATA_ELE_HAS_INPUT_VALID |

## CDD$DATA_ELE_FOR_POINTER_REF

Represents the inverse property for the CDD$DATA_ELE_FOR_POINTER_REF.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ELEMENT_POINTER_REF |
| Traversal Direction | To owner |
| Inverse Property | CDD$DATA_ELE_HAS_POINTER_REF |

## CDD$DATA_ELE_HAS_COMPUTED_VALUE

Represents the inverse property for the CDD$DATA_ELE_FOR_COMPUTED_VALUE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ELEMENT_COMPUTED_VALUE |
| Traversal Direction | To member |
| Inverse Property | CDD$DATA_ELE_FOR_COMPUTED_VALUE |

## CDD$DATA_ELE_HAS_INPUT_VALID

Represents the inverse property for the CDD$DATA_ELE_FOR_INPUT_VALID.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ELEMENT_INPUT_VALID |
| Direction | To member |
| Inverse Property | CDD$DATA_ELE_FOR_INPUT_VALID |

## CDD$DATA_ELE_HAS_POINTER_REF

Represents the inverse property for the CDD$DATA_ELE_FOR_POINTER_REF.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_ELEMENT_POINTER_REF |
| Direction | To member |
| Inverse Property | CDD$POINTER_REF_FOR_DATA_ELE |

## CDD$DATA_INSTANCE_FOR_PATH

Represents the inverse property for CDD$DATA_INSTANCE_HAS_PATH.

### CDD$DATA_INSTANCE_FOR_PATH

#### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE<br>CDD$DATA_AGGREGATE<br>CDD$DATA_DIMENSION<br>CDD$DATA_OVERLAY<br>CDD$DATA_OVERLAY_AGGREGATE<br>CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_INSTANCE_PATH |
| Direction | To owner |
| Inverse Property | CDD$DATA_INSTANCE_FOR_PATH |

## CDD$DATA_INSTANCE_FOR_ROOT

Represents the inverse property for the CDD$DATA_INSTANCE_HAS_ROOT.

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE<br>CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_INSTANCE_ROOT |
| Direction | To owner |
| Inverse Property | CDD$DATA_INSTANCE_HAS_ROOT |

## CDD$DATA_INSTANCE_HAS_PATH

Represents the inverse property for the CDD$DATA_INSTANCE_FOR_PATH.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE<br>CDD$DATA_INSTANCE<br>CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_INSTANCE_PATH |
| Traversal Direction | To member |
| Inverse Property | CDD$DATA_INSTANCE_FOR_PATH |

## CDD$DATA_INSTANCE_HAS_ROOT

Represents the inverse property for the CDD$DATA_INSTANCE_FOR_ROOT.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_INSTANCE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_INSTANCE_ROOT |
| Direction | To member |
| Inverse Property | CDD$DATA_INSTANCE_FOR_ROOT |

## CDD$DATA_INSTANCE_PATH_STEP

Contains the relative position of an element within a data instance path.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DI_PATH_STEP |
| Defined By | CDD$DATA_INSTANCE_PATH |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_OVER_AGG_CONTAINS

Represents the relationship a data overlay (variant) uses to contain fields or other structures.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_OVERLAY_AGGREGATE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_OVERLAY_AGG_CONTAINS |
| Direction | To member |
| Inverse Property | CDD$CONTAINED_BY_DATA_OVER_AGG |

## CDD$DATA_OVER_CONTAINS

Represents the inverse property for CDD$CONTAINS_DATA_OVERLAY.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_OVERLAY |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_OVERLAY_CONTAINS |
| Direction | To member |
| Inverse Property | CDD$CONTAINS_DATA_OVERLAY |

## CDD$DATA_OVER_FOR_ID

Represents the inverse property for CDD$DATA_OVER_HAS_ID.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_OVERLAY_IDENTIFICATION |
| Direction | To owner |
| Inverse Property | CDD$DATA_OVER_HAS_ID |

## CDD$DATA_OVER_HAS_ID

Represents the inverse property for the CDD$DATA_OVER_FOR_ID.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_OVERLAY |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_OVERLAY_IDENTIFICATION |
| Traversal Direction | To member |
| Inverse Property | CDD$DATA_OVER_FOR_ID |

## CDD$DATA_SEQUENCE_NUMBER

Identifies the CDD$INDEX_SEGMENT relationship within a set of keys under the owner.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_SEQ_NUMBER |
| Defined By | CDD$DATA_AGGREGATE_CONTAINS<br>CDD$DATA_OVERLAY_AGGREGATE_CONTAINS<br>CDD$DATA_OVERLAY_CONTAINS<br>CDD$FILE_AREA_ALLOCATION<br>CDD$INDEX_SEGMENT |
| Required With **new** | YES |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$DATA_VALUE_EXPRESSION

Provides an expression that a calling program can use to calculate the value of a field. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DV_EXPRESSION |
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$DATA_VALUE_FOR_DEPENDS_ON

Represents the inverse property for CDD$DATA_VALUE_HAS_DEPENDS_ON.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_VALUE_DEPENDS_ON |
| Direction | To owner |
| Inverse Property | CDD$DATA_VAL_HAS_DEPENDS_ON |

## CDD$DATA_VAL_HAS_DEPENDS_ON

Represents the inverse property for the CDD$DATA_VAL_FOR_DEPENDS_ON.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATA_VALUE_DEPENDS_ON |
| Direction | To member |
| Inverse Property | CDD$DATA_VALUE_FOR_DEPENDS_ON |

## CDD$DEPENDED_ON_BY_COMPILED

Represents the inverse property of CDD$COMPILED_HAS_DEPENDS_ON.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE<br>CDD$DATA_AGGREGATE<br>CDD$SOURCE_MODULE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$COMPILED_DEPENDS_ON |
| Direction | To owner |
| Inverse Property | CDD$COMPILED_HAS_DEPENDS_ON |

## CDD$DEPENDED_ON_BY_SOURCE

Represents the inverse property of CDD$SOURCE_HAS_DEPENDS_ON.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE<br>CDD$DATA_AGGREGATE<br>CDD$SOURCE_MODULE<br>MCS_BINARY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$SOURCE_DEPENDS_ON |
| Direction | To owner |
| Inverse Property | CDD$SOURCE_HAS_DEPENDS_ON |

## CDD$DERIVES_RDB_DB

Represents the inverse property of CDD$RDB_DB_IS_DERIVED_FROM.

### Summary

| | |
|---|---|
| Defined By | CDD$IDB_DATABASE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_DATABASE_DERIVED_FROM |
| Direction | To owner |
| Inverse Property | CDD$RDB_DB_IS_DERIVED_FROM |

## CDD$EDIT_STRING_COBOL

Represents the byte pattern used to edit entry of values into a COBOL procedure. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_EDT_STR_COB |
| Defined By | CDD$DATA_ELEMENT<br>DTR$TABLE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$EDIT_STRING_DTR

Represents the byte pattern used to edit entry of values into a VAX Datatrieve procedure. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_EDT_STR_DTR |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$EDIT_STRING_FORMS

Represents a byte pattern used to edit entry of values into the DECforms product. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_EDT_STR_FORMS |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$EDIT_STRING_PLI

Represents a byte pattern used to edit entry of values into a PL/1 procedure. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_EDT_STR_PLI |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$EDIT_STRING_RPG

Represents a byte pattern used to edit entry of values into an RPG procedure. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_EDT_STR_RPG |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$EVALUATION_TIME

Literal value specifying when the database will evaluate the constraint.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_EVALUATION_TIME |
| Defined By | CDD$CONSTRAINT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$EXTERNAL_REF

Represents a distributed definition used internally.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_EXTREF |
| Defined By | MCS_CONTAINED |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$FILE_FOR_ACCESS_INDEX

Represents the inverse property for CDD$FILE_HAS_ACCESS_INDEX.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$FILE_ACCESS_INDEX |
| Traversal Direction | To owner |
| Inverse Property | CDD$FILE_HAS_ACCESS_INDEX |

## CDD$FILE_FOR_AREA_ALLOC

Represents the inverse property for CDD$FILE_HAS_AREA_ALLOC.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_ALLOCATOIN |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$FILE_AREA_ALLOC |
| Traversal Direction | To owner |
| Inverse Property | CDD$FILE_HAS_AREA_ALLOC |

## CDD$FILE_FOR_INDEX

Represents the inverse property for CDD$FILE_HAS_INDEX.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$FILE_INDEXED_BY |
| Traversal Direction | To owner |
| Inverse Property | CDD$FILE_IS_INDEXED_BY |

## CDD$FILE_FOR_INDEX_SEG

Represents the inverse property for CDD$FILE_HAS_INDEX_SEG.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$FILE_INDEX_SEG |
| Traversal Direction | To owner |
| Inverse Property | CDD$FILE_HAS_INDEX_SEG |

## CDD$FILE_FOR_REL

Represents the inverse property for CDD$FILE_HAS_REL.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE<br>CDD$DATA_VALUE<br>CDD$FILE_ALLOCATION<br>CDD$FILE_ATTS<br>CDD$FILE_DEFINITION<br>CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RMS_FILE_DEFINITION |
| Traversal Direction | To owner |
| Inverse Property | CDD$FILE_HAS_RELS |

## CDD$FILE_HAS_ACCESS_INDEX

Represents the inverse property for the CDD$FILE_FOR_ACCESS_INDEX.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_ACCESS |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$FILE_ACCESS_INDEX |
| Traversal Direction | To member |
| Inverse Property | CDD$FILE_FOR_ACCESS_INDEX |

## CDD$FILE_HAS_AREA_ALLOC

Represents the inverse property for the CDD$FILE_FOR_AREA_ALLOC.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_DEFINITION<br>CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$FILE_AREA_ALLOC |
| Traversal Direction | To member |
| Inverse Property | CDD$AREA_ALLOC_HAS_FILE |

## CDD$FILE_HAS_INDEX_SEG

Represents the inverse property for the CDD$FILE_FOR_INDEX_SEG.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$FILE_INDEX_SEG |
| Traversal Direction | To member |
| Inverse Property | CDD$FILE_FOR_INDEX_SEG |

## CDD$FILE_HAS_RELS

Represents the inverse property for the CDD$FILE_FOR_RELS.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_ATTS<br>CDD$RMS_DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RMS_FILE_DEFINITION |
| Traversal Direction | To member |
| Inverse Property | CDD$REL_FOR_FILE |

## CDD$FILE_IS_INDEXED_BY

Represents the inverse property for CDD$INDEX_FOR_FILE.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$FILE_INDEXED_BY |
| Traversal Direction | To member |
| Inverse Property | CDD$INDEX_FOR_FILE |

## CDD$HAS_CONSTRAINT_EXPRESSION

Represents the inverse property for CDD$CONSTRAINT_EXP_FOR_DATA_VALUE.

### Summary

| | |
|---|---|
| Defined By | CDD$CONSTRAINT |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$CONSTRAINT_EXPRESSION |
| Direction | To member |
| Inverse Property | CDD$CONSTRAINT_EXP_FOR_DATA_VALUE |

## CDD$HAS_INDEX_SEGMENT

Represents the inverse property for the CDD$FOR_INDEX_SEGMENT.

### Summary

| | |
|---|---|
| Defined By | CDD$INDEX |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$INDEX_SEGMENT |
| Direction | To member |
| Inverse Property | CDD$INDEX_SEGMENT_FOR_DATA_VALUE |

## CDD$IMAGE_IS_DERIVED_FROM

Represents the inverse property for CDD$COMPILED_IS_DERIVED_FROM.

### Summary

| | |
|---|---|
| Defined By | CDD$EXECUTABLE_IMAGE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$COMPILED_DERIVED_FROM |
| Direction | To owner |
| Inverse Property | CDD$COMPILED_IS_DERIVED_FROM |

## CDD$INDEX_SEGMENT_FOR_DATA_VALUE

Represents the inverse property for CDD$INDEX_SEGMENT_HAS_DATA_VALUE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$INDEX_SEGMENT |
| Direction | To owner |
| Inverse Property | CDD$HAS_INDEX_SEGMENT |

## CDD$INPUT_EDIT_STRING

Describes the byte pattern used to edit entry of values into a compiler or special purpose data entry product. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_EDIT_STRING |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$INPUT_EDIT_STRING_DTR

Describes the byte pattern used to edit entry of values into a VAX Datatrieve Procedure. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_EDT_STR_IN_DTR |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$INPUT_EDIT_STRING_FORMS

Describes a byte pattern used to edit entry of values into the DECforms product. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DE_EDT_STR_IN_F |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$IS_DERIVED_FROM_COMPILED

Represents for the inverse property for CDD$COMPILED_IS_DERIVED_FROM.

### Summary

| | |
|---|---|
| Defined By | CDD$SOURCE_MODULE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$COMPILED_DERIVED_FROM |
| Direction | To owner |
| Inverse Property | CDD$COMPILED_IS_DERIVED_FROM |

## CDD$IS_DERIVED_FROM_IMAGE

Represents the inverse property for CDD$IMAGE_IS_DERIVED_FROM.

### Summary

| | |
|---|---|
| Defined By | CDD$COMPILED_MODULE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$IMAGE_DERIVED_FROM |
| Direction | To owner |
| Inverse Property | CDD$IMAGE_IS_DERIVED_FROM |

## CDD$IS_DERIVED_FROM_SOURCE

Represents the inverse property for CDD$SOURCE_IS_DERIVED_FROM.

### Summary

| | |
|---|---|
| Defined By | CDD$PROGRAM<br>CDD$SOURCE_MODULE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$SOURCE_DERIVED_FROM |
| Direction | To owner |
| Inverse Property | CDD$SOURCE_IS_DERIVED_FROM |

## CDD$MENU_FOR_CONTENTS

Represents the inverse property for CDD$MENU_HAS_CONTENTS.

### Summary

| | |
|---|---|
| Defined By | CDD$MENU |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$MENU_CONTAINS |
| Traversal Direction | To owner |
| Inverse Property | CDD$MENU_HAS_CONTENTS |

## CDD$MENU_HAS_CONTENTS

Represents the inverse property for the CDD$MENU_FOR_CONTENTS.

### Summary

| | |
|---|---|
| Defined By | CDD$MENU |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$MENU_CONTAINS |
| Traversal Direction | To member |
| Inverse Property | CDD$CONTENTS_FOR_MENU |

## CDD$MODIFIED_TIME

Specifies the date and time CDD/Repository last modified the repository element.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_MODIFIED_TIME |
| Defined By | MCS_NAMED_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_VMSTIME |
| Access | Read-only |

## CDD$NODE_NAME

Represents a node name used to describe a repository'slocation or the location to which a repository refers.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_NODE_NAME |
| Defined By | MCS_BINARY |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$OBJECT_FOR_FILE

Represents the inverse property for CDD$OBJECT_HAS_FILE.

### Summary

| | |
|---|---|
| Defined By | CDD$PROGRAMS<br>CDD$SOURCE_MODULE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$IN_FILE |
| Direction | To owner |
| Inverse Property | CDD$OBJECT_HAS_FILE |

## CDD$OBJECT_IN_FILE

Represents the inverse property for CDD$OBJECT_FOR_FILE.

### Summary

| | |
|---|---|
| Defined By | CDD$SOURCE_MODULE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$IN_FILE |
| Direction | To member |
| Inverse Property | CDD$OBJECT_FOR_FILE |

## CDD$OBJECT_KIND

Deterimines what kind of protocol an object-type element represents.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_OBJECT_KIND |
| Defined By | MCS_OBJECT_TYPE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MSC_SMALLINT |
| Access | Read-only |

## CDD$OWNER

Contains the VMS identifier of the CDD/Repository user who created the attribute.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_OWNER |
| Defined By | MCS_HISTORY |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-only |

## CDD$PROCESS_NAME_BAS

Supplies a name VAX BASIC can use when a BASIC program processes the entity.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_PROCESS_NAME_BAS |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$PROCESS_NAME_COB

Supplies a name VAX COBOL can use when a COBOL program processes the element type.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_PROCESS_NAME_COB |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$PROCESS_NAME_EBCDIC

Supplies an EBCDIC name for International Business Machines (IBM) information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_PROCESS_NAME_EBC |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$PROCESS_NAME_PAS

Supplies a name that VAX Pascal can use when a Pascal program processes the entity.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_PROCESS_NAME_PAS |
| Defined By | CDD$DATE_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$PROCESS_NAME_PLI

Supplies a name VAX PL/I can use when a PL/I program processes the entity.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_PROCESS_NAME_PLI |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$PROCESS_NAME_RPG

Supplies a name VAX RPG can use when an RPG program processes the entity.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_PROCESS_NAME_RPG |
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$PROTOCOL_TAG

Represents an internal property used to uniquely identify each protocol type.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_PROTOCOL_TAG |
| Defined By | MCS_ELEMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_TAG |
| Access | Read-only |

## CDD$QUALIFIED_NAME

Provides the fully qualified filename, translated down to concealed logical names, for the CDD$FILE entity.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_STOREDIN |
| Defined By | MCS_BINARY |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-create |

## CDD$REPORT_FOR_SOURCE

Represents the inverse property for CDD$REPORT_HAS_SOURCE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$REPORT_SOURCE |
| Traversal Direction | To owner |
| Inverse Property | CDD$REPORT_HAS_SOURCE |

## CDD$REPORT_HAS_SOURCE

Represents the inverse property for the CDD$REPORT_FOR_SOURCE.

### CDD$REPORT_HAS_SOURCE

#### Summary

| | |
|---|---|
| Defined By | CDD$REPORT |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$REPORT_SOURCE |
| Traversal Direction | To member |
| Inverse Property | CDD$REPORT_FOR_SOURCE |

## CDD$RDB_DA_DERIVES

Represents the inverse property for CDD$RDB_DA_IS_DERIVED_FROM

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_DA_DERIVED_FROM |
| Direction | To owner |
| Inverse Property | CDD$RDB_DA_IS_DERIVED_FROM |

## CDD$RDB_DA_IS_DERIVED_FROM

Represents the inverse property for CDD$RDB_DA_DERIVES.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_DA_DERIVED_FROM |
| Direction | To member |
| Inverse Property | CDD$RDB_DA_DERIVES |

## CDD$RDB_DB_IS_DERIVED_FROM

Represents the inverse property for CDD$DERIVES_RDB_DB.

### Summary

| | |
|---|---|
| Defined By | CDD$SOURCE_MODULE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Transverse Relation | CDD$RDB_DATABASE_DERIVED_FROM |
| Traversal Direction | To member |
| Inverse Property | CDD$DERIVES_RDB_DB |

## CDD$RDB_DESC_INDEX_SEG

Represents a property that specifies if the Rdb/VMS index is ascending or descending.

### CDD$RDB_DESC_INDEX_SEG

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RDB_DESC_IDX_SEG |
| Defined By | CDD$INDEX_SEGMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RDB_FOR_COL_SEQ

Represents the scan of element ID values that identify within ATIS one or more RDB$_DATABASE element types which use this object to establish the collating sequence of the database table.

#### Summary

| | |
|---|---|
| Defined By | CDD$COLLATING_SEQUENCE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_COL_SEQ |
| Direction | To owner |
| Inverse Property | CDD$RDB_HAS_COL_SEQ |

## CDD$RDB_FOR_CONSTRAINT

Represents the inverse property for CDD$RDB_HAS_CONSTRAINT.

### Summary

| | |
|---|---|
| Defined By | CDD$CONSTRAINT |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_CONSTRAINT |
| Traversal Direction | To owner |
| Inverse Property | CDD$RDB_HAS_CONSTRAINT |

## CDD$RDB_FOR_DATA_AGGREGATE

Represents the inverse property for CDD$RDB_HAS_DATA_AGGREGATE.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_DATA_AGGREGATE |
| Traversal Direction | To owner |
| Inverse Property | CDD$RDB_HAS_DATA_AGGREGATE |

## CDD$RDB_FOR_DATA_ELEMENT

Represents the inverse property for CDD$RDB_HAS_DATA_ELEMENT.

### CDD$RDB_FOR_DATA_ELEMENT

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_DATA_ELEMENT |
| Direction | To owner |
| Inverse Property | CDD$RDB_HAS_DATA_ELEMENT |

## CDD$RDB_FOR_INDEX

Represents the inverse property for CDD$RDB_HAS_INDEX.

#### Summary

| | |
|---|---|
| Defined By | CDD$INDEX |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_INDEX |
| Direction | To owner |
| Inverse Property | CDD$RDB_HAS_INDEX |

## CDD$RDB_FOR_SCHEMA

Represents the inverse property for CDD$RDB_HAS_SCHEMA.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE<br>CDD$DATABASE<br>CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATABASE_SCHEMA |
| Direction | To owner |
| Inverse Property | CDD$RDB_HAS_SCHEMA |

## CDD$RDB_HAS_COL_SEQ

Represents an Rdb/VMS schema containing a collating sequence.

### Summary

| | |
|---|---|
| Defined By | CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_COL_SEQ |
| Direction | To member |
| Inverse Property | CDD$RDB_FOR_COL_SEQ |

## CDD$RDB_HAS_CONSTRAINT

Represents an Rdb/VMS schema containing a constraint.

### Summary

| | |
|---|---|
| Defined By | CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_CONSTRAINT |
| Direction | To member |
| Inverse Property | CDD$RDB_FOR_CONSTRAINT |

## CDD$RDB_HAS_DATA_AGGREGATE

Represents an Rdb/VMS schema containing a data aggregate.

### Summary

| | |
|---|---|
| Defined By | CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Tranversed | CDD$RDB_DATA_AGGREGATE |
| Direction | To member |
| Inverse Property | CDD$RDB_FOR_DATA_AGGREGATE |

## CDD$RDB_HAS_DATA_ELEMENT

Represents an Rdb/VMS schema containing a data element.

### Summary

| | |
|---|---|
| Defined By | CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_DATA_ELEMENT |
| Direction | To member |
| Inverse Property | CDD$RDB_FOR_DATA_ELEMENT |

## CDD$RDB_HAS_INDEX

Represents an Rdb/VMS schema containing an index.

### Summary

| | |
|---|---|
| Defined By | CDD$RDB_DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RDB_INDEX |
| Direction | To member |
| Inverse Property | CDD$RDB_FOR_INDEX |

## CDD$RDB_HAS_SCHEMA

Represents the inverse property for the CDD$RDB_FOR_SCHEMA.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DATABASE_SCHEMA |
| Traversal Direction | To member |
| Inverse Property | CDD$RDB_FOR_SCHEMA |

## CDD$RDB_IDX_MAPPING

Indicates whether index mapping exists.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RDB_IDX_MAPPING |
| Defined By | CDD$INDEX_SEGMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MSC_SMALLINT |
| Access | Read-write |

## CDD$RDB_IDX_MAP_MAX

Represents the maximum values of index mapping.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RDB_IDX_MAP_MAX |
| Defined By | CDD$INDEX_SEGMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$RDB_IDX_MAP_MIN

Represents the minimum value for index mapping.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RDB_IDX_MAP_MIN |
| Defined By | CDD$INDEX_SEGMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$RDB_IDX_SIZE

Represents an Rdb/VMS index node size.

### CDD$RDB_IDX_SIZE

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RDB_IDX_SIZE |
| Defined By | CDD$INDEX_SEGMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RDB_NCS_NAME

Represents the name of the multinational character set used to sort the Rdb/VMS data. This property can be used as a database parameter buffer. Refer to the chapter on the relational call interface (RCI) to DSRI databases in the *Guide to DSRI Programming with Rdb/VMS*.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RDB_NCS_NAME |
| Defined By | CDD$COLLATING_SEQUENCE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$REQUIRED_ATTRIBUTE

Within the definition of an element protocol, indicates that a particular attribute must exist on each instance of the element.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_REQUIRED |
| Defined By | CDD$HAS_ATTRIBUTE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_DEFAULT_ACL

Represents the default ACL for an RMS database. This property can be used as an access control list buffer. Refer to the appendix on buffers in the *CDD /Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## CDD$RMS_DEFAULT_OWNER

Represents the default owner for an RMS database.

### CDD$RMS_DEFAULT_OWNER

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DEF_OWNER |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$RMS_FOR_DATA_AGGREGATE

Represents the inverse property for CDD$RMS_HAS_DATA_AGGREGATE.

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RMS_DATA_AGGREGATE |
| Traversal Direction | To owner |
| Inverse Property | CDD$RMS_HAS_DATA_AGGREGATE |

## CDD$RMS_FOR_FILE_DEFINITION

Represents the inverse property for CDD$RMS_HAS_FILE_DEFINITION.

### Summary

| | |
|---|---|
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RMS_FILE_DEFINITION |
| Traversal Direction | To owner |
| Inverse Property | CDD$RMS_FILE_HAS_DEFINITION |

## CDD$RMS_FAB_ALQ

Represents the allocation quantity in blocks. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_ALQ |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$RMS_FAB_BKS

Represents the bucket size. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_BKS |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_BLS

Represents the magnetic block size. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_BLS |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_CHAN_MODE

Represents the channel access model protection. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_CHAN_MODE |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_DEQ

Represents the default file extension quantity. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_DEQ |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_DNA

Represents the default file specification string address. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_DNA |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_DNS

Represents the default file string specification size. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_DNS |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FAC_BIO

Requests file access for doing block I/O operations that use Read (FAB$V_GET), Write (FAB$V_PUT, or the Space services. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FAC_BIO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FAC_BRO

Requests file access for doing either block I/O or record I/O as determined by the state of the RAB$V_BIO bit in the RAB at connect time. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FAC_BRO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FAC_DEL

Requests file access for invoking the Delete service (or the equivalent VAX language statement that deletes a record). Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FAC_DEL |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FAC_GET

Requests file access for invoking either the Get or Find service (or equivalent VAX language statment that read a record). Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FAC_GET |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FAC_PUT

Requests file access for invoking either the Put or Extend service (or the equivalent VAX language statement that writes a record or extends a file). Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FAC_PUT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FAC_TRN

Requests file access for invoking the Truncate service (or the equivalent VAX language statement that truncates a file). Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FAC_TRN |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FAC_UPD

Requests file access for invoking either an Update or Extend service (or the equivalent VAX language statement that rewrites a record or extends a file). Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FAC_UPD |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FNA

Represents an RMS file specification string address. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FNA |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FNS

Represents a file specification string size. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FNS |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_CBT

Represents contiguous best try; indicates that the file is to be allocated contiguously on a "best effort" basis. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_CBT |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_CIF

Creates a file, if nonexistent; opens an already existing file if it exists. If the file does not exist, it is created and the alternate success status RMS$_CREATED is returned to indicate that the file was created, not just opened. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_FAB_FOP_CIF

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_CIF |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_CTG

Represents a contiguous try; indicates that the space for the file is to be allocated contiguously. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_CTG |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_DFW

Represents a deferred write; indicates that writing back to the file of modified I/O buffers is to be deferred until the buffer must be used for other purposes. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_DFW |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_DLT

Deletes file on Close; indicates that the file is to be deleted when closed. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_DLT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_MXV

Maximizes version number; indicates that the version number of the file should be the maximum of the version number given in the file specification, or one greater than the highest version number for an existing file in the same directory with the same file name and file type. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_MXV |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_NEF

Indicates not to position at the end; inhibits positioning to the end of a file when a tape file is opended and the file access field indicates a Put service. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_NEF |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_NFS

Represents a non-file-structured; indicates (on an Open or Create service) that the volume is to be processed in a non-file-structured manner. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_NFS |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_POS

Represents current position; directs VMS RMS to position the magnetic tape volume set immediately after the most recently closed file when it creates the next file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_POS |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_RCK

Represents read-check; specifies that transfers from disk volumes are to be checked by a read-compare operation, which doubles the amount of disk I/O at some increase in reliability. Refer to the *VMS Record Management Services Reference Manual* for further information.

## CDD$RMS_FAB_FOP_RCK

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_RCK |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_RWC

Specifies rewind file on Close; specifies that the magnetic tape volume is to be rewound when the file is closed. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_RWC |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_RWO

Specifies rewind on Open; specifies that the magnetic tape volume is to be rewound before the file is opened or created. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_RWO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_SCF

Submits command file on Close; indicates that the file is to be submitted as a batch-command file to the process-default batch queue (SYS$BATCH) when the file is closed. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_SCF |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_SPL

Spools file on Closed; indicates that the file is to be spooled to the process-default print queue (SYS$PRINT) when the file is closed. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_SPL |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_SQO

Represents sequential processing; indicates that the file can be processed only in a sequential manner, permitting certain processing optimizations. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_SQO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_SUP

Supersedes existing file; allows an existing file to be superseded on a Create service by a new file of the same name, type, and version. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_SUP |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_TEF

Truncates at end of file; indicates that unused space allocated to a file is to be deallocated on a Close service. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_TEF |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_TMD

Represents a temporary file marked for deletion; indicates that a temporary file is to be created but is to be deleted when the file is closed. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_TMD |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_TMP

Represents a temporary file; indicates that a temporary file is to be created and retained, but that no directory entry is to be made for it. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_TMP |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_UFO

Represents an open user file; indicates that VMS RMS operations for this file are limited to opening it or creating it. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_UFO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FOP_WCK

Specifies write-check; specifies that transfers from disk volumes are to be checked by a read-compare operation, which doubles the amount of disk I/O at some increase in reliability. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FOP_WCK |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_FSZ

Represents the fixed-length control area size field that is used only for variable with fixed-length control records. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_FAB_FSZ

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_FSZ |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_GBC

Represents the global buffer count field that indicates the requested number of global buffers for a file. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_GBC |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_LNM_MODE

Represents the logical name translation access mode subfield. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_LNM_MODE |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_MRN

Represents the maximum record number field that applies only to relative files and indicates the highest record number that can be written to a file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_MRN |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$RMS_FAB_MRS

Represents the maximum record size field that defines the size of all records in a file with fixed-length records, the maximum size of variable-length records, the maximum size of the data area for variable with fixed-length control records, and the cell size (minus overhead) for relative files. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_FAB_MRS

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_MRS |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$RMS_FAB_ORG

Represents the file organization fields that assigns the organization of the file. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_ORG |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_RAT

Represents the record attributes field that indicates the record control information associated with each record in the file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_RAT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_RAT_BLK

Represents a records attribute option applicable to sequential files only. Indicates that records are not permitted to cross block boundaries. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_RAT_BLK |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_RFM

Represents the record format field that specifies the format for all the records in a file. Refer to the *VMS Record Management Services Reference Manual* for further information.

## CDD$RMS_FAB_RFM

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_RFM |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_RTV

Represents the retrieval window size field that specifies the number of retrieval pointers VMS RMS is to maintain in memory for the file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_RTV |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_SHR_DEL

Allows other users to delete records from the file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_SHR_DEL |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_SHR_GET

Allows other users to read the file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_SHR_GET |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_SHR_MSE

Allows multistream access and is relevant for record operations only. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_SHR_MSE |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_SHR_NIL

Prohibits any file sharing by other users. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_SHR_NIL |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_SHR_PUT

Allows other users to write records to the file or to extend the file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_SHR_PUT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_SHR_UPD

Allows other users to update records that currently exist in the file or extend the file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_SHR_UPD |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FAB_SHR_UPI

Allows other users to assume responsibility for interlocking of multiple simultanteous accessors of a VMS RMS file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_FAB_SHR_UPI

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_FAB_SHR_UPI |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_FILE_HAS_DEFINITION

Represents an RMS schema containing a definition.

#### Summary

| | |
|---|---|
| Defined By | CDD$RMS_DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RMS_FILE_DESCRIPTION |
| Traversal Direction | To member |
| Inverse Property | CDD$RMS_FOR_FILE_DEFINITION |

## CDD$RMS_HAS_DATA_AGGREGATE

Represents an RMS schema containing a data aggregate.

### Summary

| | |
|---|---|
| Defined By | CDD$RMS_DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$RMS_DATA_AGGREGATE |
| Traversal Direction | To member |
| Inverse Property | CDD$RMS_FOR_DATA_AGGREGATE |

## CDD$RMS_RAB_BKT

Represents bucket code. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_BKT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_MBC

Represents multiblock count. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_MBC |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_MBF

Represents multibuffer count. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_MBF |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_PBF

Represents prompt buffer address. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_PBF |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_PSZ

Represents prompt buffer size. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_PSZ |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_RAC

Represents record access code. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_RAC |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_RFA

Represents record file address. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_RFA |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_QUADWORD |
| Access | Read-write |

## CDD$RMS_RAB_ROP_ASY

Represents record-processing options. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_ASY |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_BIO

Represents record-processing options. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_BIO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_CCO

Represents the Cancel Ctrl/O option; guarantees that terminal output is not discarded if the operator press Ctrl/O. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_RAB_ROP_CCO

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_CCO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_CVT

Represents the Convert option; changes characters to uppercase on a read from a terminal. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_CVT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_EOF

Represents the end-of-file option; indicates that VMS RMS is to position the record stream to the end of the file for the connect record operation only. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_EOF |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_ETO

Represents the extended terminal operation option; indicates presence of a Terminal XAB (XABTRM) to describe terminal input using extended terminal characteristics. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_ETO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_FDL

Represents the fast delete option; reduces the time required to delete records in indexed files. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_RAB_ROP_FDL

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_FDL |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_KGE

Returns the record with an equal key value, or the next key value according to the sort order for the current key of reference. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_KGE |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_KGT

Returns the record with the next key value according to the sort order for the current key of reference. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_KGT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_LIM

Permits you to use VMS RMS as a limit sensor when accessing a file sequentially. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_LIM |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_LOA

Specifies to load buckets according to the fill size established when the file is created. Refer to the *VMS Record Management Services Reference Manual* for further information.

## CDD$RMS_RAB_ROP_LOA

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_LOA |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_LOC

Represents the locate mode option; under specified conditions, you have the option of specifying locate mode instead of move mode, the default method of buffer handling. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_LOC |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_NLK

Represents a record locking option; specifies that the record accessed through a Get or Find service is not be locked. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_NLK |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_NXR

Represents the synonym for RAB$V_KGT. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_NXR |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_PMT

Represents the prompt option; indicates that the contents of the prompt buffer are to be used as a prompt for reading data from a terminal. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_RAB_ROP_PMT

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_PMT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_PTA

Represents the purge type-ahead option; eliminates any information that may be in the type-ahead buffer on a read from a terminal. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_PTA |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_RAH

Represents the read-ahead option; used with multiple buffers to indicate read-ahead options. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_RAH |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_REA

Represents the lock record-for read option; specifies that the record is to be locked for a read option for this process, while allowing other acccessors to read the record (but not to modify the record). Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_REA |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_RLK

Represents the lock record-for write option; specifies that a user who locks a record for modification is allowing the locked record to be read by other accessors. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_RLK |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_RNE

Represents the read-no echo option; indicates that input data is not echoed (displayed) on the terminal as it is entered on the keyboard. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_RNE |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_RNF

Represents the read-no filter option; indicates that Ctrl/U, Ctrl/R, and DELETE are not to be considered control commands on terminal input but are to be passed to the application program. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_RNF |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_RRL

Represents the read-regardless of lock option; reads the record even if another stream has locked the record. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_RRL |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_TMO

Represents the timeout option; if the RAB$V_WAT option was specified, the RAB$B_TMO field contains the maximum time value, in seconds, to be allowed for a record input wait caused by a locked record. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_RAB_ROP_TMO

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_TMO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_TPT

Represents the truncate-on-put option; specifies that a Put or Write service using sequential record access mode can occur at any point in the file, truncating the file at that point. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_TPT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_UIF

Represents the update-if option; if a Put service is invoked for a record that already exists in the file, the operation is converted to an Update. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_UIF |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_ULK

Represents the manual unlocking option; prohibits VMS RMS from automatically unlocking records. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_ULK |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_WAT

Represents the wait option; if the record is currently locked by another stream, wait until it is available. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_WAT |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_ROP_WBH

Represents the write-behind option; used with multiple buffers. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_ROP_WBH |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_RAB_TMO

Represents a timeout period. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RAB_TMO |
| Defined By | CDD$FILE_ACCESS<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABALL_AID

Represents the area identification number. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_AID |
| Defined By | CDD$FILE_ALLOCATION |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABALL_ALN

Represents an alignment boundary type. Refer to the *VMS Record Management Services Reference Manual* for further information.

## CDD$RMS_XABALL_ALN

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_ALN |
| Defined By | CDD$FILE_ALLOCATION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABALL_ALQ

Represents an allocation quantity. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_ALQ |
| Defined By | CDD$FILE ALLOCATION<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$RMS_XABALL_AOP_CBT

Represents allocation options. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_AOP_CBT |
| Defined By | CDD$FILE_ALLOCATION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABALL_AOP_CTG

Represents the contiguous option; indicates that the intial allocation extension must use contiguous blocks only. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_AOP_CTG |
| Defined By | CDD$FILE_ALLOCATION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABALL_AOP_HRD

Represents the hard option; if the requested alignment cannot be performed, an error will be returned. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_XABALL_AOP_HRD

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_AOP_HRD |
| Defined By | CDD$FILE_ALLOCATION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABALL_AOP_ONC

Represents the cylinder boundary option; indicates that VMS RMS is to begin the allocation on any available cylinder boundary. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_AOP_ONC |
| Defined By | CDD$FILE_ALLOCATION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABALL_BZK

Represents the bucket size using numeric values ranging from 0 through 63 to represent the number of blocks in a bucket. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_BKZ |
| Defined By | CDD$FILE_ALLOCATION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABALL_DEQ

Represents the default extension quantity field that specifies the number (0 through 65,535) of blocks to be added when VMS RMS automatically extends the area. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_DEQ |
| Defined By | CDD$FILE_ALLOCATION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABALL_LOC

Represents the location field that contains a numeric value that indicates the beginning point for area allocation. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_LOC |
| Defined By | CDD$FILE_ALLOCATION<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## CDD$RMS_XABALL_RFI

Represents a related file identifier. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_RFI |
| Defined By | CDD$FILE_ALLOCATION<br>CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_QUADWORD |
| Access | Read-write |

## CDD$RMS_XABALL_VOL

Represents a related volume number. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABALL_VOL |
| Defined By | CDD$FILE_ALLOCATION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_AREA

Represents an RMS key definition XAB (XABKEY). Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_AREA |
| Defined By | CDD$FILE_AREA_ALLOCATION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_CHG

Represents a key value within the record in the file that can be changed by a program during an Update service. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_XABKEY_CHG

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_CHG |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_DAT_NCMPR

Specifies not to compress data. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_DAT_NCMPR |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_DFL

Represents the data bucket fill size. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_DFL |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_DTP

Represents the data type. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_DTP |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_DUP

Specifies that the key value within the record in the file may have the same key value as another record (or other records) within the file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_XABKEY_DUP

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_DUP |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_IDX_NCMPR

Specifies not to compress the index. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_IDX_NCMPR |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_IFL

Represents the index budget file size. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_IFL |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_KEY_NCMPR

Specifies not to compress the key. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_KEY_NCMPR |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_KNM

Represents the key name buffer address. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_KNM |
| Defined By | CDD$FILE_DEFINITION<br>CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## CDD$RMS_XABKEY_NUL

Represents the null key value. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_NUL |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_NULL_VALUE

Represents the null value. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_NULL_VALUE |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_PROLOG

Represents the prolog level. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_PROLOG |
| Defined By | CDD$FILE_INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_REF

Represents the key reference. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_XABKEY_REF

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_REF |
| Defined By | CDD$FILE_INDEXED_BY |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABKEY_SEG

Defines characteristics of each key in an indexed file. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABKEY_SEG |
| Defined By | CDD$FILE_INDEX_SEGMENT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_GRP_NODEL

Specifies no delete-access for group. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_GRP_NODEL |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_GRP_NOEXE

Specifies no execute-access for group. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_GRP_NOEXE |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_GRP_NOREAD

Specifies no read-access for group. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_XABPRO_GRP_NOREAD

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_GRP_NOREA |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_GRP_NOWRITE

Specifies no write-access for group. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_WLD_NOWRI |
| Defined By | CDD$FILE_DEFINTION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_MTACC

Represents magnetic tape accessiblity. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_MTACC |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_OWN_NODEL

Specifies no delete-access for owner. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_OWN_NODEL |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALL_INT |
| Access | Read-write |

## CDD$RMS_XABPRO_OWN_NOEXE

Specifies no execute-access for owner. Refer to the *VMS Record Management Services Reference Manual* for further information.

## CDD$RMS_XABPRO_OWN_NOEXE

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_OWN_NOEXE |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_OWN_NOREAD

Specifies no read-access for owner. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_WLD_NOREA |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_OWN_NOWRITE

Specifies no write-access for owner. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_WLD_NOWRI |
| Defined By | CDD$FILE_DEFINTION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_PROPAGATE

Represents the protection specification for a file. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_PROPAGATE |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_SYS_NODEL

Specifies no delete-access for system. Refer to the *VMS Record Management Services Reference Manual* for further information.

### CDD$RMS_XABPRO_SYS_NODEL

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_OWN_NODEL |
| Defined By | CDD$FILE_DEFINTION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_SYS_NOEXE

Specifies no execute-access for system. Refer to the *VMS Record Management Services Reference Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_OWN_NOEXE |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_SYS_NOREAD

Specifies no read-access for system. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_OWN_NOREA |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_SYS_NOWRITE

Specifies no write-access for system. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_SYS_NOWRI |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_WLD_NODEL

Specifies no delete-access for world. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_WLD_NODEL |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_WLD_NOEXE

Specifies no execute-access for world users. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_WLD_NOEXE |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_WLD_NOREAD

Specifies no read-access for world. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_WLD_NOREA |
| Defined By | CDD$FILE_DEFINITIN |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$RMS_XABPRO_WLD_NOWRITE

Specifies no write-access for world. Refer to the *VMS Record Management Services Reference Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_XABPRO_WLD_NOWRI |
| Defined By | CDD$FILE_DEFINITION |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$SOURCE_HAS_DEPENDS_ON

Represents a dependency upon another object.

### CDD$SOURCE_HAS_DEPENDS_ON

#### Summary

| | |
|---|---|
| Defined By | CDD$SOURCE_MODULE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$SOURCE_DEPENDS_ON |
| Direction | To member |
| Inverse Property | CDD$DEPENDED_ON_BY_SOURCE |

## CDD$SOURCE_IS_DERIVED_FROM

Represents the inverse property for CDD$IS_DERIVED_FROM_SOURCE.

#### Summary

| | |
|---|---|
| Defined By | CDD$SOURCE_MODULE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$SOURCE_DERIVED_FROM |
| Direction | To member |
| Inverse Property | CDD$IS_DERIVED_FROM_SOURCE |

## CDD$UNIQUE_INDEX

Literal value specifying whether duplicates are allowed in an index defined for a DSRI-compliant database.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_UNIQUE_INDEX |
| Defined By | CDD$INDEX |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## CDD$VIDEO_DISPLAY_BASED_ON

Represents the inverse property of CDD$VIDEO_DISPLAY_FOR_BASED_ON.

### Summary

| | |
|---|---|
| Defined By | CDD$VIDEO_DISPLAY |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$VIDEO_DISPLAY |
| Traversal Direction | To member |
| Inverse Property | CDD$VIDEO_DISPLAY_FOR_BASED_ON |

## CDD$VIDEO_DISPLAY_ELEMENTS

Represents the property that identifies the fields or records used on a form.

**CDD$VIDEO_DISPLAY_ELEMENTS**

### Summary

| | |
|---|---|
| Defined By | CDD$VIDEO_DISPLAY |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DISPLAY_ELEMENTS |
| Traversal Direction | To member |
| Inverse Property | CDD$VIDEO_DISPLAY_FOR_ELEMENTS |

## CDD$VIDEO_DISPLAY_FOR_BASED_ON

Represents the inverse property for CDD$VIDEO_DISPLAY_HAS_BASED_ON.

### Summary

| | |
|---|---|
| Defined By | CDD$VIDEO_DISPLAY |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$VIDEO_DISPLAY_BASED_ON |
| Traversal Direction | To owner |
| Inverse Property | CDD$VIDEO_DISPLAY_IS_BASED_ON |

## CDD$VIDEO_DISPLAY_FOR_ELEMENTS

Represents the inverse property for CDD$VIDEO_DISPLAY_HAS_ELEMENTS.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$DISPLAY_ELEMENTS |
| Traversal Direction | To owner |
| Inverse Property | CDD$VIDEO_DISPLAY_ELEMENTS |

## CDD$VIDEO_DISPLAY_IS_BASED_ON

Represents the inverse property for CDD$VIDEO_DISPLAY_FOR_BASED_ON.

### Summary

| | |
|---|---|
| Defined By | CDD$VIDEO_DISPLAY |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$VIDEO_DISPLAY_BASED_ON |
| Traversal Direction | To member |
| Inverse Property | CDD$VIDEO_DISPLAY_FOR_BASED_ON |

## DBM$DBMS_FOR_AREAS

Represents the inverse property for DBM$DBMS_HAS_AREAS.

### DBM$DBMS_FOR_AREAS

#### Summary

| | |
|---|---|
| Defined By | DBM$AREA |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$AREAS |
| Traversal Direction | To owner |
| Inverse Property | DBM$DBMS_HAS_AREAS |

## DBM$DBMS_FOR_DATA_AGGREGATES

Represents the inverse property for DBM$DBMS_HAS_DATA_AGGREGATES.

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$DATA_AGGREGATES |
| Traversal Direction | To owner |
| Inverse Property | DBM$DBMS_HAS_DATA_AGGREGATES |

## DBM$DBMS_FOR_REALMS

Represents the inverse property for DBM$DBMS_HAS_REALMS.

### Summary

| | |
|---|---|
| Defined By | DBM$REALM |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$REALMS |
| Traversal Direction | To owner |
| Inverse Property | DBM$DBMS_HAS_REALMS |

## DBM$DBMS_FOR_SECURITY_SCHEMAS

Represents the inverse property for DBM$DBMS_HAS_SECURITY_SCHEMAS.

### Summary

| | |
|---|---|
| Defined By | DBM$SECURITY_SCHEMA |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SECURITY_SCHEMAS |
| Traversal Direction | To owner |
| Inverse Property | DBM$DBMS_HAS_SECURITY_SCHEMAS |

## DBM$DBMS_FOR_SEC_SCHEMA_INST

Represents the inverse property for DBM$DBMS_HAS_SEC_SCHEMA_INST.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SEC_SCHEMA_INST |
| Traversal Direction | To owner |
| Inverse Relation | DBM$HAS_SEC_SCHEAM_INST |

## DBM$DBMS_FOR_SETS

Represents the inverse property for DBM$DBMS_HAS_SETS.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SETS |
| Traversal Direction | To owner |
| Inverse Relation | DBM$DBMS_HAS_SETS |

## DBM$DBMS_FOR_SET_MEMBERS

Represents the inverse property for DBM$DBMS_HAS_SET_MEMBERS.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SET_MEMBERS |
| Traversal Direction | To owner |
| Inverse Property | DBM$DBMS_HAS_SET_MEMBERS |

## DBM$DBMS_FOR_SET_OWNERS

Represents the inverse property for DBM$DBMS_HAS_SET_OWNERS.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SET_OWNERS |
| Traversal Direction | To owner |
| Inverse Property | DBM$DBMS_HAS_SET_OWNERS |

## DBM$DBMS_FOR_STORAGE_SCHEMAS

Represents the inverse property for DBM$DBMS_HAS_STORAGE_SCHEMAS.

### Summary

| | |
|---|---|
| Defined By | DBM$STORAGE_SCHEMA |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$STORAGE_SCHEMAS |
| Traversal Direction | To owner |
| Inverse Property | DBM$DBMS_HAS_STORAGE_SCHEMAS |

## DBM$DBMS_FOR_STO_SCHEMA_INST

Represents the inverse property for DBM$DBMS_HAS_STO_SCHEMA_INST.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$STO_SCHEMA_INST |
| Traversal Direction | To owner |
| Inverse Relation | DBM$HAS_STO_SCHEMA_INST |

## DBM$DBMS_FOR_SUBSCHEMAS

Represents the inverse property for DBM$DBMS_HAS_SUBSCHEMAS.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE<br>DBM$SUBSCHEMA |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SUBSCHEMAS |
| Traversal Direction | To owner |
| Inverse Relation | DBM$DBMS_HAS_SUBSCHEMAS |

## DBM$DBMS_FOR_SUBSCHEMAS_INST

Represents the inverse property for DBM$DBMS_HAS_SUBSCHEMAS_INST.

### Summary

| | |
|---|---|
| Defined By | DBM$SUBSCHEMA |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SUBSCHEMAS_INST |
| Traversal Direction | To owner |
| Inverse Relation | DBM$HAS_SUBSCHEMAS_INST |

## DBM$DBMS_HAS_AREAS

Represents the inverse property for DBM$DBMS_FOR_AREAS.

### Summary

| | |
|---|---|
| Defined By | DBM$REALM<br>DBM$SCHEMA |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$AREAS |
| Traversal Direction | To member |
| Inverse Property | DBM$DBMS_FOR_AREAS |

## DBM$DBMS_HAS_DATA_AGGREGATES

Represents the inverse property for DBM$DBMS_FOR_DATA_AGGREGATES.

### Summary

| | |
|---|---|
| Defined By | DBM$SUBSCHEMA |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$DATA_AGGREGATES |
| Traversal Direction | To member |
| Inverse Property | DBM$DBMS_FOR_DATA_AGGREGATES |

## DBM$DBMS_HAS_REALMS

Represents the inverse property for DBM$DBMS_FOR_REALMS.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE<br>DBM$SCHEMA |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Transversed | DBM$REALMS |
| Traversal Direction | To member |
| Inverse Property | DBM$DBMS_FOR_REALMS |

## DBM$DBMS_HAS_SECURITY_SCHEMAS

Represents the inverse property for DBM$DBMS_FOR_SECURITY_SCHEMAS.

### Summary

| | |
|---|---|
| Defined By | DBM$SCHEMA |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SECURITY_SCHEMAS |
| Traversal Direction | To member |
| Inverse Property | DBM$DBMS_FOR_SECURITY_SCHEMAS |

## DBM$DBMS_HAS_SETS

Represents the inverse property for DBM$DBMS_FOR_SETS.

### Summary

| | |
|---|---|
| Defined By | DBM$SUBSCHEMA |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SETS |
| Traversal Direction | To member |
| Inverse Property | DBM$DBMS_FOR_SETS |

## DBM$DBMS_HAS_SET_MEMBERS

Represents the inverse property for DBM$DBMS_FOR_SET_MEMBER.

### Summary

| | |
|---|---|
| Defined By | DBM$SET |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SET_MEMBERS |
| Traversal Direction | To member |
| Inverse Property | DBM$DBMS_FOR_SET_MEMBERS |

## DBM$DBMS_HAS_SET_OWNERS

Represents the inverse property for DBM$DBMS_FOR_SET_OWNERS.

### Summary

| | |
|---|---|
| Defined By | DBM$SET |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SET_OWNERS |
| Traversal Direction | To member |
| Inverse Property | DBM$DBMS_FOR_SET_OWNERS |

## DBM$DBMS_HAS_STORAGE_SCHEMAS

Represents the inverse property for DBM$DBMS_FOR_STORAGE_SCHEMAS.

### Summary

| | |
|---|---|
| Defined By | DBM$SCHEMA |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$STORAGE_SCHEMAS |
| Traversal Direction | To member |
| Inverse Property | DBM$DBMS_FOR_STORAGE_SCHEMAS |

## DBM$DBMS_HAS_SUBSCHEMAS

Represents the inverse property for DBM$DBMS_FOR_SUBSCHEMAS.

### Summary

| | |
|---|---|
| Defined By | DBM$SCHEMA |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SUBSCHEMAS |
| Traversal Direction | To owner |
| Inverse Relation | DBM$DBMS_FOR_SUBSCHEMAS |

## DBM$DDBLK

Comprises the internal representation of a parcticular schema, subschema, storage schema, or security schema. This property can be used as a DBMS internal buffer.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DBM_DDBLK |
| Defined By | DBM$SCHEMA<br>DBM$SEC_SCHEMA<br>DBM$STO_SCHEMA<br>DBM$SUBSCHEMA |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## DBM$DDNAME

Comprises the internal representation of a parcticular schema, subschema, storage schema, or security schema. This property can be used as a DBMS internal buffer.

### Summary

| | |
|---|---|
| Defined By | DBM$SCHEMA<br>DBM$SEC_SCHEMA<br>DBM$STO_SCHEMA<br>DBM$SUBSCHEMA |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## DBM$DDSET

Comprises the internal representation of a parcticular schema, subschema, storage schema, or security schema. This property can be used as a DBMS internal buffer.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DBM_DDSET |
| Defined By | DBM$SCHEMA<br>DBM$SEC_SCHEMA<br>DBM$STO_SCHEMA<br>DBM$SUBSCHEMA |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## DBM$HAS_SEC_SCHEMA_INST

Represents the inverse property for DBM$FOR_SEC_SCHEMA_INST.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SEC_SCHEMA_INST |
| Direction | To member |
| Inverse Relation | DBM$DBMS_FOR_SEC_SCHEMA_INST |

## DBM$HAS_STO_SCHEMA_INST

Represents the inverse property for DBM$FOR_STO_SCHEMA_INST.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$STO_SCHEMA_INST |
| Direction | To member |
| Inverse Relation | DBM$DBMS_FOR_STO_SCHEMA_INST |

## DBM$HAS_SUBSCHEMAS_INST

Represents the inverse property for DBM$FOR_SUBSCHEMAS_INST.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DBM$SUBSCHEMAS_INST |
| Direction | To member |
| Inverse Relation | DBM$DBMS_FOR_SUBSCHEMA_INST |

## DBM$ID

Represents the subschema identification.

### Summary

| | |
|---|---|
| Tag | |
| Defined By | CDD$DATA_AGGREGATE_CONTAINS<br>DBM$AREA<br>DBM$REALM<br>DBM$RECORD<br>DBM$SET |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## DBM$INSERTION

Represents the subschema set insertion mode.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DBM_INSERTION |
| Defined By | DBM$SET_MEMBER |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## DBM$ORDER

Represents the subschema set ordering mode.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DBM_ORDER |
| Defined By | DBM$SET_MEMBER |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## DBM$RELSCH_STAMP

Represents the subschema element type create date and time stamp.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DBM_RELSCH_STAMP |
| Defined By | DBM$SEC_SCHEMA<br>DBM$STO_SCHEMA<br>DBM$STO_SCHEMA |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_VMSTIME |
| Access | Read-write |

## DBM$RETENTION

Represents the subschema set retention mode.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DBM_RETENTION |
| Defined By | DBM$SET_MEMBER |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## DTR$CODES

Represents a list of VAX Datatrieve table code strings.

### DTR$CODES

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_CODES |
| Defined By | DTR$TABLE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## DTR$CODE_FIELD

Represents the name of a domain field that can be used as a code field in a VAX Datatrieve table. This property can be used as an edit string buffer. Refer to the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_CODE_FIELD |
| Defined By | DTR$TABLE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$DATABASE_PATHNAME

Represents the CDD path name of a database used in a Datatrieve domain definition.

### Summary

| | |
|---|---|
| Defined By | DTR_PATHNAME |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$DESCRIPTIONS

Represents a list of translations strings that correspond to codes in DTR$CODES. This property can be used as an unstructured buffer that can contain one or more string literals.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_DESCRIPTIONS |
| Defined By | DTR$TABLE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## DTR$DESCRIPTION_FIELD

Represents the name of a domain field that can be used as a translation field in a VAX Datatrieve table.

### DTR$DESCRIPTION_FIELD

#### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_DESC_FIELD |
| Defined By | DTR$TABLE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$DOMAIN_FOR_SOURCE

Represents the inverse property for DTR$DOMAIN_HAS_SOURCE.

#### Summary

| | |
|---|---|
| Defined By | CDD$DATA_AGGREGATE<br>CDD$DATABASE<br>CDD$DATA_VALUE<br>CDD$DOMAIN<br>DTR$DATABASE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$DOMAIN_SOURCE |
| Traversal Direction | To owner |
| Inverse Property | DTR$DOMAIN_HAS_SOURCE |

## DTR$DOMAIN_HAS_SOURCE

Represents the inverse property for DTR$DOMAIN_FOR_SOURCE.

### Summary

| | |
|---|---|
| Defined By | DTR$DOMAIN<br>DTR$PROCEDURE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$DOMAIN_SOURCE |
| Traversal Direction | To owner |
| Inverse Property | DTR$DOMAIN_FOR_SOURCE |

## DTR$DOMAIN_PATH

Represents the CDD path name of a domain that a VAX Datatrieve table is based on.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_DOMAIN_PATH |
| Defined By | DTR$TABLE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$DOMAIN_TYPE

Represents a type of VAX Datatrieve domain such as, RMS, DBMS, view, port, remote, Rdb/VMS, and Rdb/VMS relation.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_DOMAIN_TYPE |
| Defined By | DTR$DOMAIN |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_SMALLINT |
| Access | Read-write |

## DTR$DTR_DB_FOR_SCHEMA

Represents the inverse property for DTR$DTR_DB_HAS_SCHEMA.

### Summary

| | |
|---|---|
| Defined By | DTR$SCHEMA |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$DATABASE_SCHEMA |
| Traversal Direction | To owner |
| Inverse Property | DTR$DTR_DB_HAS_SCHEMA |

## DTR$DTR_DB_FOR_SUBSCHEMA

Represents the inverse property for DTR$DTR_DB_HAS_SUBSCHEMA.

### Summary

| | |
|---|---|
| Defined By | DBM$SUBSCHEMA |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$DATABASE_SUBSCHEMA |
| Traversal Direction | To owner |
| Inverse Property | DTR$DTR_DB_HAS_SUBSCHEMA |

## DTR$DTR_DB_HAS_SCHEMA

Represents the inverse property for DTR$DTR_DB_FOR_SCHEMA.

### Summary

| | |
|---|---|
| Defined By | DTR$DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$DATABASE_SCHEMA |
| Traversal Direction | To member |
| Inverse Property | DTR$DTR_DB_FOR_SCHEMA |

## DTR$DTR_DB_HAS_SUBSCHEMA

Represents the inverse property for DTR$DTR_DB_FOR_SUBSCHEMA.

### Summary

| | |
|---|---|
| Defined By | DTR$DATABASE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$DATABASE_SUBSCHEMA |
| Traversal Direction | To member |
| Inverse Property | DTR$DTR_DB_FOR_SUBSCHEMA |

## DTR$FIELD_TREE

Represents a record layout already translated to VAX Datatrieve internal format used in records defined within VAX Datatrieve with the optimized clause. This property can be used as a VAX Datatrieve internal buffer.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_FIELD_TREE |
| Defined By | CDD$DATA_AGGREGATE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## DTR$FILE

Represents the filename, which can be either an RMS or a database root file, of the actual data storage in a VAX Datatrieve domain definition.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_FILE |
| Defined By | DTR$DATABASE<br>DTR$DOMAIN |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$FORM_LIBRARY

Represents the name of a form library (FMS, TDMS) or a form shareable image or file (DECforms) containing a form associated to a VAX Datatrieve domain.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_FORM_LIBRARY |
| Defined By | DTR$DOMAIN |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$FORM_NAME

Represents the name of a form associated with a VAX Datatrieve domain.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_FORM_NAME |
| Defined By | DTR$DOMAIN |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$FORMAT_REC

Represents the dictionary path name of a record structure used by VAX Datatrieve to exchange data with DECforms forms. The record can be either in DMU or in CDO format.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_FORMAT_REC |
| Defined By | DTR$DOMAIN |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$MENU_FOR_CONTENTS

Represents the inverse property for DTR$MENU_HAS_CONTENTS.

### Summary

| | |
|---|---|
| Defined By | CDD$MENU |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | CDD$MENU_CONTAINS |
| Traversal Direction | To owner |
| Inverse Property | CDD$MENU_HAS_CONTENTS |

## DTR$NODE

Represents the name of the node where a remote VAX Datatrieve domain resides.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_NODE |
| Defined By | DTR$DOMAIN |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$PLOT_ARGS

Represents a list of arguments for a VAX Datatrieve plot. This property can be used as an unstructured buffer that can contain one or more string literals.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_PLOT_ARGS |
| Defined By | DTR$PLOT |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## DTR$PLOT_CODE

Represents a binary string designating the content of a VAX Datatrieve plot block. This property can be used as an unstructured buffer that can contain a binary string representing the contents of a VAX Datatrieve plot block.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_PLOT_CODE |
| Defined By | DTR$PLOT |
| Required With **new** | Yes |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## DTR$PROCEDURE_FOR_DTR_DB

Represents the inverse property for DTR$PROCEDURE_HAS_DTR_DB.

### Summary

| | |
|---|---|
| Defined By | DTR$DATABASE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$PROCEDURE_DTR_DB |
| Traversal Direction | To owner |
| Inverse Property | DTR$PROCEDURE_HAS_DTR_DB |

## DTR$PROCEDURE_FOR_FIELD

Represents the inverse property for DTR$PROCEDURE_HAS_FIELD.

### Summary

| | |
|---|---|
| Defined By | CDD$DATA_ELEMENT |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$PROCEDURE_FIELD |
| Traversal Direction | To owner |
| Inverse Property | DTR$PROCEDURE_HAS_FIELD |

## DTR$PROCEDURE_FOR_SOURCE

Represents the inverse property for DTR$PROCEDURE_HAS_SOURCE.

### Summary

| | |
|---|---|
| Defined By | DTR$DOMAIN |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$PROCEDURE_SOURCE |
| Traversal Direction | To owner |
| Inverse Property | DTR$PROCEDURE_HAS_SOURCE |

## DTR$PROCEDURE_FOR_TABLE

Represents the inverse property for DTR$PROCEDURE_HAS_TABLE.

### Summary

| | |
|---|---|
| Defined By | DTR$TABLE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$PROCEDURE_TABLE |
| Traversal Direction | To owner |
| Inverse Property | DTR$PROCEDURE_HAS_TABLE |

## DTR$PROCEDURE_HAS_FIELD

Represents the inverse property for DTR$PROCEDURE_FOR_FIELD.

### Summary

| | |
|---|---|
| Defined By | DTR$PROCEDURE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$PROCEDURE_FIELD |
| Traversal Direction | To member |
| Inverse Property | DTR$PROCEDURE_FOR_FIELD |

## DTR$PROCEDURE_HAS_DTR_DB

Represents the inverse property for DTR$PROCEDURE_FOR_DTR_DB.

### Summary

| | |
|---|---|
| Defined By | DTR$PROCEDURE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$PROCEDURE_DTR_DB |
| Traversal Direction | To member |
| Inverse Property | DTR$PROCEDURE_FOR_DTR_DB |

## DTR$PROCEDURE_HAS_SOURCE

Represents the inverse property for DTR$PROCEDURE_FOR_SOURCE.

### Summary

| | |
|---|---|
| Defined By | DTR$PROCEDURE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$PROCEDURE_SOURCE |
| Traversal Direction | To member |
| Inverse Property | DTR$PROCEDURE_FOR_SOURCE |

## DTR$PROCEDURE_HAS_TABLE

Represents the inverse property for DTR$PROCEDURE_FOR_TABLE.

### Summary

| | |
|---|---|
| Defined By | DTR$PROCEDURE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$PROCEDURE_TABLE |
| Traversal Direction | To member |
| Inverse Property | DTR$PROCEDURE_FOR_TABLE |

## DTR$RECORD_PATHNAME

Represents the CDD path name of a record associated with a VAX Datatrieve domain.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_RECORD_PATH |
| Defined By | DTR$DOMAIN |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$SCHEMA

Represents a name of a DBMS schema used within VAX Datatrieve.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_SCHEMA |
| Defined By | DTR$DATABASE |
| Required With **new** | False |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$SOURCE_TEXT

Represents the source text of the definition of any VAX Datatrieve element type. This property can be used as a text buffer. Refer the appendix on buffers in the *CDD/Repository Callable Interface Manual* for further information.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_SOURCE_TEXT |
| Defined By | DTR$DATABASE<br>DTR$DOMAIN<br>DTR$PLOT<br>DTR$PROCEDURE<br>DTR$TABLE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_MEMBLOCK |
| Access | Read-write |

## DTR$SUBSCHEMA

Represents a name of a DBMS subschema used within VAX Datatrieve.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_DTR_SUBSCHEMA |
| Defined By | DTR$DATABASE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## DTR$TABLE_FOR_SOURCE

Represents the inverse property for DTR$TABLE_HAS_SOURCE.

### Summary

| | |
|---|---|
| Defined By | DTR$DOMAIN |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$TABLE_SOURCE |
| Traversal Direction | To owner |
| Inverse Property | DTR$TABLE_HAS_SOURCE |

## DTR$TABLE_HAS_SOURCE

Represents the inverse property for DTR$TABLE_FOR_SOURCE.

### Summary

| | |
|---|---|
| Defined By | DTR$TABLE |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | DTR$TABLE_SOURCE |
| Traversal Direction | To member |
| Inverse Property | DTR$TABLE_FOR_SOURCE |

## NSDS$DRIVER_NAME

Represents the external program used to access this record for the NSDS product.

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_STRING |
| Access | Read-write |

## RALLY$AFILE_POINTER

Represents a property from the RALLY application definition to the file it lives in that is known as an AFILE.

### Summary

| | |
|---|---|
| Tag | NAD$K_ATT_RALLY_AFILE_PTR |
| Defined By | RALLY$APP_CONT |
| Required With **new** | No |
| Type | Normal |
| Datatype | MCS_LONGINT |
| Access | Read-write |

## RALLY$APP_FOR_CONTENTS

Represents the inverse property for RALLY$APP_HAS_CONTENTS.

### Summary

| | |
|---|---|
| Defined By | CDD$MENU<br>CDD$REPORT<br>CDD$VIDEO_DISPLAY<br>RALLY$ADL<br>RALLY$DSD<br>RALLY$PACKET<br>RALLY$TASK |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | RALLY$APPLICATION_CONTAINS |
| Traversal Direction | To owner |
| Inverse Property | RALLY$APP_HAS_CONTENTS |

## RALLY$APP_HAS_CONTENTS

Represents the inverse property for RALLY$APP_FOR_CONTENTS.

### RALLY$APP_HAS_CONTENTS

#### Summary

| | |
|---|---|
| Defined By | RALLY$APPLICAITON |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | RALLY$APPLICATION_CONTAINS |
| Traversal Direction | To member |
| Inverse Property | RALLY$APP_FOR_CONTENTS |

## RALLY$DSD_HAS_SOURCE

Represents the inverse property for RALLY$DSD_FORM_SOURCE.

#### Summary

| | |
|---|---|
| Defined By | RALLY$DSD |
| Required With **new** | No |
| Type | Relation |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | RALLY$DSD_SOURCE |
| Traversal Direction | To member |
| Inverse Property | RALLY$SOURCE_FROM_DSD |

## RALLY$SOURCE_FROM_DSD

Represents the inverse property for RALLY$DSD_HAS_SOURCE.

## RALLY$SOURCE_FROM_DSD

### Summary

| | |
|---|---|
| Defined By | CDD$DATABASE<br>CDD$DATA_VALUE |
| Required With **new** | No |
| Type | Closure |
| Datatype | MCS_SCAN |
| Access | Read-write |
| Relation Traversed | RALLY$DSD_SOURCE |
| Traversal Direction | To owner |
| Inverse Property | RALLY$DSD_HAS_SOURCE |

# A

# Type_Hierarchy Structure

The following illustrates the type hierarchy structure.

```
0 MCS_ELEMENT
  1 MCS_EVENT
  1 MCS_NAME_ELEMENT
    2 CDD$DIRECTORY
    2 MCS_DATABASE
    2 MCS_PARTITION
    2 MCS_PERSISTANCE_PROCESS
    2 MCS_CONTEXT
    2 MCS_VERSION
      3 MCS_AGGREGATE
        4 MCS_BINARY
          5 MCS_BINARY_TOOL
          5 MCS_TEXT
            6  MCS_TEXT_TOOL
            6 CDD$SOURCE_MODULE
            6 MCS_C_SOURCE_FILE
            6 MCS_LISTING_FILE
            6 MCS_FOR_SOURCE_FILE
            6 MCS_COB_SOURCE_FILE
          5 CDD$FILE
          5 MCS_DIAGNOSTIC_FILE
          5 MCS_ANALYSIS_DATA_FILE
          5 MCS_OBJECT_FILE
          5 MCS_EXECUTABLE_FILE
        4 MCS_COMPOSITE
          5 MCS_COLLECTION
          5 CDD$DATA_GROUP
            6 CDD$DATA_AGGREGATE
            6 CDD$DATA_ELEMENT
            6 CDD$DATA_OVERLAY
            6 CDD$DATA_OVERLAY_AGGREGATE
          5 CDD$CDD_DATABASE
            6 CDD$DATABASE
            6 CDD$RDB_DATABASE
            6 DBM$SCHEMA
            6 DBM$SECURITY_SCHEMA
            6 DBM$STORAGE_SCHEMA
            6 DBM$SUBSCHEMA
```

```
        6 DBM$SET
        6 DBM$RMS_DATABASE
      5 CDD$4GL
        6 CDD$MENU
        6 CDD$REPORT
        6 CDD$VIDEO_DISPLAY
      5 RALLY$DATA_SOURCE_DEFINTION
      5 RALLY$APPLICATION
      5 MCS_METHOD_MAP
      5 DTR$DOMAIN
      5 ACMS$GROUPS
        6 ACMS$TASK
        6 ACMS$TASK_GROUP
        6 ACMS$APPLICATION
3 MCS_TYPE
  4 MCS_DATA_TYPE
  4 MCS_ELEMENT_TYPE
    5 MCS_RELATION_TYPE
  4 MCS_PROPERTY_TYPE
  4 CDD$LINK_TYPE
3 CDD$DATA_DESCRIPTION
  4 CDD$DATA_DIMENSION
  4 CDD$DATA_INSTANCE
  4 CDD$DATA_VALUE
  4 CDD$CONSTRAINT
  4 CDD$INDEX
  4 CDD$COLLATING_SEQUENCE
3 DBM$OBJECT
  4 DBM$AREA
  4 DBM$REALM
3 CDD$FILE_ATTS
  4 CDD$FILE_ACCESS
  4 CDD$FILE_ALLOCATION
  4 CDD$FILE_DEFINTION
  4 CDD$FILE_INDEX
3 CDD$PROGRAMS
  4 CDD$COMPILED_MODULES
  4 CDD$EXECUTABLE_IMAGE
  4 GEN$PROGRAM
  4 CDD$PROCEDURE
3 RALLY$OBJECTS
  4 RALLY$TASK
  4 RALLY$PACKET
  4 RALLY$PROCEDURE
  4 ACMS$SERVER
  4 ACMS$TASK_TASK_ITEM
  4 ACMS$PROCEDURE
  4 ACMS$TASK_PROCEDURE_ITEM
  4 ACMS$APPL_TASK_ITEM
  4 ACMS$APPL_SERVER_ITEM
  4 ACMS$MENU_TASK_ITEM
```

```
    3 CDD$KEYWORD
    3 MCS_MESSAGE
    3 MCS_MSGARG
    3 MCS_TOOL
      4 MCS_METHOD
        5 MCS_VALIDATION
1 MCS_RELATION
  2 MCS_DEPENDS_ON
    3 MCS_AGGREGATE_ART
      4 CDD$HAS_LINK
      4 CDD$ATT_VALIDATION
      4 MCS_COLLECTION_PART
        5 MCS_COLLECTION_PART
        5 CDD$4GL_GROUP_REL
          6 CDD$DISPLAY_ELEMENTS
          6 CDD$MENU_CONTAINS
          6 CDD$VIDEO_DISPLAY_BASED_ON
          6 DTR$DOMAIN_SOURCE
          6 RALLY$APPLICATION_CONTAINS
          6 RALLY$DSD_SOURCE
        5 ACMS$GROUP_REL
          6 ACMS$TASK_DATA_AGG
          6 ACMS$TASK_VIDEO_DISPLAY
          6 ACMS$TASK_GROUP_DATA_AGG
          6 ACMS$TASK_GROUP_VIDEO_DISPLAY
          6 ACMS$TASK_GROUP_TASK
          6 ACMS$APPLICATION_TASK_GROUP
        5 CDD$DATA_DESC_GROUP_REL
          6 CDD$DATA_AGGREGATE_BASED_0N
          6 CDD$DATA_AGGREGATE_CONTAINS
          6 CDD$DATA_OVERLAY_AGG_CONTAINS
          6 CDD$DATA_OVERLAY_CONTAINS
          6 CDD$DATA_ELEMENT_BASED_0N
        5 CDD$DATABASE_GROUP_REL
          6 CDD$DATABASE_SCHEMA
          6 CDD$RDB_CONSTRAINT
          6 DBM$REALMS
          6 DBM$DATA_AGGREGATES
          6 DBM$SETS
          6 DBM$SET_MEMBERS
          6 DBM$SET_OWNERS
          6 DBM$SECURITY_SCHEMAS
          6 DBM$STORAGE_SCHEMAS
          6 DBM$SUBSCHEMAS
        5 CDD$RDB_DATA_REL
          6 CDD$RDB_DATA_AGGREGATE
          6 CDD$RDB_DATA_ELEMENT
          6 CDD$RDB_INDEX
        5 CDD$RMS_DATA_AGGREGATE
      4 MCS_HAS_PROPERTY
        5 MCS_HAS_COMPUTED_PROPERTY
        5 MCS_HAS_RELATION_PROPERTY
```

```
  4 MCS_HAS_RELATION
  4 MCS_HAS_RELATION_MEMBER
  4 MCS_OWNS_RELATION
3 MCS_METHOD_PARAMETER
  4 CDD$SOURCE_DEPENDS_ON
  4 CDD$SOURCE_DERIVED_FROM
  4 CDD$COMPILED_DEPENDS_ON
  4 CDD$COMPILED_DERIVED_FROM
  4 CDD$IMAGE_DERIVED_FROM
3 CDD$DATA_DESC_REL
  4 CDD$DATA_INSTANCE_PATH
  4 CDD$DATA_INSTANCE_ROOT
  4 CDD$DATA_OVERLAY_IDENTIFICATION
  4 CDD$DATA_VALUE_DEPENDS_ON
  4 CDD$DATA_AGGRGATE_COMPUTED_VAL
  4 CDD$DATA_ARRAY_HAS_DIMENSION
  4 CDD$DATA_DIM_HIGH_BOUND_HI_REL
  4 CDD$DATA_DIM_HIGH_BOUND_LOW_REL
  4 CDD$DATA_DIMENSION_HIGH_BD_REL
  4 CDD$DATA_DIMENSION_INDEX
  4 CDD$DATA_DIMENSION_LOW_BD_REL
  4 CDD$DATA_DTA_ELE_VALS
    5 CDD$DATA_ELEMENT_COMPUTED_VAL
      6 CDD$DATA_ELEMENT_INITIAL_DEF
      6 CDD$DATA_ELEMENT_MISSING_DEF
    5 CDD$DATA_ELEMENT_POINTER_RF
    5 CDD$DATA_ELEMENT_INPUT_VALID
  4 CDD$RDB_COL_SEQ
3 CDD$DATABASE_REL
  4 CDD$DATABASE_FILE
  4 CDD$INDEX_SEGMENT
  4 CDD$CONSTRAINT_EXPRESSION
  4 CDD$RDB_DA_DERIVED_FROM
  4 CDD$RDB_DATABASE_DERIVED_FROM
  4 CDD$IN_FILE
3 CDD$DBM_REL
  4 DBM$SEC_SCHEMA_INST
  4 DBM$STO_SCHEMA_INST
  4 DBM$SUBSCHEMAS_INST
3 DBM$AREAS
3 CDD$FILE_REL
  4 CDD$FILE_ACCESS_INDEX
  4 CDD$FILE_AREA_ALLOC
  4 CDD$FILE_INDEX_SEG
  4 CDD$FILE_INDEXED_BY
  4 CDD$FILE_FILE_DEFINITION
3 CDD$4GL_REL
  4 CDD$REPORT_SOURCE
3 ACMS$OBJECT_REL
  4 ACMS$TASK_PROCEDURE
  4 ACMS$TASK_TASK
  4 ACMS$TASK_BASED_ON
```

```
    4 ACMS$TASK_GROUP_BASED_ON
    4 ACMS$TASK_GROUP_TDB_FILE
    4 ACMS$TASK_GROUP_RLB_FILE
    4 ACMS$TASK_GROUP_MSG_FILE
    4 ACMS$APPLICATION_TASK
    4 ACMS$APPLICATION_ADB_FILE
    4 ACMS$APPLICATION_SERVER
    4 ACMS$MENU_MDB_FILE
    4 ACMS$MENU_TASK
    4 ACMS$SERVER_INIT_PROCEDURE
    4 ACMS$SERVER_ABORT_PROCEDURE
    4 ACMS$SERVER_END_PROCEDURE
    4 ACMS$SERVER_ACTION_PROCEDURE
    4 ACMS$SERVER_MODULE
    4 ACMS$SERVER_BASED_ON
    4 ACMS$PROCEDURE_DATA_AGGREGATE
    4 ACMS$PROCEDURE_SERVER
    4 ACMS$PROCEDURE_PROCEDURE
    4 ACMS$PROCEDURE_ENTRY_PT
    4 ACMS$TASK_ITEM_DATA_AGGREGATE
    4 ACMS$TASK_ITEM_TASK
    4 ACMS$APPLICATION_TASK_TSK
    4 ACMS$APPLICATION_TASK_GRP
    4 ACMS$APPLICATION_SRV_SRV
    4 ACMS$APPLICATION_SRV_GRP
    4 ACMS$MENU_TASK_TASK
    4 ACMS$MENU_TASK_APPL
2 CDD$CONTAINS_COPY
2 CDD$KEYWORD_GENERALIZATION
2 CDD$PREFERRED_TERM
2 CDD$ASSOCIATED_KEYWORD
```

# B

# Implemented Information Model Diagrams

This appendix contains implemented information model (IIM) diagrams for those protocols shipped with the CDD/Repository exclusive of the generic protocols that implement the basic service models described within *CDD /Repository Information Model Volume I*.

The CDD$HISTORY and CDD$KEYWORD element types are associated with all other element types through the CDD$HISTORY_LIST and CDD$ASSOCIATED_BY_WORD relation types, respectively. Because of space limitations, these element types and relation types do not appear on the IIM diagrams. Figure B–1 lists the IIM diagram conventions.

**Figure B–1 IIM Diagram Conventions**

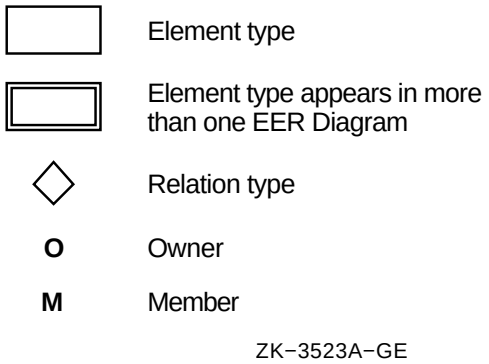


Figure B–2 shows the CDD Element Type Relationship Model.

**Figure B–2 CDD Entity Relationship Model**

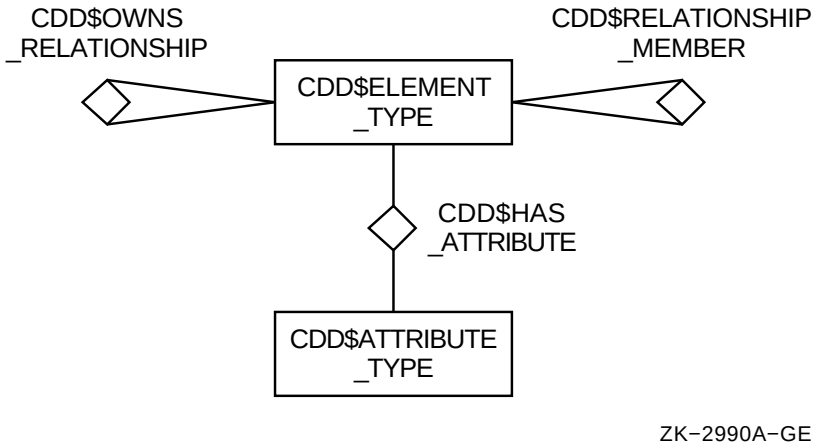


Figure B–3 shows page 1 of 3 for the ACMS protocols.

**Figure B–3 ACMS Protocols (1 of 3)**

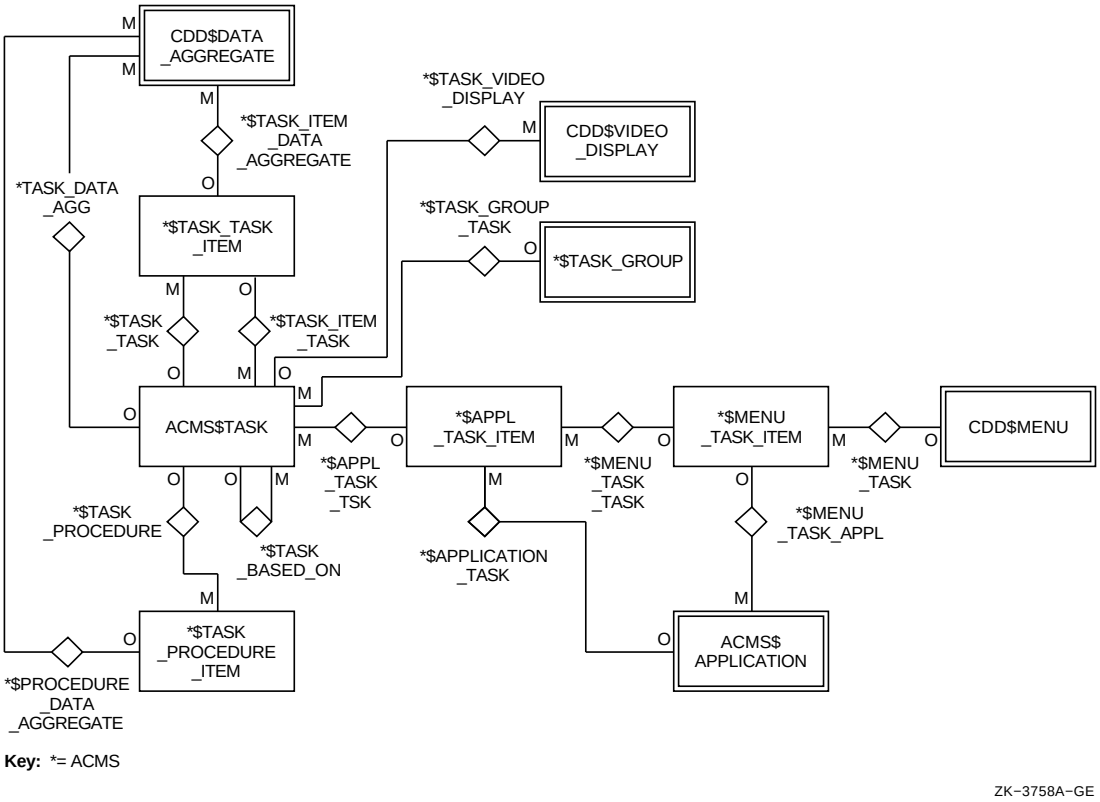


Figure B–4 shows page 2 of 3 for the ACMS protocols.

**Figure B–4 ACMS Protocols (2 of 3)**

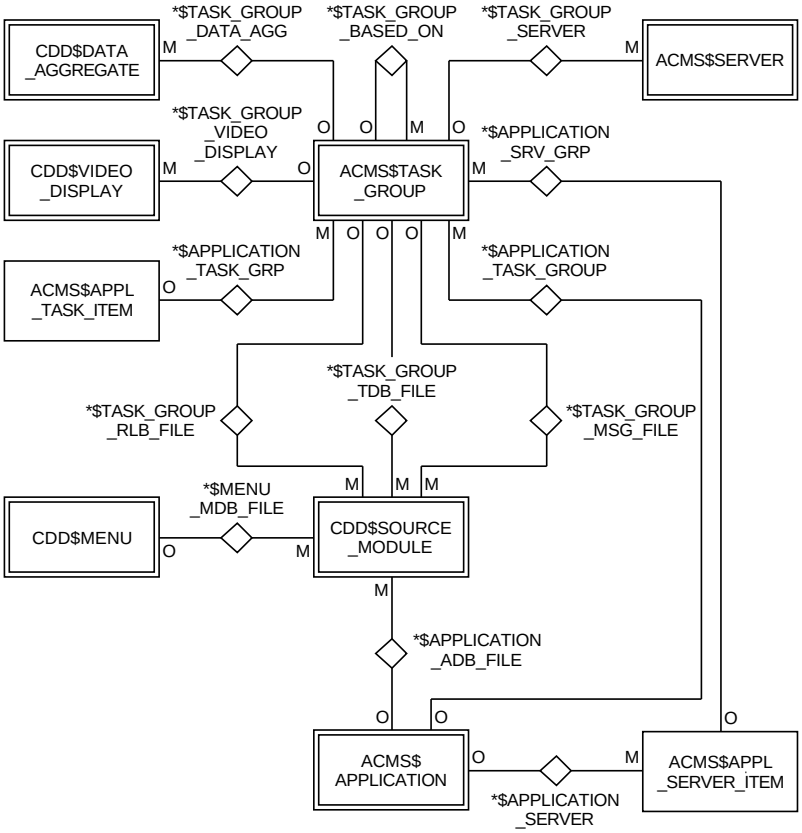


Figure B–5 shows page 3 of 3 for the ACMS protocols.

**Figure B–5 ACMS Protocols (3 of 3)**

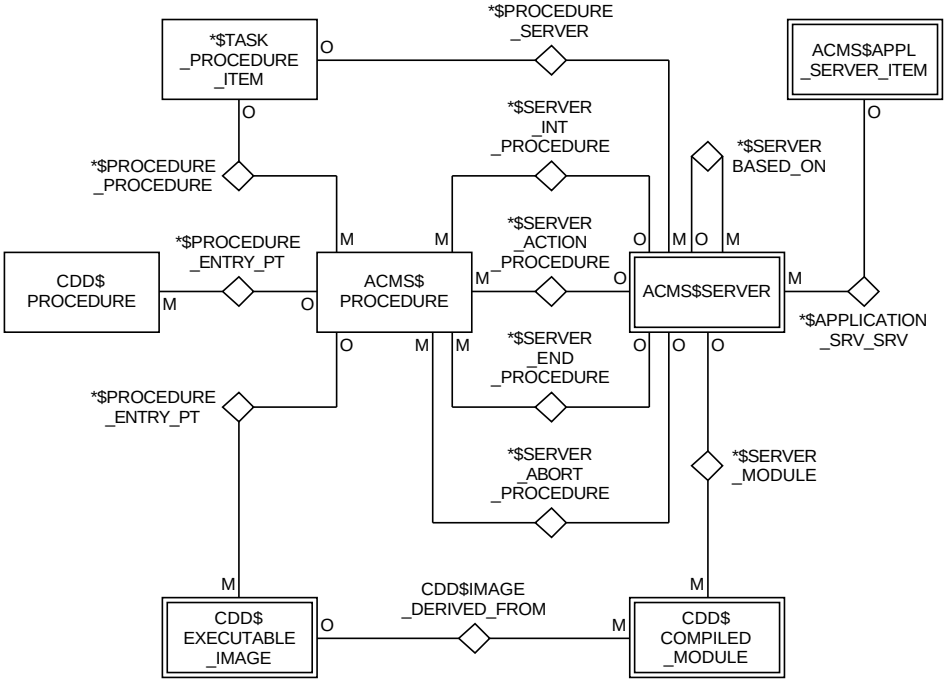


Figure B–6 shows the data element protocols.

**Figure B–6 Data Element Protocols**

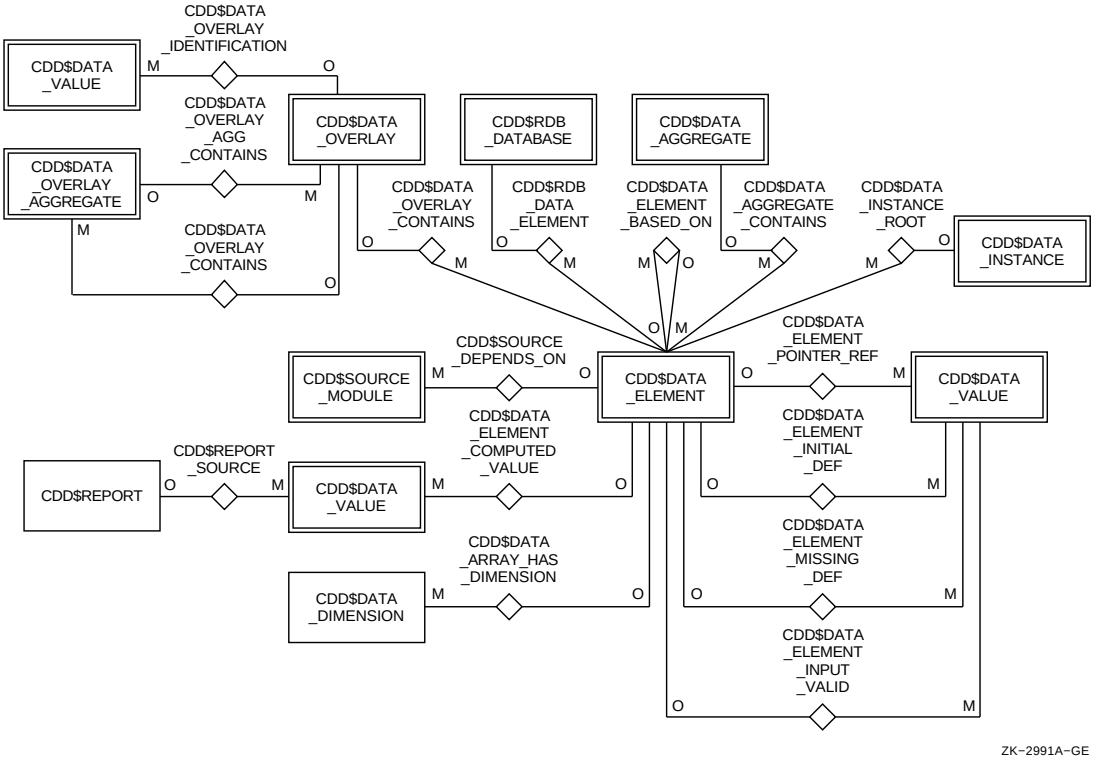


Figure B–7 shows the data aggregate protocols.

**Figure B–7 Data Aggregate Protocols**

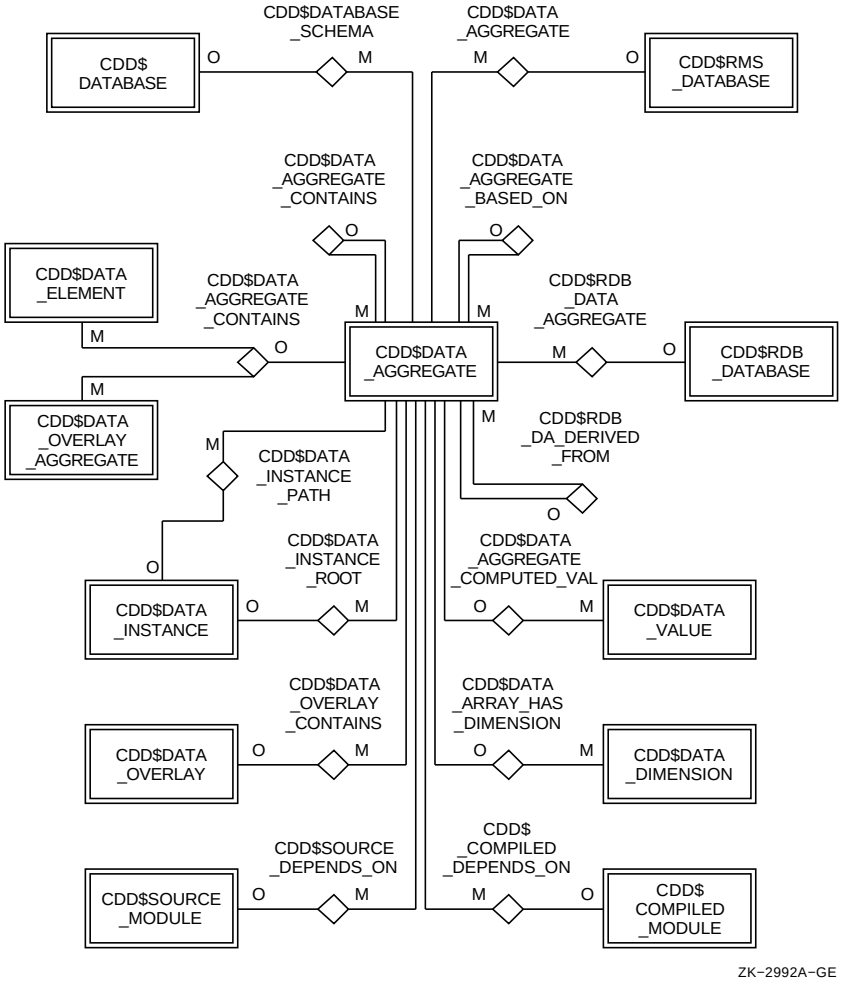


Figure B–8 shows the array description protocols.

**Figure B–8 Array Description Protocols**

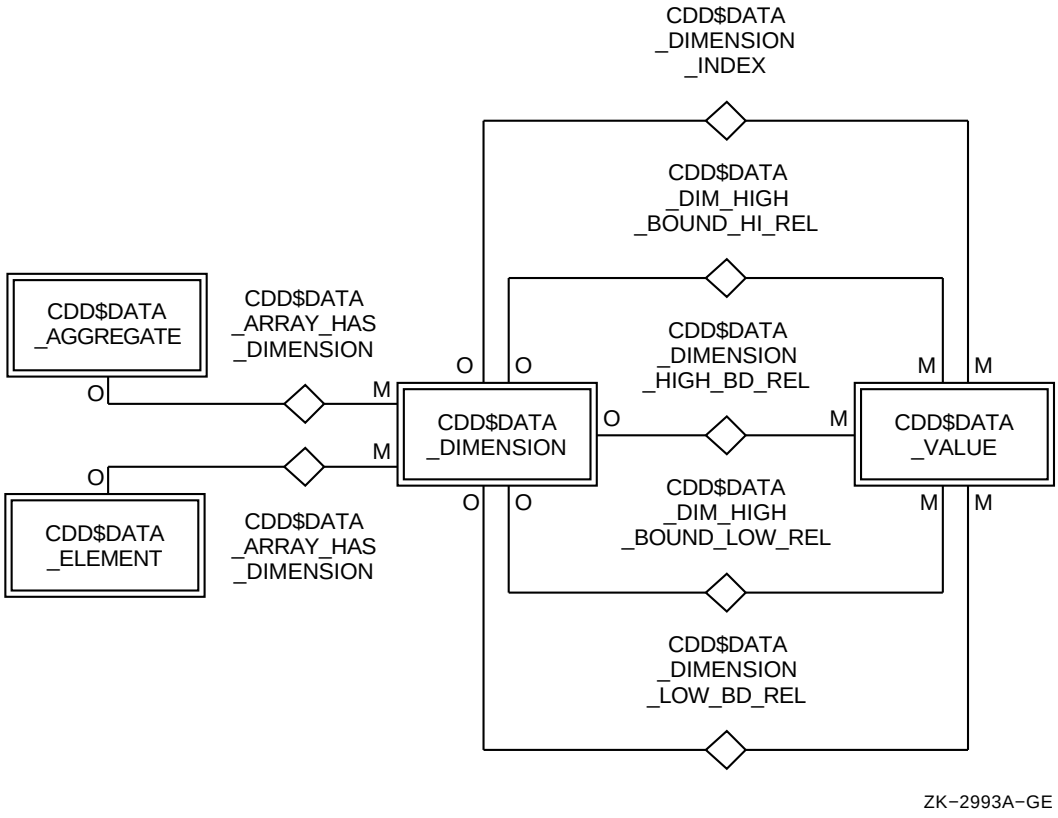


Figure B–9 shows the VAX Datatrieve protocols.

**Figure B–9 Datatrieve Protocols**

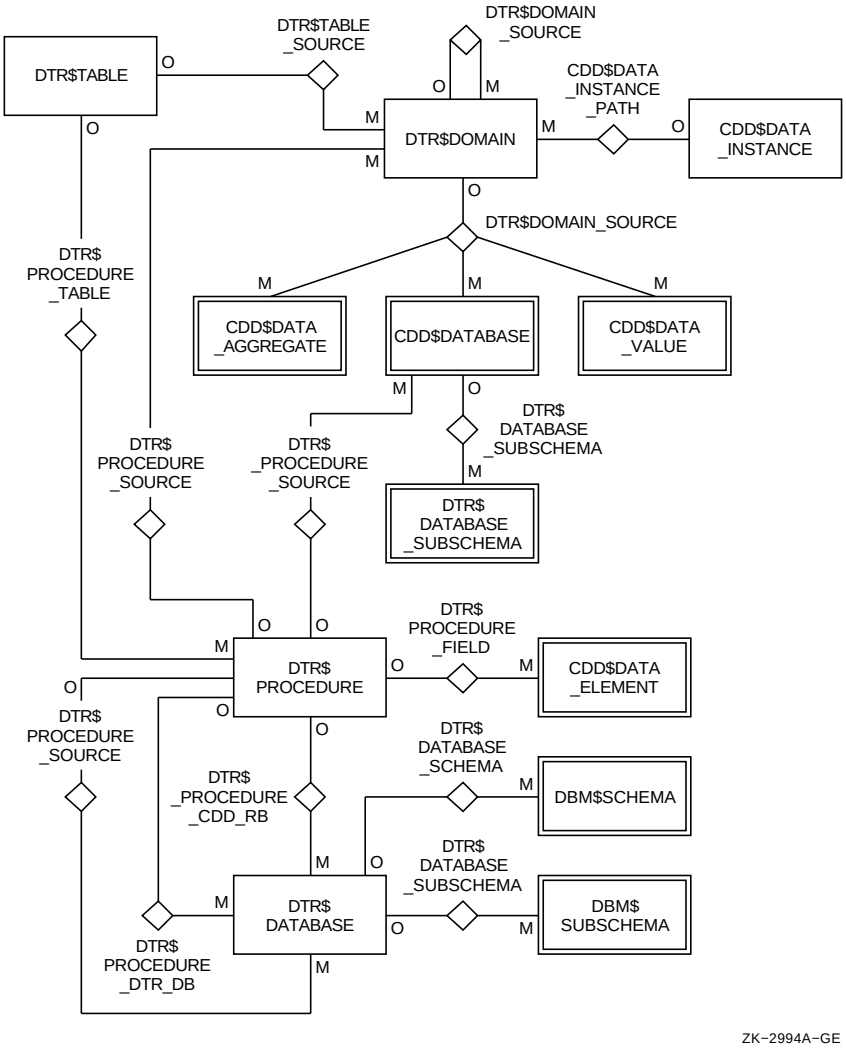


Figure B–10 shows the RALLY protocols.

**Figure B–10 Rally Protocols**

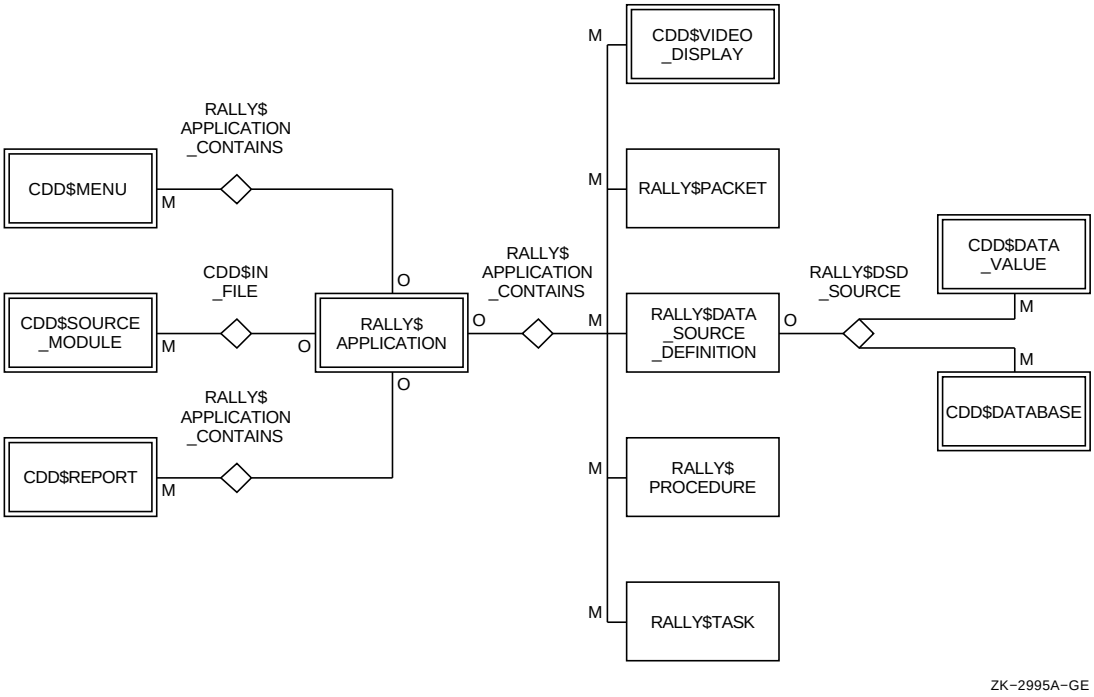


Figure B–11 shows the Rdb protocols.

**Figure B–11 Rdb Protocols**

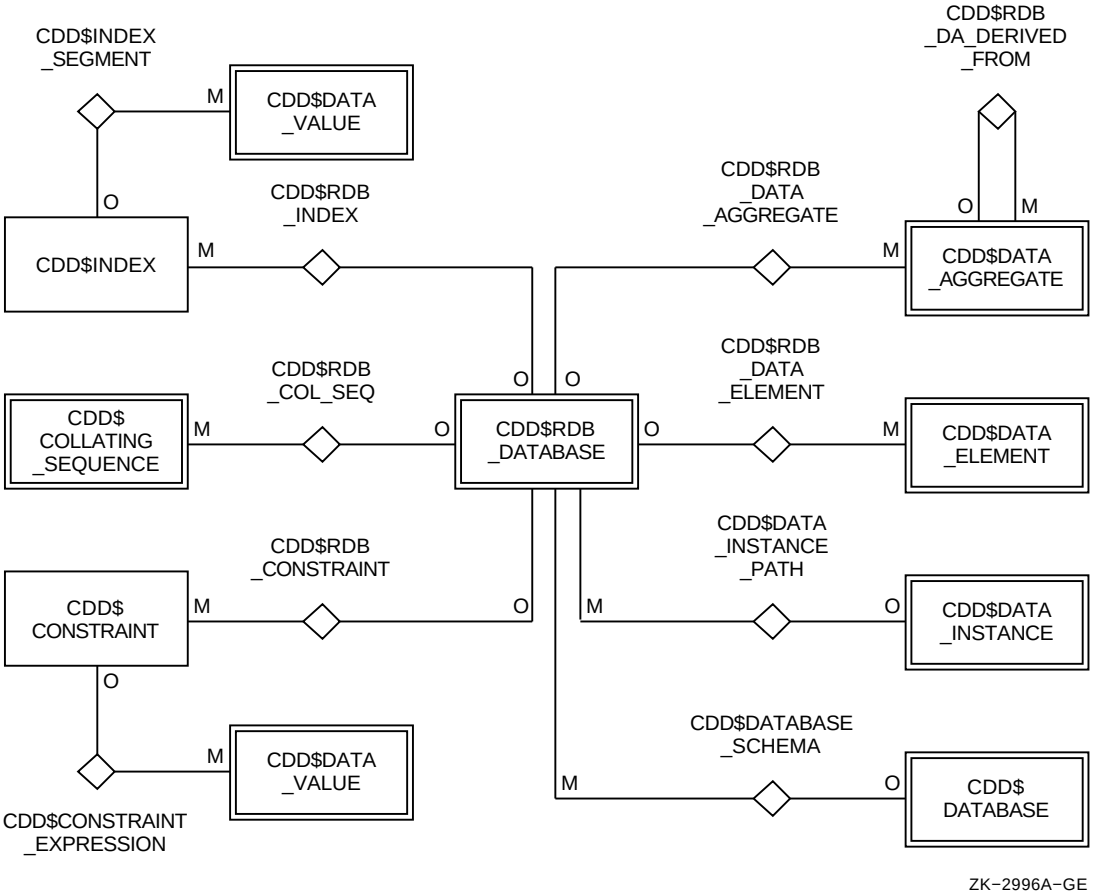


Figure B–12 shows the RMS protocols.

**Figure B–12 RMS Protocols**

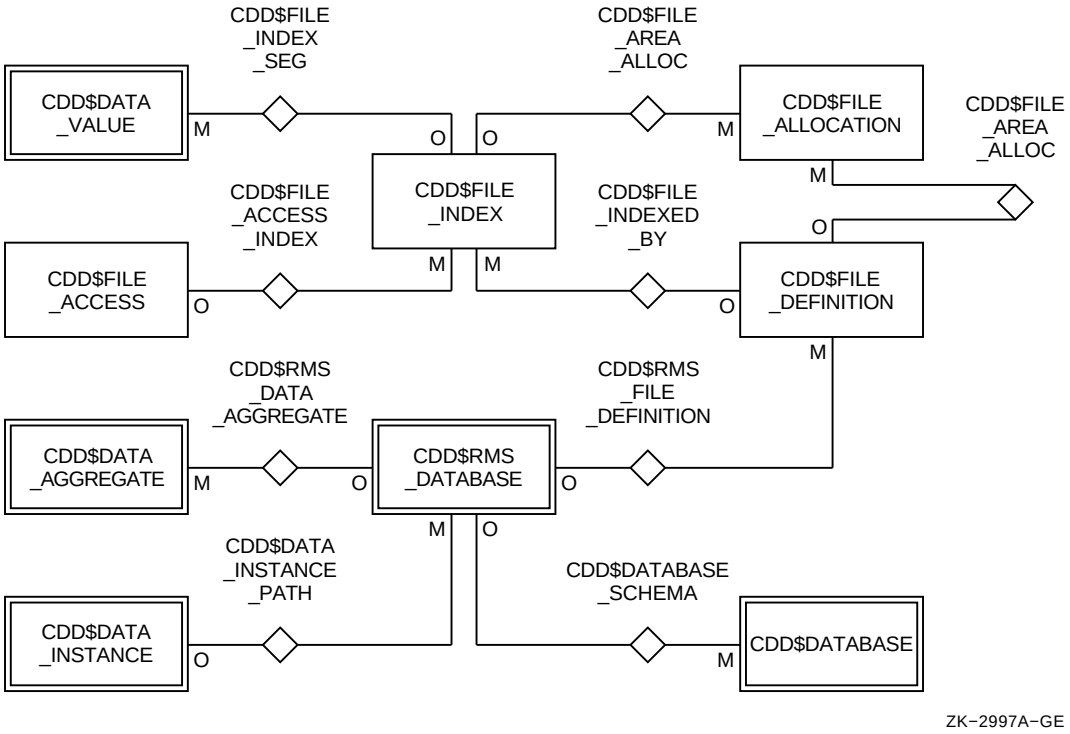


Figure B–13 shows page 1 of 3 for the DBMS protocols.

**Figure B–13 DBM Protocols (1 of 3)**

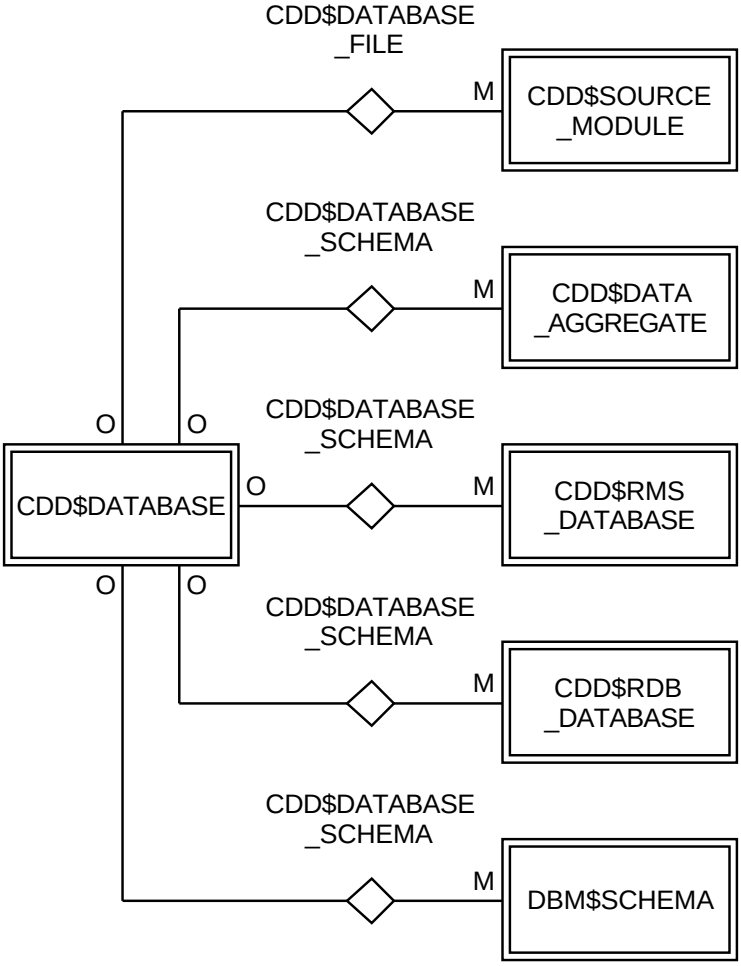


Figure B–14 shows page 2 of 3 for the DBMS protocols.

**Figure B–14 DBM Protocols (2 of 3)**

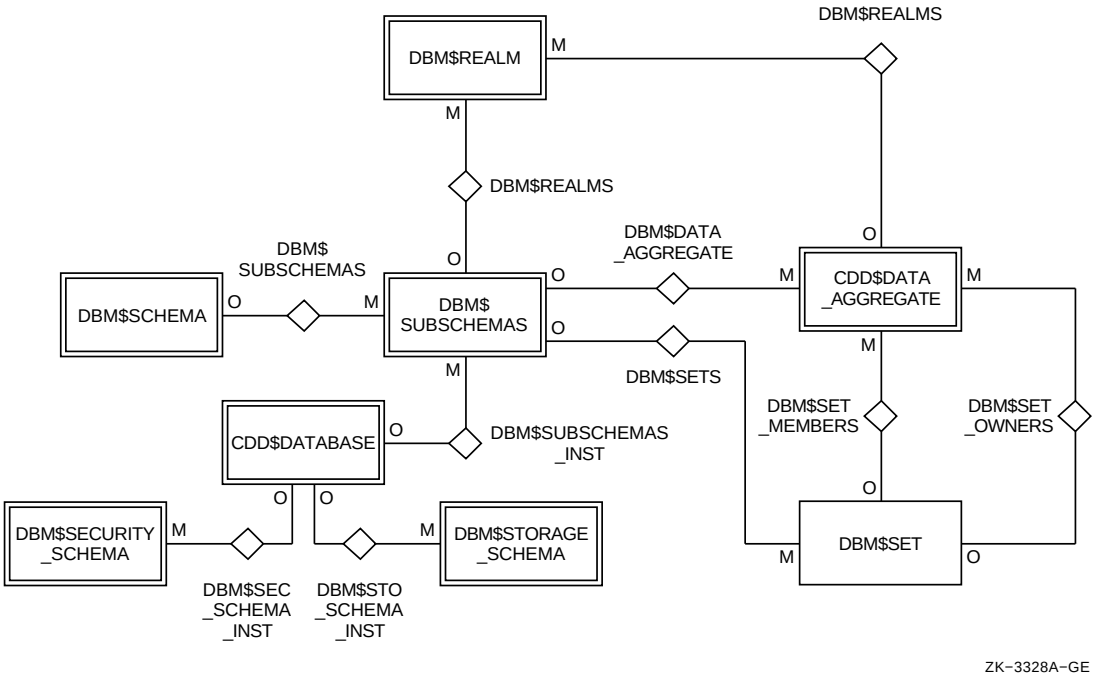


Figure B–15 shows page 3 of 3 for the DBMS protocols.

**Figure B–15 DBM Protocols (3 of 3)**

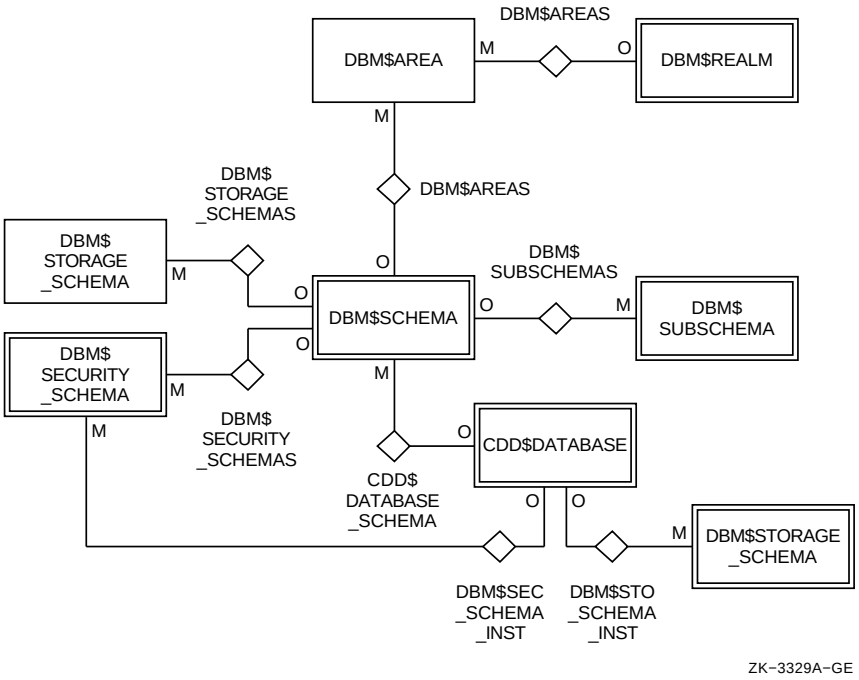


Figure B–16 shows page 1 of 3 for the module tracking protocols.

**Figure B–16 Module Tracking Protocols (1 of 3)**

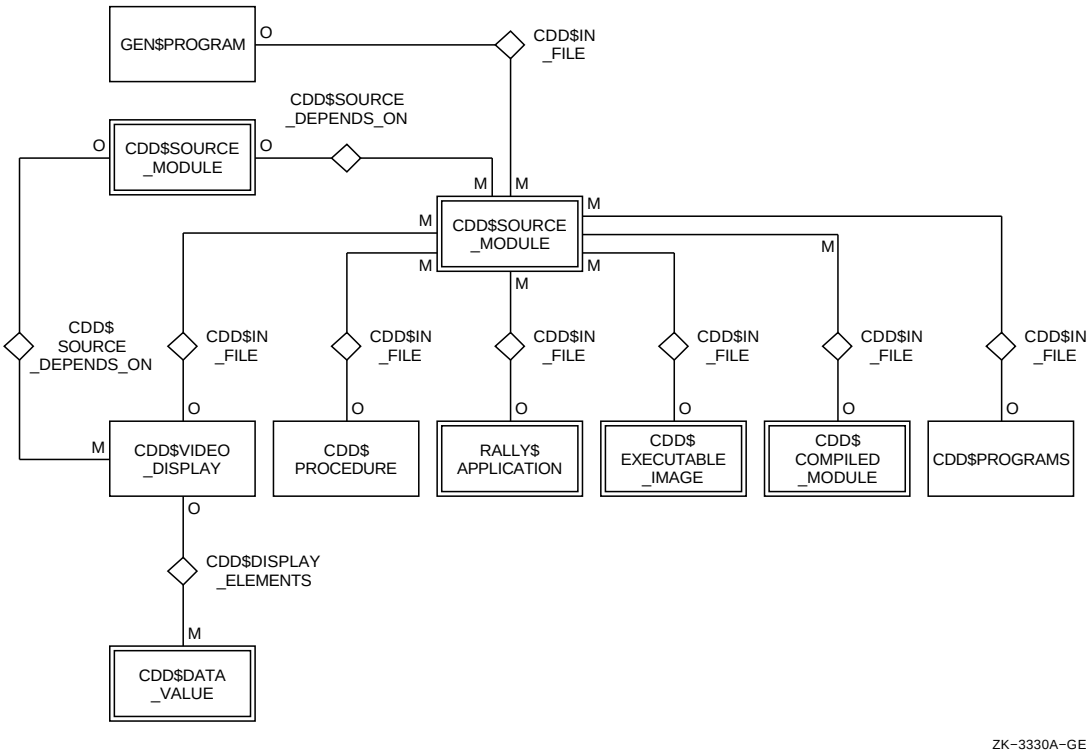


Figure B–17 shows page 2 of 3 for the model tracking protocols.

**Figure B–17 Module Tracking Protocols (2 of 3)**

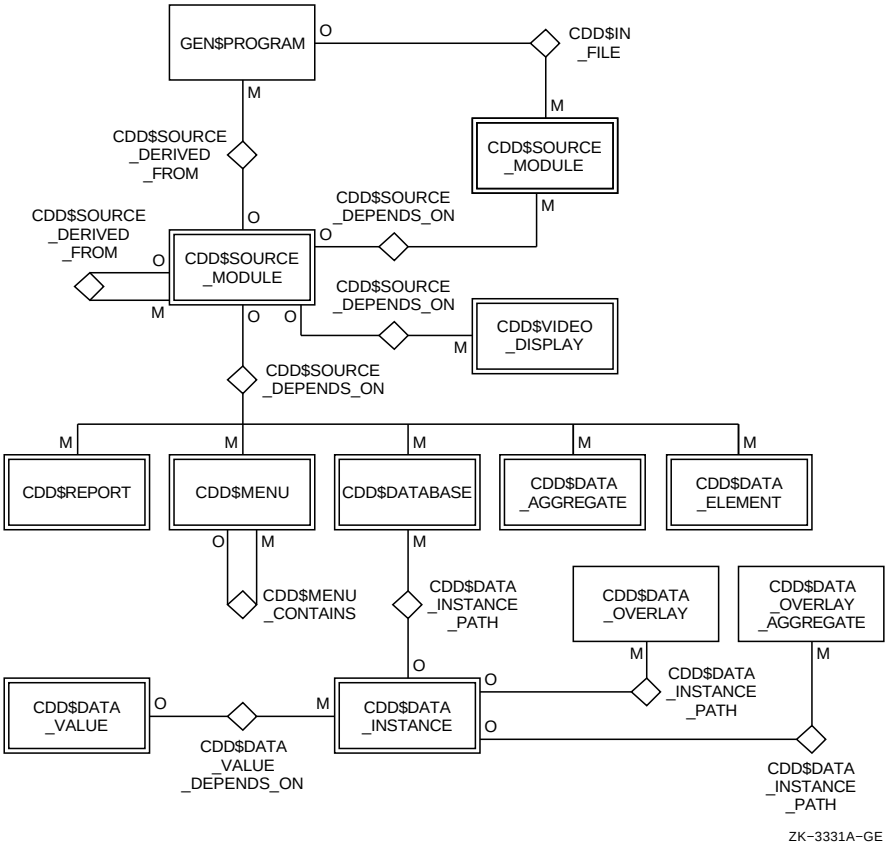


Figure B–18 shows page 3 of 3 for the module tracking protocols.

**Figure B–18 Module Tracking Protocols (3 of 3)**

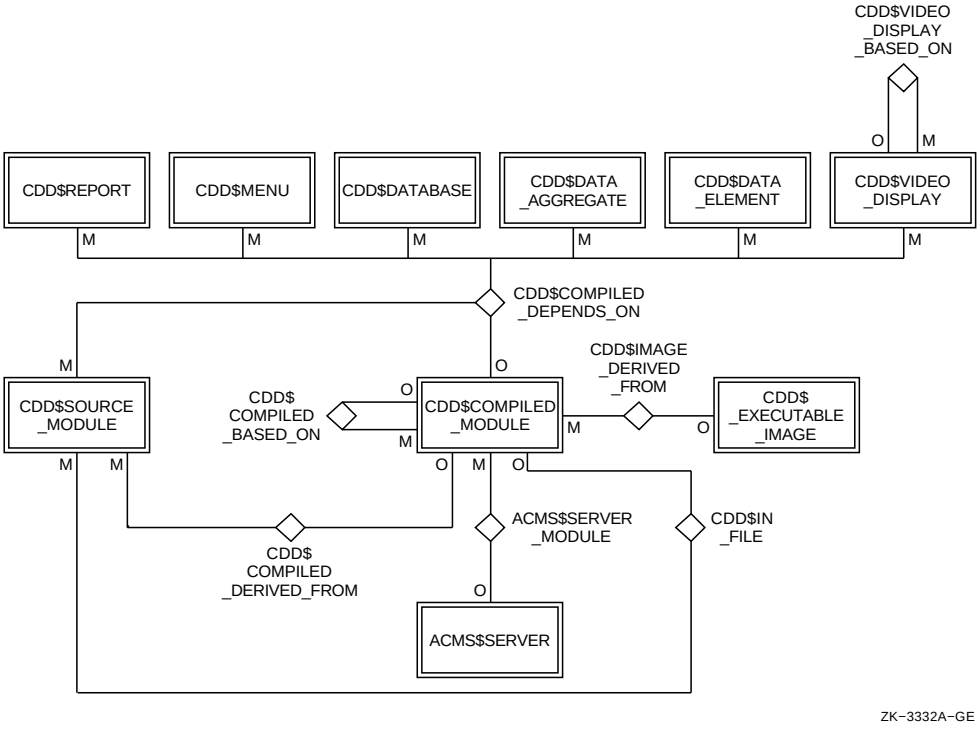
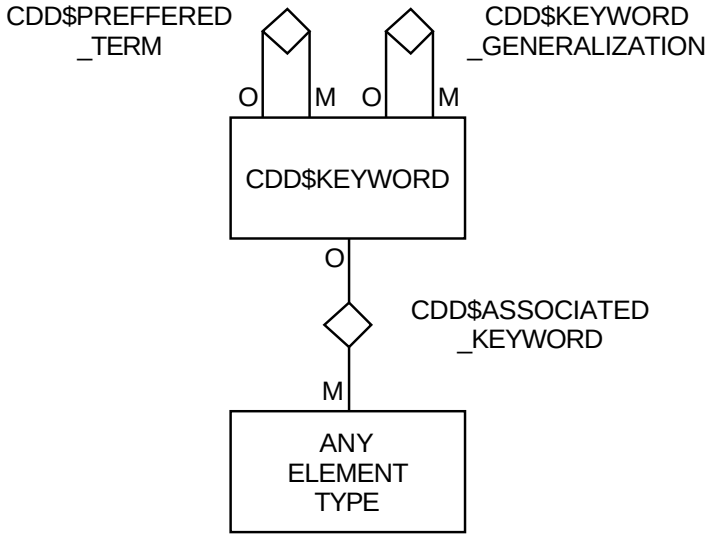


Figure B–19 shows the Keyword Protocols.

**Figure B–19 Keyword Protocols**




ZK–3698A–GE

# Index

## A

## C

## D

## G



## I