

IBM VisualAge TeamConnection



Information Model Reference

Version 2.0

IBM VisualAge TeamConnection



Information Model Reference

Version 2.0

Second Edition (September 1997)

Note

Before using this document, read the general information under "Notices" on page xxvii.

Order publications by phone or fax. The IBM Software Manufacturing Company takes publication orders between 8:30 a.m. and 7:00 p.m. eastern standard time (EST). The phone number is (800) 879-2755. The fax number is (800) 284-4721.

You can also order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation
Attn: Information Development
Department T99B/Building 062
P.O. Box 12195
Research Triangle Park, NC, USA 27709-2195

You can fax comments to (919) 254-0206.

If you have comments about the product, address them to:

IBM Corporation
Attn: Department TH0/Building 062
P.O. Box 12195
Research Triangle Park, NC, USA 27709-2195

You can fax comments to (919) 254-4914.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1992, 1996. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	.xxvii
Trademarks	.xxix
Conventions	.xxxi
Tell us what you think	.xxxiii
Chapter 1. Information Model Overview	1
GLO Global Subject Area.	2
SDD Shareable Data Structure & Element	3
SDD Alias Support For DataElement and DataStructure	6
HLL Global and Local Data Definitions	7
DBD area	9
PSB area	12
HLL Global and Local Data Definitions	14
HLL Include Components.	16
HLL Sub-Structure, Array, and Union	19
HLL Elementary Data Item	20
Extensions for COBOL and PL/I	22
VisualAge Requirements Tool	24
Business Facts	25
Business Information	25
Business Rules	25
Chapter 2. Global Model	27
Object Class: ADAnnotatedObject	27
Attributes.	27
Cache View Methods	27
Object Class: TCPart	27
Attributes.	28
Cache View Methods	30
Object Class: ADAnnotationText	35
Attributes.	36
Cache View Methods	36
Object Class: fdApplicationCollector	36
Attributes.	37
Cache View Methods	37
Relationship Class: fdLinkApplColl2TCPart	37
Attributes.	37
Cache View Methods	38
Relationship Class: ADLinkADAnnotObj2AnnotTxt.	38
Attributes.	38
Cache View Methods	39
Relationship Class: ADLinkTCPartDependsOnTCPart	39
Attributes.	39
Cache View Methods	39
Chapter 3. Data Atlas Dictionary Model.	41
Object Class: DSAtlasObject	41
Attributes.	41
Cache View Methods	41
Object Class: DSAlias	45

Attributes.	46
Cache View Methods	46
Object Class: DSDeviceObject	46
Attributes.	47
Cache View Methods	47
Object Class: DSTechnologyObject	48
Attributes.	48
Cache View Methods	48
Object Class: DSContent	49
Attributes.	50
Cache View Methods	50
Object Class: DSGenericFolder	51
Attributes.	52
Cache View Methods	52
Object Class: DSWorkFolder	52
Attributes.	52
Cache View Methods	53
Object Class: DSMainWindow	53
Attributes.	53
Cache View Methods	53
Object Class: DSRepositoryFolder	53
Attributes.	54
Cache View Methods	54
Object Class: DSProfileFolder	54
Attributes.	55
Cache View Methods	55
Object Class: DSTemplateFolder	55
Attributes.	55
Cache View Methods	55
Object Class: DSQueryFolder	55
Attributes.	56
Cache View Methods	56
Object Class: DSReportFolder	56
Attributes.	57
Cache View Methods	57
Object Class: DSIMSMVSFolder	57
Attributes.	57
Cache View Methods	57
Object Class: DSApplicationMVSFolder	57
Attributes.	58
Cache View Methods	58
Object Class: DSApplicationOS2Folder	58
Attributes.	59
Cache View Methods	59
Object Class: DSIMSOS2Folder	59
Attributes.	59
Cache View Methods	59
Object Class: DSCobolOS2Folder	59
Attributes.	60
Cache View Methods	60
Object Class: DSCobolMVSFolder	60
Attributes.	61
Cache View Methods	61
Object Class: DSPLIOS2Folder	61
Attributes.	61
Cache View Methods	61

Object Class: DSPLIMVFolder	61
Attributes.	62
Cache View Methods	62
Object Class: DSPLIMVSPDS	62
Attributes.	63
Cache View Methods	63
Object Class: DSCobolMVSPDS	63
Attributes.	63
Cache View Methods	63
Object Class: DSApplicationMVSPDS	63
Attributes.	64
Cache View Methods	64
Object Class: DSIMSMVSPDS	64
Attributes.	65
Cache View Methods	65
Object Class: DSPLIOS2Directory	65
Attributes.	65
Cache View Methods	65
Object Class: DSCobolOS2Directory	65
Attributes.	66
Cache View Methods	66
Object Class: DSApplicationOS2Directory.	66
Attributes.	67
Cache View Methods	67
Object Class: DSIMSOS2Directory	67
Attributes.	67
Cache View Methods	67
Object Class: DSCobolOS2File	67
Attributes.	68
Cache View Methods	68
Object Class: DSCobolMVSMember.	68
Attributes.	69
Cache View Methods	69
Object Class: DSPLIOS2File	69
Attributes.	69
Cache View Methods	69
Object Class: DSPLIMVSMember	69
Attributes.	70
Cache View Methods	70
Object Class: DSApplicationOS2File.	70
Attributes.	71
Cache View Methods	71
Object Class: DSApplicationMVSMember	71
Attributes.	71
Cache View Methods	71
Object Class: DSIMSOS2File	71
Attributes.	72
Cache View Methods	72
Object Class: DSIMSMVSMember	72
Attributes.	73
Cache View Methods	73
Object Class: DSVisualQueryFolder	73
Attributes.	73
Cache View Methods	73
Object Class: DSSQLQueryFolder	73
Attributes.	74

Cache View Methods	74
Object Class: DSVisualQueryObject	74
Attributes.	75
Cache View Methods	75
Object Class: DSSQLQueryObject	75
Attributes.	75
Cache View Methods	75
Object Class: DSSQLQueryResultsObject	75
Attributes.	76
Cache View Methods	76
Object Class: DSVisualQueryResultsFolder	76
Attributes.	77
Cache View Methods	77
Object Class: DSDB2SubSystem	77
Attributes.	77
Cache View Methods	77
Relationship Class: DSLinkDevObj2Content	77
Attributes.	78
Cache View Methods	78
Relationship Class: DSLinkContent2AtlasObj	78
Attributes.	79
Cache View Methods	79
 Chapter 4. VisualAge Generator Model	 81
Object Class: VGObject	81
Attributes.	81
Cache View Methods	82
Object Class: VGApplication	82
Attributes.	83
Cache View Methods	83
Object Class: VGDataItem	83
Attributes.	84
Cache View Methods	84
Object Class: VGGUI	85
Attributes.	86
Cache View Methods	86
Object Class: VGProcess.	86
Attributes.	87
Cache View Methods	88
Object Class: VGPSB	88
Attributes.	88
Cache View Methods	89
Object Class: VGRecord	89
Attributes.	90
Cache View Methods	90
Object Class: VGStmtGroup.	91
Attributes.	91
Cache View Methods	92
Object Class: VGTable.	92
Attributes.	92
Cache View Methods	93
Object Class: VGMap	93
Attributes.	94
Cache View Methods	94
Object Class: VGMapGroup.	94
Attributes.	95

Cache View Methods	95
Object Class: VGDataDef	95
Attributes.	96
Cache View Methods	101
Relationship Class: DSLinkSmpIType2VGDatDef	101
Attributes.	102
Cache View Methods	102
Relationship Class: DSLinkVGDatItn2DEAlias	103
Attributes.	103
Cache View Methods	104
Relationship Class: DSLinkVGObject2VGObject	104
Attributes.	105
Cache View Methods	105
 Chapter 5. Conceptual Data Modeler Model	 107
Object Class: DSEntityType	107
Attributes.	107
Cache View Methods	107
Storage View Methods.	107
Constraints	107
Object Class: DSAssociatedEntityType	107
Attributes.	108
Cache View Methods	108
Storage View Methods.	108
Constraints	108
Object Class: DSAttribute	108
Attributes.	109
Cache View Methods	110
Storage View Methods.	110
Constraints	110
Object Class: DSDataElement	110
Attributes.	111
Cache View Methods	111
Storage View Methods.	111
Constraints	111
Object Class: DSDataModel.	111
Attributes.	111
Cache View Methods	111
Storage View Methods.	112
Constraints	112
Object Class: DSAggregate	112
Attributes.	112
Cache View Methods	113
Storage View Methods.	113
Constraints	113
Object Class: DSFundamentalEntityType	113
Attributes.	113
Cache View Methods	114
Storage View Methods.	114
Constraints	114
Object Class: DSRelationshipBundle	114
Attributes.	115
Cache View Methods	117
Storage View Methods.	117
Constraints	117
Object Class: DSRelationshipType	117

Attributes.	118
Cache View Methods	118
Storage View Methods.	118
Constraints	118
Relationship Class: DSEnt_characterizedBy_Attr	118
Attributes.	119
Cache View Methods	119
Storage View Methods.	119
Constraints	119
Relationship Class: DSRel_classOfConceptsIn_Model	119
Attributes.	120
Cache View Methods	120
Storage View Methods.	120
Constraints	120
Relationship Class: DSEnt_classOfThingsIn_Model	121
Attributes.	122
Cache View Methods	122
Storage View Methods.	122
Constraints	122
Relationship Class: DSAggr_classOfDomainsIn_Model	122
Attributes.	123
Cache View Methods	123
Storage View Methods.	123
Constraints	123
Relationship Class: DSEnt_atHomeIn_Aggr	123
Attributes.	124
Cache View Methods	124
Storage View Methods.	124
Constraints	124
Relationship Class: DSRel_atHomeIn_Aggr	124
Attributese	125
Cache View Methods	125
Storage View Methods.	125
Constraints	126
Relationship Class: DSAggr_isTOCfor_Aggr	126
Attributes.	127
Cache View Methods	127
Storage View Methods.	127
Constraints	127
Relationship Class: DSRelBundle_atHomeIn_Aggr	127
Attributes.	128
Cache View Methods	128
Storage View Methods.	128
Constraints	128
Relationship Class: DSKeyPairing	128
Attributes.	129
Cache View Methods	129
Storage View Methods.	129
Constraints	129
Relationship Class: DSRel_enactedBy_KeyPairing	129
Attributes.	130
Cache View Methods	131
Storage View Methods.	131
Constraints	131
Relationship Class: DSRel_ifNeedsTable_AssEnt	131
Attributes.	132

Cache View Methods	132
Storage View Methods.	132
Constraints	132
Relationship Class: DSRelBundle_interrelates_Rel	132
Attributes.	133
Cache View Methods	133
Storage View Methods.	133
Constraints	133
Relationship Class: DSFundEnt_isSourceOf_RelBundle	133
Attributes.	134
Cache View Methods	134
Storage View Methods.	134
Constraints	135
Relationship Class: DSRel_linksTo_Ent	135
Attributes.	136
Cache View Methods	136
Storage View Methods.	136
Constraints	137
Relationship Class: DSAttr_ofType_DataElemn.	137
Attributes.	138
Cache View Methods	138
Storage View Methods.	138
Constraints	138
Chapter 6. Shareable Data Definitions Model	139
Object Class: DSIncludeComponent.	139
Attributes.	139
Cache View Methods	139
Object Class: DSDataDefinition	139
Attributes.	140
Cache View Methods	141
Object Class: DSSimpleTypeDef	141
Attributes.	142
Cache View Methods	146
Object Class: DSStructureDef	150
Attributes.	151
Cache View Methods	152
Object Class: DSSharedDataDefinition	152
Attributes.	153
Cache View Methods	153
Object Class: DSDataElement	153
Attributes.	154
Cache View Methods	155
Object Class: DSDataStructure	156
Attributes.	156
Cache View Methods	157
Object Class: DSDataComponent.	158
Attributes.	159
Cache View Methods	159
Object Class: DSDataltem	159
Attributes.	161
Cache View Methods	168
Object Class: DSSDDAlias	172
Attributes.	172
Cache View Methods	174
Object Class: DSDataElementAlias	174

Attributes.	174
Cache View Methods	175
Object Class: DSDataStructureAlias	175
Attributes.	176
Cache View Methods	176
Relationship Class: DSLinkDataElmn2SmplType	176
Attributes.	177
Cache View Methods	177
Relationship Class: DSLinkDataItm2SDD	177
Attributes.	178
Cache View Methods	178
Relationship Class: DSLinkDataItm2DataDef	179
Attributes.	179
Cache View Methods	180
Relationship Class: DSLinkDataItmLikeItem	180
Attributes.	181
Cache View Methods	181
Relationship Class: DSLinkDataItmRedefinesItem	181
Attributes.	182
Cache View Methods	182
Relationship Class: DSLinkDEAlias2DataElmn	182
Attributes.	183
Cache View Methods	183
Relationship Class: DSLinkDSAlias2DataStrc	183
Attributes.	184
Cache View Methods	184
Relationship Class: DSLinkDataStrc2StrctDef	185
Attributes.	185
Cache View Methods	186
Relationship Class: DSLinkDataItm2SDDAlias	186
Attributes.	187
Cache View Methods	187
Relationship Class: DSLinkDataItm2SDDAlias1	187
Attributes.	188
Cache View Methods	188
Relationship Class: DSLinkDataItm2SDDAlias2	188
Attributes.	189
Cache View Methods	189
Relationship Class: DSLinkDataItm2SDDAlias3	189
Attributes.	190
Cache View Methods	190
Relationship Class: DSLinkSmplTypeSizeVariable	190
Attributes.	191
Cache View Methods	191
Relationship Class: DSLinkStrctDefComponent	191
Attributes.	192
Cache View Methods	192
Relationship Class: DSLinkDataItmBASEDItem	192
Attributes.	193
Cache View Methods	193
 Chapter 7. Information Management System (IMS) Model	 195
Object Class: DSDBD	195
Attributes.	195
Cache View Methods	197
Object Class: DSDBDaccess	200

Attributes.	200
Cache View Methods	202
Object Class: DSDBDdedb	202
Attributes.	203
Cache View Methods	203
Object Class: DSDBDhdam	203
Attributes.	204
Cache View Methods	205
Object Class: DSDBDhidam.	205
Attributes.	205
Cache View Methods	206
Object Class: DSDBDindex	206
Attributes.	206
Cache View Methods	208
Object Class: DSDBDmsdb	209
Attributes.	210
Cache View Methods	210
Object Class: DSsegment	211
Attributes.	211
Cache View Methods	215
Object Class: DSsegmentComplex	216
Attributes.	216
Cache View Methods	219
Object Class: DSsegmentDEDB	219
Attributes.	219
Cache View Methods	220
Object Class: DSsegmentLogical	221
Attributes.	221
Cache View Methods	222
Object Class: DSIMSfield	222
Attributes.	223
Cache View Methods	225
Object Class: DSXDFLD	225
Attributes.	226
Cache View Methods	228
Object Class: DSCompRoutine	229
Attributes.	229
Cache View Methods	230
Object Class: DSExit	230
Attributes.	231
Cache View Methods	233
Object Class: DSDDNAME	233
Attributes.	234
Cache View Methods	235
Object Class: DSArea	235
Attributes.	235
Cache View Methods	237
Object Class: DSIMSdataset	237
Attributes.	238
Cache View Methods	241
Object Class: DSPSB	242
Attributes.	243
Cache View Methods	246
Object Class: DSPCB	247
Attributes.	247
Cache View Methods	248

Object Class: DSPCBtype	249
Attributes.	250
Cache View Methods	250
Object Class: DSPCBbased.	251
Attributes.	251
Cache View Methods	252
Object Class: DSPCBdb	252
Attributes.	253
Cache View Methods	254
Object Class: DSPCBtp	255
Attributes.	256
Cache View Methods	257
Object Class: DSOutputDest	258
Attributes.	258
Cache View Methods	259
Object Class: DSSenSegment	259
Attributes.	260
Cache View Methods	265
Object Class: DSSenField	265
Attributes.	266
Cache View Methods	267
Relationship Class: DSBasedOn	268
Attributes.	269
Cache View Methods	269
Relationship Class: DSCompressedBy	269
Attributes.	270
Cache View Methods	270
Relationship Class: DSContainsSegment	270
Attributes.	271
Cache View Methods	271
Relationship Class: DSDefinedByAccess	271
Attributes.	272
Cache View Methods	272
Relationship Class: DSHasAlternate.	272
Attributes.	273
Cache View Methods	273
Relationship Class: DSHasDuplicateData	273
Attributes.	274
Cache View Methods	274
Relationship Class: DSHasField	274
Attributes.	275
Cache View Methods	275
Relationship Class: DSHasIndex	275
Attributes.	276
Cache View Methods	276
Relationship Class: DSHasIndexSource	277
Attributes.	278
Cache View Methods	278
Relationship Class: DSLChild	278
Attributes.	279
Cache View Methods	280
Relationship Class: DSHasOverflow.	281
Attributes.	282
Cache View Methods	282
Relationship Class: DSHasPair	282
Attributes.	283

Cache View Methods	283
Relationship Class: DSHasParentSenseg	283
Attributes.	284
Cache View Methods	284
Relationship Class: DSParent	284
Attributes.	285
Cache View Methods	286
Relationship Class: DSIndex	286
Attributes.	287
Cache View Methods	287
Relationship Class: DSHasRootSequence	287
Attributes.	288
Cache View Methods	288
Relationship Class: DSHasSearch	288
Attributes.	289
Cache View Methods	289
Relationship Class: DSHasSensitive.	289
Attributes.	290
Cache View Methods	290
Relationship Class: DSSource	291
Attributes.	292
Cache View Methods	292
Relationship Class: DSHasSubsequence	292
Attributes.	293
Cache View Methods	293
Relationship Class: DSDSColl	293
Attributes.	294
Cache View Methods	295
Relationship Class: DSHasArea	296
Attributes.	297
Cache View Methods	297
Relationship Class: DSHasDataset	298
Attributes.	299
Cache View Methods	299
Relationship Class: DSHasIMSfldName	299
Attributes.	300
Cache View Methods	300
Relationship Class: DSHasIMSsegmName	300
Attributes.	301
Cache View Methods	301
Relationship Class: DSIncludesPCB.	301
Attributes.	302
Cache View Methods	302
Relationship Class: DSIndexedBy	302
Attributes.	303
Cache View Methods	303
Relationship Class: DSIsPartOf	303
Attributes.	304
Cache View Methods	304
Relationship Class: DSIsUseOf	304
Attributes.	305
Cache View Methods	305
Relationship Class: DSMakesUseOf.	305
Attributes.	306
Cache View Methods	306
Relationship Class: DSMappedBy	306

Attributes.	307
Cache View Methods	307
Relationship Class: DSMapsTo	308
Attributes.	309
Cache View Methods	309
Relationship Class: DSRefersTo	309
Attributes.	310
Cache View Methods	310
Relationship Class: DSSequencedBy	310
Attributes.	311
Cache View Methods	311
Relationship Class: DSSharesWith	311
Attributes.	312
Cache View Methods	312
Relationship Class: DSSpecifiedByType	312
Attributes.	313
Cache View Methods	313
Relationship Class: DSUsedAsSENFLD	313
Attributes.	314
Cache View Methods	314
Relationship Class: DSUsesSENFLD	314
Attributes.	315
Cache View Methods	315
Relationship Class: DSUsesDBDexit	315
Attributes.	316
Cache View Methods	316
Relationship Class: DSUsesSEGMexit	317
Attributes.	318
Cache View Methods	318
Relationship Class: DSSharesUseOf	318
Attributes.	319
Cache View Methods	319
Chapter 8. High Level Languages Model	321
Object Class: DSArray.	321
Attributes.	321
Cache View Methods	321
Object Class: DSUnion	321
Attributes.	322
Cache View Methods	322
Object Class: DSEnumeratedList	322
Attributes.	323
Cache View Methods	323
Object Class: DSPointer	323
Attributes.	324
Cache View Methods	324
Object Class: DSArrayDef	324
Attributes.	325
Cache View Methods	328
Object Class: DSUnionDef	329
Attributes.	330
Cache View Methods	330
Object Class: DSEnumerationDef.	330
Attributes.	331
Cache View Methods	332
Object Class: DSPointerDef.	332

Attributes.	333
Cache View Methods	333
Object Class: DSIncludedSourceDef.	334
Attributes.	334
Cache View Methods	335
Object Class: DSIncludeStatement	337
Attributes.	337
Cache View Methods	338
Object Class: DSCobolDataDefinition	338
Attributes.	339
Cache View Methods	340
Object Class: DSCobolSpecialDataName	340
Attributes.	341
Cache View Methods	342
Object Class: DSPLIDataDefinition	342
Attributes.	343
Cache View Methods	345
Object Class: DSPLIArea.	345
Attributes.	346
Cache View Methods	346
Object Class: DSOOffset	347
Attributes.	347
Cache View Methods	347
Object Class: DSFunctionDeclaration	348
Attributes.	348
Cache View Methods	349
Object Class: DSPLIEntryDataDCL	349
Attributes.	350
Cache View Methods	352
Object Class: DSTypeDefReference	352
Attributes.	352
Cache View Methods	353
Object Class: DSCFunctionPrototype	353
Attributes.	354
Cache View Methods	354
Relationship Class: DSLinkArray2ArrayDef	354
Attributes.	355
Cache View Methods	355
Relationship Class: DSLinkUnion2UnionDef	355
Attributes.	356
Cache View Methods	356
Relationship Class: DSLinkEnumLst2EnumDef	356
Attributes.	357
Cache View Methods	357
Relationship Class: DSLinkPointer2PtrDef	357
Attributes.	358
Cache View Methods	358
Relationship Class: DSLinkIncSrcDef2InclDcmp	358
Attributes.	360
Cache View Methods	360
Relationship Class: DSLinkCobolRenamesThru	360
Attributes.	361
Cache View Methods	361
Relationship Class: DSLinkArrayDef2DataDef	361
Attributes.	362
Cache View Methods	362

Relationship Class: DSLinkArrayDef2SDD	362
Attributes.	363
Cache View Methods	363
Relationship Class: DSLinkArrDef2TDRRef	363
Attributes.	364
Cache View Methods	364
Relationship Class: DSLinkArrayDefKey	364
Attributes.	365
Cache View Methods	365
Relationship Class: DSLinkCobolLevel88	366
Attributes.	366
Cache View Methods	367
Relationship Class: DSLinkUnionDefMember	367
Attributes.	368
Cache View Methods	368
Relationship Class: DSLinkFunctionParameterType	368
Attributes.	369
Cache View Methods	369
Relationship Class: DSLinkOffsetArea	369
Attributes.	370
Cache View Methods	370
Relationship Class: DSLinkArrayDefLBVariable	370
Attributes.	371
Cache View Methods	371
Relationship Class: DSLinkPtrDef2DataDef	371
Attributes.	372
Cache View Methods	372
Relationship Class: DSLinkPtrDef2TDRRef	372
Attributes.	373
Cache View Methods	373
Relationship Class: DSLinkInclDStm2InclDsrc	373
Attributes.	374
Cache View Methods	374
Relationship Class: DSLinkDataItm2CblDef	374
Attributes.	375
Cache View Methods	375
Relationship Class: DSLinkDataItm2PLIDtDfn	376
Attributes.	376
Cache View Methods	377
Relationship Class: DSLinkFnctnDcl2PLIEntry	377
Attributes.	378
Cache View Methods	378
Relationship Class: DSLinkFunctionReturnType	378
Attributes.	379
Cache View Methods	379
Relationship Class: DSLinkFnctnDcl2CFnPrtp	379
Attributes.	380
Cache View Methods	380
Relationship Class: DSLinkAreaSizeVariable	380
Attributes.	381
Cache View Methods	381
Relationship Class: DSLinkCobolRenames	381
Attributes.	382
Cache View Methods	382
Relationship Class: DSLinkTDRRef2SDDAlias2	382
Attributes.	383

Cache View Methods	383
Relationship Class: DSLinkTDRRef2SDDAlias1	383
Attributes.	384
Cache View Methods	384
Relationship Class: DSLinkTDRRef2SDD	384
Attributes.	385
Cache View Methods	385
Relationship Class: DSLinkArrayDefUBVariable	385
Attributes.	386
Cache View Methods	386
Relationship Class: DSLinkCobolLevel66DataItm	387
Attributes.	387
Cache View Methods	388
 Chapter 9. Relational Database: Common Submodel	 389
Object Class: DSRNamedRelation	389
Attributes.	389
Cache View Methods	390
Object Class: DSRAlternateName	390
Attributes.	391
Cache View Methods	392
Object Class: DSRCheckConstraint	392
Attributes.	393
Cache View Methods	393
Object Class: DSRColumnDefinition	393
Attributes.	394
Cache View Methods	398
Object Class: DSRColumnType	398
Attributes.	399
Cache View Methods	400
Object Class: DSRDatabase	400
Attributes.	400
Cache View Methods	401
Object Class: DSRIndex	401
Attributes.	402
Cache View Methods	403
Object Class: DSRKeyDefinition	403
Attributes.	404
Cache View Methods	405
Object Class: DSRPersistentRelation	405
Attributes.	406
Cache View Methods	406
Object Class: DSRPersistentRelDef	406
Attributes.	407
Cache View Methods	408
Object Class: DSRPhysicalDesign	408
Attributes.	408
Cache View Methods	409
Object Class: DSRRCollection	409
Attributes.	410
Cache View Methods	410
Object Class: DSRRDBSystem	411
Attributes.	411
Cache View Methods	413
Object Class: DSRRelationalDesign	413
Attributes.	414

Cache View Methods	414
Object Class: DSRTTable	414
Attributes.	415
Cache View Methods	416
Object Class: DSRTTableDefinition.	417
Attributes.	418
Cache View Methods	419
Object Class: DSRTTableMap	419
Attributes.	420
Cache View Methods	420
Object Class: DSRUniqueConstraint.	420
Attributes.	421
Cache View Methods	421
Object Class: DSRView	421
Attributes.	422
Cache View Methods	422
Object Class: DSRViewDefinition	422
Attributes.	423
Cache View Methods	424
Relationship Class: DSRLinkPRelHasCDef	424
Attributes.	425
Cache View Methods	425
Relationship Class: DSRLinkPRelDefInRDesign	426
Attributes.	426
Cache View Methods	427
Relationship Class: DSRColumnNameAlias	427
Attributes.	428
Cache View Methods	428
Relationship Class: DSRLinkHasCConstraint	428
Attributes.	429
Cache View Methods	429
Relationship Class: DSRLinkHasColumns.	429
Attributes.	430
Cache View Methods	430
Relationship Class: DSRLinkHasColumnType	430
Attributes.	431
Cache View Methods	431
Relationship Class: DSRLinkDBaseInRDBSys	431
Attributes.	432
Cache View Methods	432
Relationship Class: DSRLinkIndexInRCollection	432
Attributes.	433
Cache View Methods	433
Relationship Class: DSRLinkHasKey	433
Attributes.	434
Cache View Methods	435
Relationship Class: DSRLinkNRelationInRColl	435
Attributes.	436
Cache View Methods	436
Relationship Class: DSRLinkRCollectionInRDBSys	436
Attributes.	437
Cache View Methods	437
Relationship Class: DSRLinkMapsConstraints	437
Attributes.	438
Cache View Methods	439
Relationship Class: DSRLinkHasTDefinition	439

Attributes.	440
Cache View Methods	440
Relationship Class: DSRLinkHasUConstraint	440
Attributes.	441
Cache View Methods	441
Relationship Class: DSRLinkHasVDefinition	441
Attributes.	442
Cache View Methods	442
Relationship Class: DSRLinkIncludesCConstraint	442
Attributes.	443
Cache View Methods	443
Relationship Class: DSRLinkIncludesUConstraint	443
Attributes.	444
Cache View Methods	444
Relationship Class: DSRLinkIsPrimaryKey	444
Attributes.	445
Cache View Methods	445
Relationship Class: DSRLinkKeyColumns.	445
Attributes.	446
Cache View Methods	446
Relationship Class: DSRLocalType	447
Attributes.	448
Cache View Methods	448
Relationship Class: DSRLinkAltNameInPDesign	448
Attributes.	449
Cache View Methods	449
Relationship Class: DSRLinkRConstrainedBy	449
Attributes.	450
Cache View Methods	451
Relationship Class: DSRLinkPDesignInRDesign	452
Attributes.	453
Cache View Methods	453
Relationship Class: DSRLinkReferencesCD	453
Attributes.	454
Cache View Methods	454
Relationship Class: DSRLinkResolvesInPRD	454
Attributes.	455
Cache View Methods	455
Relationship Class: DSRLinkParentTable	455
Attributes.	456
Cache View Methods	456
Relationship Class: DSRLinkSynonymOrAliasFor	456
Attributes.	457
Cache View Methods	457
Relationship: DSRLinkIsChildTable	457
Attributes.	458
Cache View Methods	458
Relationship Class: DSRLinkIsChildTable	458
Attributes.	459
Cache View Methods	459
Relationship Class: DSRLinkUsesKey	459
Attributes.	460
Cache View Methods	460
Chapter 10. Relational Database: MVS Submodel.	461
Object Class: DSRMRDBufferpool	461

Attributes.	461
Cache View Methods	463
Object Class: DSRMRDColumnDefExt	463
Attributes.	463
Cache View Methods	466
Object Class: DSRMRDDatabase.	466
Attributes.	466
Cache View Methods	469
Object Class: DSRMRDICFCatalog	469
Attributes.	469
Cache View Methods	470
Object Class: DSRMRDIndex	471
Attributes.	471
Cache View Methods	474
Object Class: DSRMRDPartition	474
Attributes.	475
Cache View Methods	477
Object Class: DSRMRDIndexPartition	477
Attributes.	478
Cache View Methods	479
Object Class: DSRMRDSpace	479
Attributes.	479
Cache View Methods	481
Object Class: DSRMRDIndexspace	482
Attributes.	482
Cache View Methods	483
Object Class: DSRMRDPhysicalDesign	483
Attributes.	484
Cache View Methods	485
Object Class: DSRMRDStoragegroup	485
Attributes.	486
Cache View Methods	489
Object Class: DSRMRDTable	489
Attributes.	489
Cache View Methods	493
Object Class: DSRMRDTablespace	493
Attributes.	494
Cache View Methods	498
Object Class: DSRMRDTSPartition	498
Attributes.	498
Cache View Methods	499
Object Class: DSRMRDView	500
Attributes.	500
Cache View Methods	501
Object Class: DSRMRDVolume	501
Attributes.	501
Cache View Methods	502
Relationship Class: DSRContainedInMRDD	502
Attributes.	503
Cache View Methods	503
Relationship Class: DSRLinkContainsPartition	503
Attributes.	504
Cache View Methods	504
Relationship Class: DSRLinkMRDBPoolInRDBSys	504
Attributes.	505
Cache View Methods	505

Relationship Class: DSRLinkClusteredISForMTB	505
Attributes.	505
Methods	506
Relationships	506
Relationship: DSRLinkHasPartitions	506
Attributes.	507
Cache View Methods	507
Relationship Class: DSRLinkHasPartitions	507
Attributes.	508
Cache View Methods	508
Relationship Class: DSRLinkMRDStgrInRDBSys.	508
Attributes.	509
Cache View Methods	509
Relationship Class: DSRLinkTSInMRDDatabase	509
Attributes.	510
Cache View Methods	510
Relationship Class: DSRLInMRDDatabase	510
Attributes.	511
Cache View Methods	511
Relationship Class: DSRLinkForMRDTable	511
Attributes.	512
Cache View Methods	512
Relationship Class: DSRLinkMRDStoredIn	512
Attributes.	513
Cache View Methods	513
Relationship Class: DSRLinkMRDDbaseInRColl	513
Attributes.	514
Cache View Methods	514
Relationship Class: DSRLinkMRDStgrInRColl	514
Attributes.	515
Cache View Methods	515
Relationship Class: DSRLinkMRDTblspInRColl	515
Attributes.	516
Cache View Methods	516
Relationship Class: DSRLinkPartitionsTS	517
Attributes.	517
Cache View Methods	518
Relationship Class: DSRLinkRefersBuffer	518
Attributes.	519
Cache View Methods	519
Relationship Class: DSRLinkRefersBufferpool	519
Attributes.	520
Cache View Methods	520
Relationship Class: DSRLinkRefersCatalog	520
Attributes.	521
Cache View Methods	521
Relationship Class: DSRLinkRefersStogroup	521
Attributes.	522
Cache View Methods	522
Relationship Class: DSRLinkRefinedByMCDE	523
Attributes.	524
Cache View Methods	524
Relationship Class: DSRLinkROShare	524
Attributes.	525
Cache View Methods	525
Relationship Class: DSRLinkSpaceRefersStogroup	525

Attributes.	526
Cache View Methods	527
Relationship Class: DSRLinkSpaceUsesCatalog	527
Attributes.	528
Cache View Methods	528
Relationship Class: DSRLinkUsesBufferpool.	528
Attributes.	529
Cache View Methods	529
Relationship Class: DSRLinkUsesCatalog.	530
Attributes.	530
Cache View Methods	531
Relationship Class: DSRLinkUsesDatabase	531
Attributes.	532
Cache View Methods	532
Relationship Class: DSRLinkUsesStogroup	532
Attributes.	533
Cache View Methods	533
Relationship Class: DSRLinkUsesTablespace	533
Attributes.	534
Cache View Methods	534
Relationship Class: DSRLinkUsesVolume.	534
Attributes.	535
Cache View Methods	535
Relationship Class: DSRLinkTableInMPDesign	535
Attributes.	536
Cache View Methods	536
Relationship Class: DSRLinkTblspaceInMPDesign	536
Attributes.	537
Cache View Methods	537
Relationship Class: DSRLinkStogroupInMPDesign	537
Attributes.	538
Cache View Methods	538
Relationship Class: DSRLinkViewInMPDesign	538
Attributes.	539
Cache View Methods	539
Relationship Class: DSRLinkIndexInMPDesign	539
Attributes.	540
Cache View Methods	540
Relationship Class: DSRLinkDatabaseInMPDesign	540
Attributes.	541
Cache View Methods	541
Relationship Class: DSRLinkBufpoolInMPDesign	542
Attributes.	542
Cache View Methods	543
Chapter 11. Relational Database: Common Server Submodel	545
Object Class: DSRORDDatabase.	545
Attributes.	545
Cache View Methods	546
Object Class: DSRORDIndex	546
Attributes.	547
Cache View Methods	547
Object Class: DSRORDPhysicalDesign	547
Attributes.	548
Cache View Methods	549
Object Class: DSRORDTable	549

Attributes.	549
Cache View Methods	550
Object Class: DSRORDView	551
Attributes.	551
Cache View Methods	551
Object Class: DSRWRDColumnDefExt.	552
Attributes.	552
Cache View Methods	553
Object Class: DSORDTablespace.	553
Attributes.	553
Cache View Methods	555
Relationship Class: DSRLinkIndexHasORDD	555
Attributes.	556
Cache View Methods	556
Relationship Class: DSRLinkTableHasORDD	556
Attributes.	557
Cache View Methods	557
Relationship Class: DSRLinkViewHasORDD.	557
Attributes.	558
Cache View Methods	558
Relationship Class: DSRLinkForORDTable	558
Attributes.	559
Cache View Methods	559
Relationship Class: DSRLinkRefinedByWCDE	559
Attributes.	560
Cache View Methods	560
Relationship Class: DSRLinkTableInOPDesign	561
Attributes.	562
Cache View Methods	562
Relationship Class: DSRLinkViewInOPDesign	562
Attributes.	563
Cache View Methods	563
Relationship Class: DSRLinkIndexInOPDesign	563
Attributes.	564
Cache View Methods	564
Relationship Class: DSRLinkDatabaseInOPDesign	564
Attributes.	565
Cache View Methods	565
Relationship Class: DSDSRLinkORDTblInTblspace	565
Attributes.	566
Cache View Methods	566
Relationship Class: DSDSRLinkORDTblIndexInTSpace.	566
Attributes.	567
Cache View Methods	567
Relationship Class: DSDSRLinkORDTblLongInTSpace	567
Attributes.	568
Cache View Methods	568
Relationship Class: DSDSRLinkTblspaceInOPDesign	568
Attributes.	569
Cache View Methods	569
Relationship Class: DSDSRLinkORDTblspaceInRDBSys	569
Attributes.	570
Cache View Methods	570
Chapter 12. Relational Database: Oracle Submodel	571
Object Class: DSOacleIndex	571

Attributes.	571
Cache View Methods	572
Object Class: DSOaclePhysicalDesign	572
Attributes.	572
Cache View Methods	573
Object Class: DSOacleTable	573
Attributes.	574
Cache View Methods	574
Object Class: DSOacleView	574
Attributes.	575
Cache View Methods	575
Object Class: DSOacleTablespace	576
Attributes.	576
Cache View Methods	577
Relationship Class: DSRLinkOclIndexForTable	577
Attributes.	578
Cache View Methods	578
Relationship Class: DSRLinkOclTableInPDesign	578
Attributes.	579
Cache View Methods	579
Relationship Class: DSRLinkOclIndexInPDesign	579
Attributes.	580
Cache View Methods	580
Relationship Class: DSRLinkOclTSpaceInPDesign	580
Attributes.	581
Cache View Methods	581
Relationship Class: DSRLinkOclViewInPDesign	581
Attributes.	582
Cache View Methods	582
Relationship Class: DSRLinkOclTableInTblspace	582
Attributes.	583
Cache View Methods	583
Relationship Class: DSRLinkOclIndexInTblspace	583
Attributes.	584
Cache View Methods	584
Relationship Class: DSRLinkOclTblspcInRDBSys	584
Attributes.	585
Cache View Methods	585

Chapter 13. Relational Database: Designer Extensions to the MVS

Submodel	587
Object Class: DSRMRDTablePartition	587
Attributes.	587
Cache View Methods	591
Object Class: DSRTodoList	591
Attributes.	591
Cache View Methods	593
Relationship Class: DSRLinkMRDHasPartitions	593
Attributes.	594
Cache View Methods	594
Relationship Class: DSRLinkMRDTPHasColumns.	595
Attributes.	595
Cache View Methods	596
Relationship Class: DSRLinkToDoListHasAltNames	596
Attributes.	597
Cache View Methods	597

Relationship Class: DSRLinkToDoListHasTables	597
Attributes.	598
Cache View Methods	598
Relationship Class: DSRLinkToDoListHasTblspaces	598
Attributes.	599
Cache View Methods	599
Relationship Class: DSRLinkToDoListHasStogroup	599
Attributes.	600
Cache View Methods	600
Relationship Class: DSRLinkToDoListHasViews	600
Attributes.	601
Cache View Methods	601
Relationship Class: DSRLinkToDoListHasIndexes	601
Attributes.	602
Cache View Methods	602
Relationship Class: DSRLinkToDoListHasDatabases.	602
Attributes.	603
Cache View Methods	603
Relationship Class: DSRLinkToDoListHasBufpools	604
Attributes.	604
Cache View Methods	605
Relationship Class: DSRPDesignHasToDoList	605
Attributes.	606
Cache View Methods	606
 Chapter 14. Transform Mapping Model	 607
Object Class: cdmModel	607
Attributes.	607
Cache View Methods	608
Object Class: cdmTransformProcess	608
Attributes.	609
Cache View Methods	610
Object Class: cdmAttributeToColumnMapping	611
Attributes.	611
Cache View Methods	612
Object Class: cdmEntityToTableMapping	612
Attributes.	612
Cache View Methods	613
Object Class: cdmMapping	613
Attributes.	614
Cache View Methods	614
Object Class: cdmRelshipToFKeyMapping	614
Attributes.	615
Cache View Methods	615
Relationship Class: cdmA2CMPartOfE2TM	615
Attributes.	616
Cache View Methods	616
Relationship Class: cdmLinkAttribute2CDef	617
Attributes.	617
Cache View Methods	618
Relationship Class: cdmLinkEntity2Table	618
Attributes.	619
Cache View Methods	619
Relationship Class: cdmLinkEntity2TDef	619
Attributes.	620
Cache View Methods	620

Relationship Class: cdmLinkModelTransformP	620
Attributes.	621
Cache View Methods	621
Relationship Class: cdmLinkRelship2FKKey	621
Attributes.	622
Cache View Methods	622
Relationship Class: cdmLinkTransformPEnt2Tab	622
Attributes.	623
Cache View Methods	623
Relationship Class: cdmModelUsesDE	623
Attributes.	624
Cache View Methods	624
Relationship Class: cdmR2FKPartOfE2TM	624
Attributes.	625
Cache View Methods	625
Relationship Class: cdmTPTransformed2RD.	625
Attributes.	626
Cache View Methods	626
Customer support	627
Bibliography	629
IBM VisualAge TeamConnection library	629
Tool Builder's Development Kit.	629
ObjectStore.	630
IBM Exchange library	630
Related publications	630

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY, USA 10594.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact the Site Counsel, IBM Corporation, P.O. Box 12195, 3039 Cornwallis Road, Research Triangle Park, NC 27709-2195, USA. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

IBM may change this publication, the product described herein, or both. These changes will be incorporated in new editions of the publication.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

BookManager	MVS/XA
Common User Access	NetView
C Set++	Operating System/2
CUA	OS/2
C/370	TeamConnection
IBM	VisualAge Generator
MVS/ESA	XGA

The following terms are trademarks of other companies:

ObjectStore

ObjectStore Design, Inc.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Conventions

This book uses the following highlighting conventions:

- *Italics* are used to indicate the first occurrence of a word or phrase that is defined in the glossary. They are also used for information that you must replace.
- **Bold** is used to indicate items on the GUI.
- Monospace font is used to indicate exactly how you type the information.
- File names follow Intel conventions: **mydir\myfile.txt**. AIX and HP-UX users should render this file name **mydir/myfile.txt**.

Tips or platform specific information is marked in this book as follows:



Shortcut techniques and other tips



IBM VisualAge TeamConnection for OS/2



IBM VisualAge TeamConnection for Windows 3.1



IBM VisualAge TeamConnection for Windows/NT



IBM VisualAge TeamConnection for Windows 95



IBM VisualAge TeamConnection for AIX



IBM VisualAge TeamConnection for HP-UX

Tell us what you think

In the back of this book is a comment form. Please take a few moments to tell us what you think about this book. The only way for us to know if you are satisfied with our books or if we can improve their quality is through feedback from customers like you.

Chapter 1. Information Model Overview

This overview section is intended to introduce the general content of the Information Model by introducing segments of the model called subject areas. In each subject area, the object classes that participate in the subject area are briefly described and a diagram is provided to graphically depict the role or roles each object class plays in that subject area. Complete definitions of each object class and relationship class are located in the other sections of this reference.

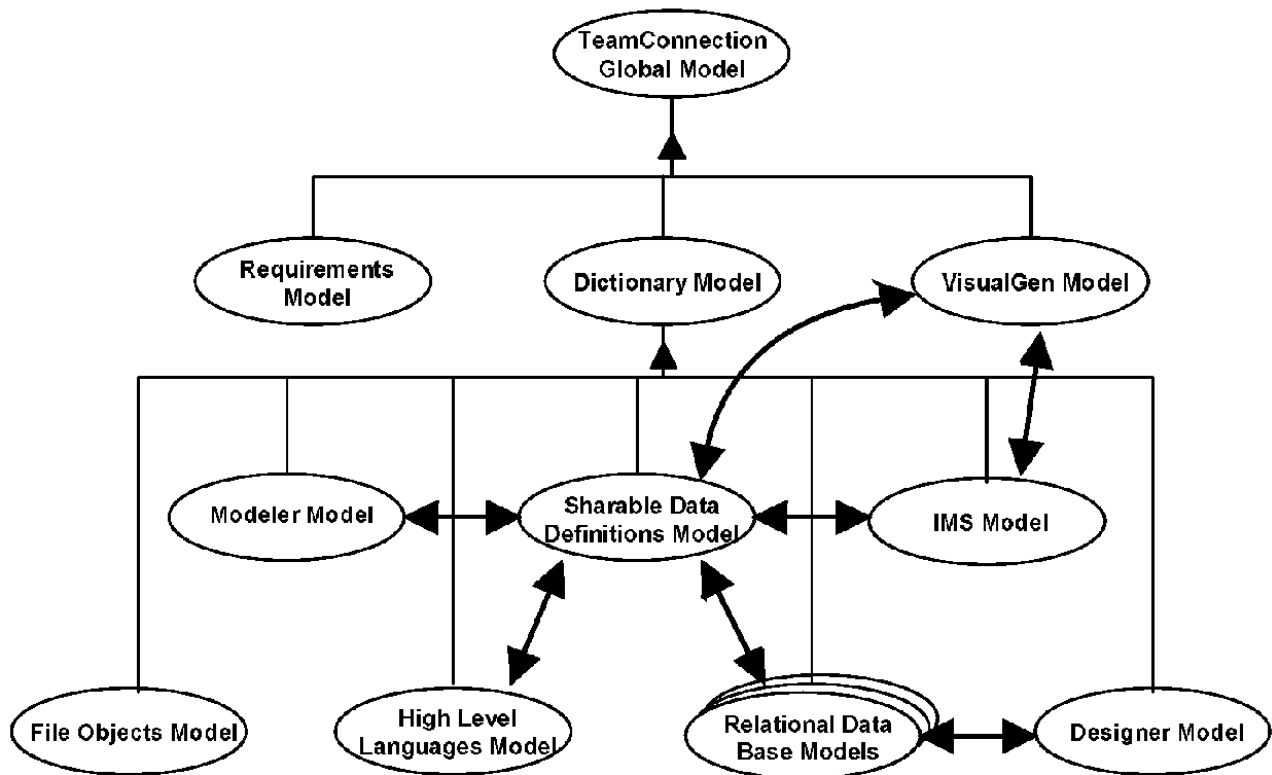


Figure 1. Overview of the Information Model

The Information Model can be viewed as the tightly integrated union of several submodels. The current set of submodels include:

- “GLO Global Subject Area” on page 2

The shareable, top-level objects and their relationships to objects in other areas.

In general, there are two categories of objects in this model:

- Named objects
- Un-named objects.

All objects surfaced to tools’ users are considered as named objects. These objects are made subclasses of the **TCPart** class, which itself is a subclass of **ADAnnotatedObject**. Other objects, which do not fit in the first category, are made directly subclasses of **ADAnnotatedObject** or its superclass, **ADObject**.

Only named objects can have alias names.

- “SDD Shareable Data Structure & Element” on page 3

This section describes the main objects that are currently shared by and can be extended to support the following technologies:

- Relational databases
- Information Management System (IMS)
- COBOL
- PL/I
- External Source Format
- C and C++ programming languages

- “HLL Global and Local Data Definitions” on page 7

Descriptions of the user objects that support data definitions in the following high level languages:

- COBOL
- PL/I
- External Source Format (ESF) for VisualGen

Extensions to support C and C++ programming languages are considered during the design but have not been completely defined.

- “VisualAge Requirements Tool” on page 24

The Requirements model is designed to help business users determine and specify requirements for a business operations. The Requirements model contains three elements:

- “Business Facts” on page 25
- “Business Information” on page 25
- “Business Rules” on page 25

These submodels are tied together through connecting relationships and through common or shared classes. Any boundaries inferred or perceived between submodels or subject areas are purely artificial and only useful for understanding the usage of the object classes and relationship classes which form the Information model.

GLO Global Subject Area

Category Name

GLO Global Subject Area

Description

Classes in this category are the top of the class hierarchy. They provide the basic features and functionality found throughout the Information Model. The classes in this subject area define the basic behaviors found in all classes in the Information Model.

Class Name

Description

ADAnnotationText

An instance of this object class represents a labeled string that forms the annotation text of an object.

This text string can have a label to indicate the type of the annotation text. The default label is DESCRIPTION. Possible annotation text labels are:

- DESCRIPTION
- EXAMPLE
- NOTE

The label of an annotation text can be changed from one to another.

TCPart

This is a superclass of all named objects. Only subclasses of TCPart may serve as the root class for a view. Only subclasses of TCPart may be identified as participants, inputs, or outputs of a Build process. Being a named object, any TCPart subclass instance may be accessed and manipulated directly via its unique name without necessitating navigation via existing relationship instances.

ADAnnotatedObject

An instance of this object class represents an object that can be associated with annotation text.

ApplicationCollector

An instance of this class type represents a collection of TCParts. A given collection may serve to identify the component objects which form an application, a system, subsystem, or any other logical group.

AtlasObject

This class defines a representation of all named objects and the common behaviors and functions supported for named objects.

BuildEvent

This class is introduced to provide a representation of a Build Event. Each instance of BuildEvent identifies those named objects (subclasses of TCPart) which are transformed by, generated by, or associated with a single Build process.

DeviceObject

This is the base class for all device objects, such as Repository, Library, and RDB Catalog. These objects can be opened into a container view on the User Interface.

ewsContent

An instance of this class represents the collection of named objects nested within a device or technology object. This class is used and maintained to keep track of the nested objects to be displayed.

TechnologyObject

This is the base class for all technology objects. All named objects that represent technology-based concepts and components are subclasses of TechnologyObject.

SDD Shareable Data Structure & Element

Category Name

SDD Shareable Data Structure & Element

Description

Classes in this category establish the basis for sharing common data element and data structure definitions. These global definitions can be referenced within many technologies to enable consistency and accuracy throughout all components of an application.

Class Name	
Description	

DataComponent

This is an abstract superclass of objects that can be members of a data structure definition. This class is needed to support PLI INCLUDE and COBOL COPY statement within a data structure. See the subclasses for details.

DataDefinition

This is an abstract superclass of objects that provide non-shareable data type definitions. These data type definitions identify the characteristics of a data item. See its subclasses for details.

DataElement

An object of this class represents a ***shareable named*** data element.

DataItem

An instance of this object class represents a storage location mapped by a high level language (HLL) data declaration.

DataStructure

An object of this class represents a ***shareable named*** data structure.

IncludeComponent

This is an abstract superclass of all objects that can be components of an include file. This class is needed to maintain the order of the components within the include file. See the subclasses for details.

SharedDataDefinition

This object class is an abstract superclass which defines a representation of a ***shareable named*** data definition. See its subclasses for details.

SimpleTypeDef

An instance of this class represents a simple type definition for an elementary data item. This simple type definition can be used in more than one technology, such as relational database, IMS, and high level languages.

StructureDef

An instance of this class represents the structure type definition of an HLL record, an IMS Segment, or a VisualGen Record.

For COBOL, this object represents the definition of a data item that contains subordinate data items.

For PL/I, this object represents the definition of a major structure or a minor structure.

For C and C++, this object represents:

- A data type of struct with a tag name.
- A data type of struct without a tag name.

TechnologyObject

This is the base class for all technology objects. All named objects that represent technology-based concepts and components are subclasses of TechnologyObject.

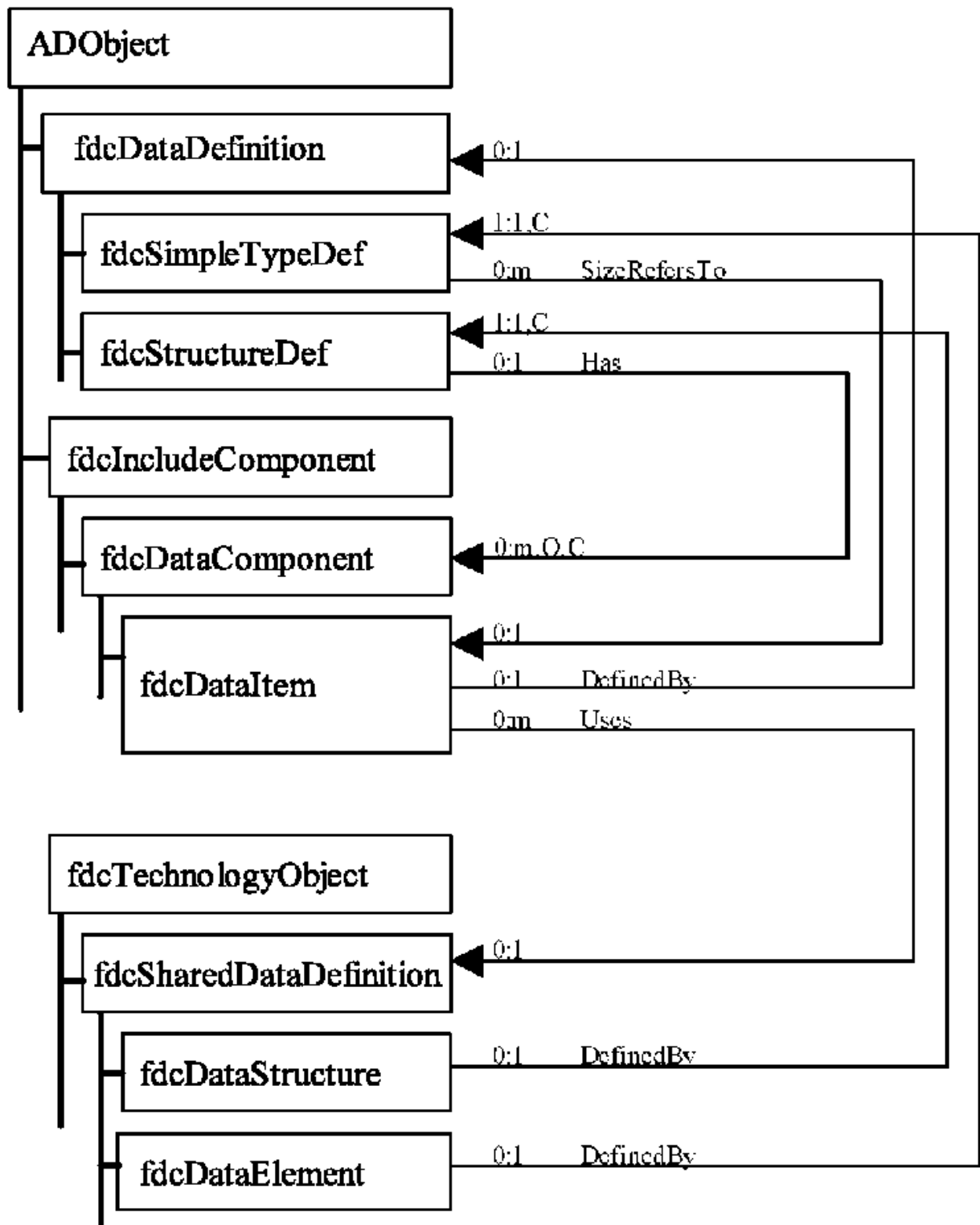


Figure 2. Category Class-Relationships Diagram - Shareable Data Structure & Element

SDD Alias Support For DataElement and DataStructure

Category Name

SDD Alias Support For DataElement and DataStructure

Description

This section describes the model to support aliases of a shared DataElement and aliases for a shared Data Structure. The mechanisms used to define aliases for both DataElement and DataStructure are generally identical. The primary difference exists in the sets of object classes that can establish aliases for DataElement versus DataStructure.

DataElement When an instance of one of the following object classes uses an instance of DataElement as the source of its type definition, its name is considered to be an alias of the related DataElement:

- Dataltem
- CobolDataDefinition
- PLIDataDefinition
- ColumnDefinition
- IMSField

DataStructure When an instance of one of the following object classes uses an instance of DataStructure as the source of its type definition, its name is considered to be an alias of the related DataStructure:

- Sub-structure data item
- Segment

Class Name

Description

ADAlias

This object class defines a representation of an alias name of a named object in the repository. See its subclasses for details.

ColumnDefinition

An instance of this object class represents a column for relational Tables. A column uses some information from a Data Element.

DataElement

An object of this class represents a **shareable named** data element.

DataElementAlias

An instance of this class represents an alias name for a **DataElement** object.

Dataltem

An instance of this object class represents a storage location mapped by a high level language (HLL) data declaration.

DataStructure

An object of this class represents a **shareable named** data structure.

DataStructureAlias

An instance of this class represents an alias name for a **DataStructure** object.

IMSField

An instance of this object type is used to associate an instance of Segment with a Dataltem instance that is contained in the DataStructure instance

connected to the Segment and to provide the attributes that apply only to a DataItem used by a Segment in a specific DBD.

Segment

An instance of this object class (or of one of its sub-classes) represents a segment within a DBD.

SDDAlias

This is an abstract superclass object class which defines a representation of an alias for a *shareable named* data definition. See its subclasses for details.

SharedDataDefinition

This object class is an abstract superclass which defines a representation of a *shareable named* data definition. See its subclasses for details.

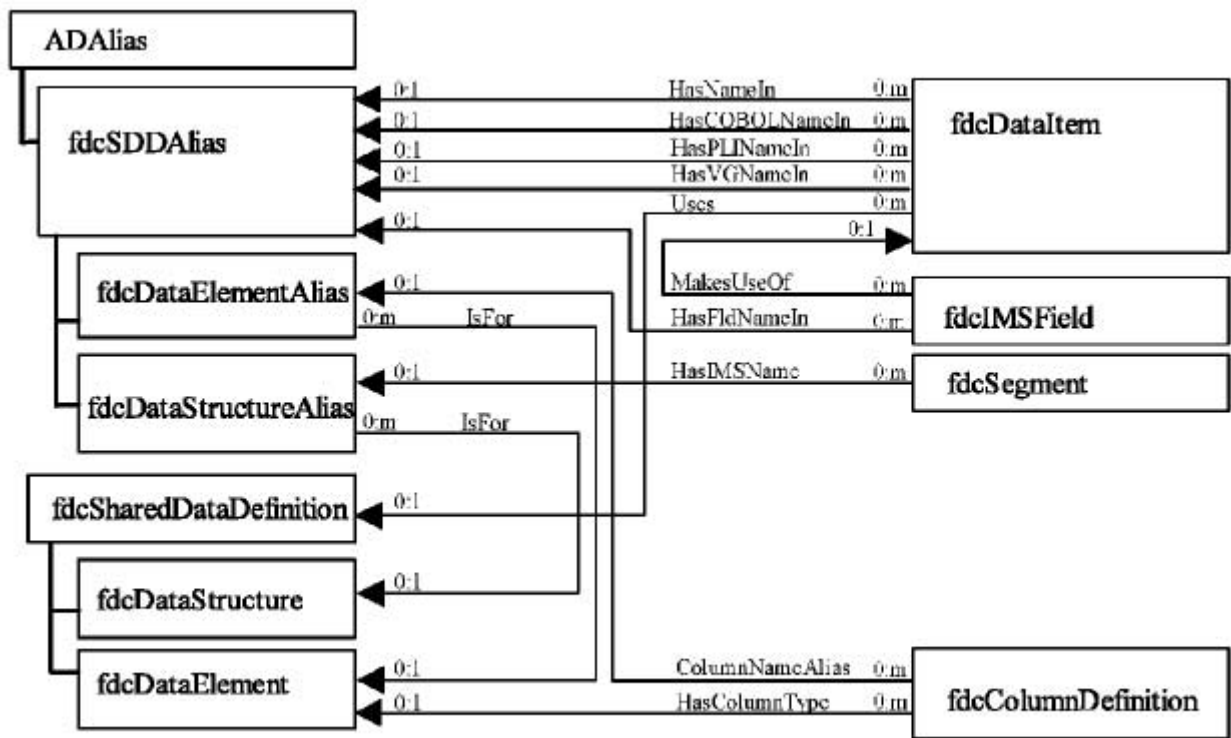


Figure 3. Category Class-Relationships Diagram for SDD Alias Support For DataElement

HLL Global and Local Data Definitions

Category Name

HLL Global and Local Data Definitions

Description

The object classes in this section are members of one of two sub hierarchies of the Information Model. One set (subclasses of DataDefinition) defines the detailed data types for data components, whether local to one particular use or globally shareable (reusable). The other set (subclasses of

SharedDataDefinition) attaches globally unique names to data type definitions thus making the definitions shareable.

Class Name

Description

Array An object of this class represents a *shareable named* array.

ArrayDef

An instance of this class represents the definition of one dimension of an array. Multi-dimensional array is represented using the relationship ArrayDef_HasElement_DataDef and the related classes.

DataDefinition

This is an abstract superclass of objects that provide non-shareable data type definitions. These data type definitions identify the characteristics of a data item. See its subclasses for details.

DataElement

An object of this class represents a *shareable named* data element.

DataStructure

An object of this class represents a *shareable named* data structure.

EnumeratedList

An object of this class represents a *shareable named* enumerated list.

EnumerationDef

An instance of this class represents a C *enum* data type or a PL/I ordinal data type.

Pointer

An object of this class represents a *shareable named* pointer.

PointerDef

An instance of this object class represents a high level language pointer type.

SharedDataDefinition

This object class is an abstract superclass which defines a representation of a *shareable named* data definition. See its subclasses for details.

SimpleTypeDef

An instance of this class represents a simple type definition for an elementary data item. This simple type definition can be used in more than one technology, such as relational database, IMS, and high level languages.

StructureDef

An instance of this class represents the structure type definition of an HLL record, an IMS Segment, and/or a VisualGen Record.

For COBOL, this object represents the definition of a data item that contains subordinate data items.

For PL/I, this object represents the definition of a major structure or a minor structure.

For C/C++, this object represents:

- A data type of struct with a tag name.
- A data type of struct without a tag name.

Union An object of this class represents a *shareable named* HLL union.

UnionDef

An instance of this object class represents C union data type, or PL/I UNION attribute.

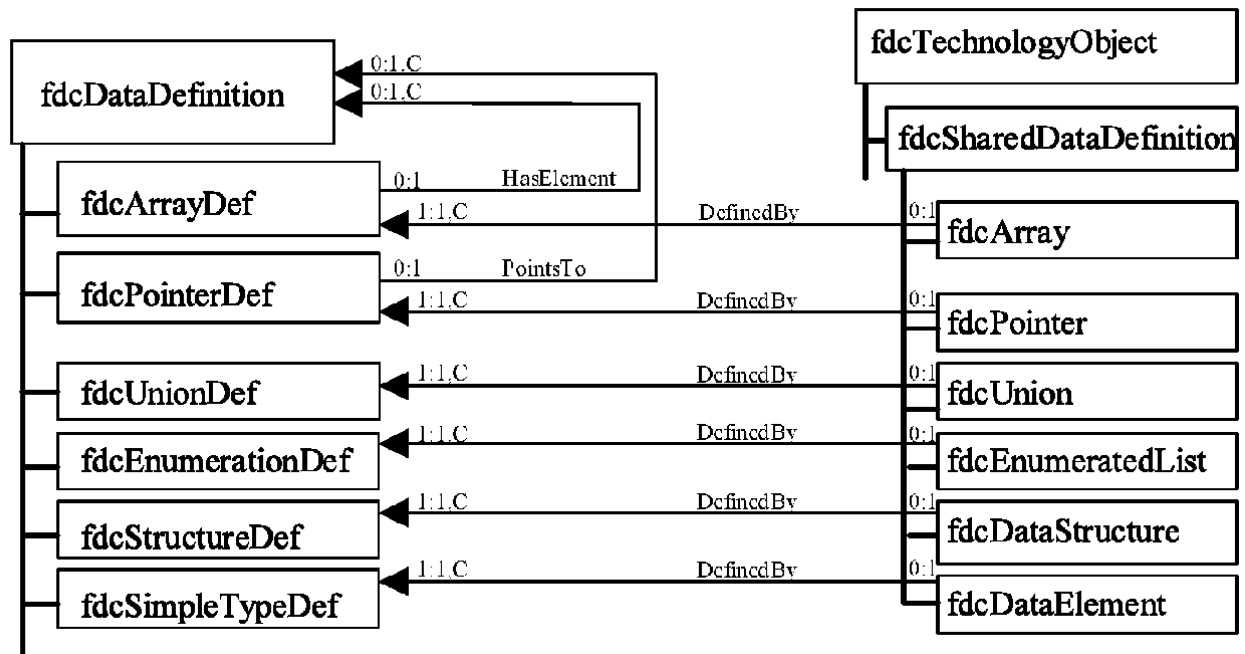


Figure 4. Category Class-Relationships Diagram for HLL Global and Local Data Definitions

DBD area

Category Name

DBD area

CompRoutine

An instance of this object type is used to describe the compression routine parameters for a segment.

Description

This category describes the internal structure of an IMS Data Base Description (DBD).

Class Name

Description

Dataltem

An instance of this object class represents a storage location mapped by a high level language (HLL) data declaration.

DataPropagationExit

An instance of this object type is used to hold the attributes that affect the relationship between a DBD or a Segment and the data capture routine (EXIT keyword on DBD or SEGM statement). It is possible for an instance of DataPropagationExit to be used by a DBD and more than one segment within the DBD.

DataStructure

An object of this class represents a *shareable named* data structure.

DataStructure

An instance of this class represents an alias name for a *DataStructure* object.

DBD An instance of this object class represents an IMS Data Base description.

DBDaccess

An instance of this class holds the access method of the DBD and access-specific attributes of a DBD object. Instances of this DBDaccess object class are used by DBDs with access methods of GSAM, HSAM, HISAM, SHISAM, SHSAM, and LOGICAL. DBDs with other access methods (MSDB, INDEX, HIDAM, DEDB, HDAM) use instances of the subclasses of this object class.

DBDdedb

An instance of this object class represents a DBD object that has access=DEDB.

DBDhdam

An instance of this object class represents a DBD object that has access=HDAM. These attributes apply to the randomizing module relationship that a valid HDAM DBD must have.

DBDhidam

An instance of this object class represents a DBD object with an access method of HIDAM. A HIDAM DBD must have a primary index relationship to be valid. The relationship maps to an LCHILD statement under the root segment that has POINTER=INDX and no associated XDFLD statement.

DBDindex

An instance of the DBDindex object class represents a DBD object that can be used to index a HIDAM database or a segment in a HDAM, HIDAM or HISAM database. The indexing relationship maps to the LCHILD statement in the macro language description of an index DBD.

DBDmsdb

a DBD with access=MSDB (Mass Storage Data Base).

DDNAME

An instance of this object class represents the DD name specified for a DATASET or AREA statement in a DBD.

IMSArea

Instances of this object type are used to add attributes for the AREA macro statement to the relationship between a DBD instance and the instance of DDNAME that represents the DD1 attribute on the AREA statement.

IMSDataset

Instances of this object type are used to add attributes for the DATASET macro statement to the relationship between a DBD instance and the instance of DDNAME that represents the DD1 attribute on the DATASET statement.

IMSField

An instance of this object type is used to associate an instance of Segment with a Dataltem instance that is contained in the DataStructure instance connected to the Segment and to provide the attributes that apply only to a Dataltem used by a Segment in a specific DBD.

SDDAlias

This is an abstract superclass object class which defines a representation of an alias for a ***shareable named*** data definition. See its subclasses for details.

SecondaryIndex

Instances of this object type are used to associate a SegmentComplex instance to the DBDindex instances that act as secondary indexes for the DBD object (maps to combination of LCHILD and XDFLD statements).

Segment

An instance of this object class (or of one of its sub-classes) represents a segment within a DBD object.

SegmentComplex

An instance of this class represents a segment in a DBD that can have logical and index relationships. The DBD access method must be HDAM, HISAM, or HIDAM.

SegmentDEDB

An instance of this object class acts as a Segment within a DBD object that has access=DEDB.

SegmentLogical

An instance of this object class represents a segment in a DBD object that has access method of LOGICAL. Segments in a logical DBD have relationships to segments in other DBDs instead of a MappedBy relationship to an dataStructure.

CompRoutine

An instance of this object type is used to describe the compression routine parameters for a segment.

DBD An instance of this object class represents an IMS Data Base description.

DBDaccess

An instance of this class holds the access method of the DBD and access-specific attributes of a DBD object. Instances of this DBDaccess object class are used by DBDs with access methods of GSAM, HSAM, HISAM, SHISAM, SHSAM, and LOGICAL. DBDs with other access methods (MSDB, INDEX, HIDAM, DEDB, HDAM) use instances of the subclasses of this object class.

DBDindex

An instance of the DBDindex object class represents a DBD object that can be used to index a HIDAM database or a segment in a HDAM, HIDAM or HISAM database. The indexing relationship maps to the LCHILD statement in the macro language description of an index DBD.

IMSField

An instance of this object type is used to associate an instance of Segment with a Dataltem instance that is contained in the DataStructure instance connected to the Segment and to provide the attributes that apply only to a Dataltem used by a Segment in a specific DBD.

PCB Instances of this object class identify IMS Program Communication Blocks (PCB's). A PCB is a series of macro instructions contained in a PSB. Each PCB represents a message destination or data base to be used by the application program.

PCBbased

An instance of this object class represents an IMS PCB (Program Communication Block) that refers to a data base to be used by an application program. It states the DBD Name of the data base the application program accesses and defines the types of operations (get, insert, replace, etc.) the application program can perform on that data base.

PCBdatabase

An instance of this object class represents a data base PCB. A data base PCB describes an application programs access abilities to a physical or logical, non-GSAM, non-Index, hierarchical data base. This type of PCB defines a logical structure which is the hierarchical set of data segments to which the associated application program is sensitive.

PCBteleprocessing

An instance of this object class represents an Alternate PCB object (TP PCB). This type of PCB describes a destination other than the source of the current input message. It allows the application program to send output messages to a destination other than the source of an input message.

PCBtype

This is an abstract superclass whose subtypes hold the type-specific attributes of PCBs.

PSB An instance of this object class represents an IMS Program Specification Block (PSB). Within IMS, a PSB is a series of PCB macro instructions that describe an application program's I/O operations and its view and use of segments and fields in IMS databases. The types of PCBs are TP PCB (PCBteleprocessing) which describes interactions with logical terminals,

GSAM PCB (PCBbased) which is based on a GSAM DBD used as an input or output dataset, and DB PCB (PCBdatabase) which can relate to segments and fields in its base DBD.

Segment

An instance of this object class (or of one of its sub-classes) represents a segment within a DBD object.

SensitiveField

Instances of this object class are used to associate a SensitiveSegment instance to the IMSField instances that represent the fields in the segment to which the PCB must be sensitive.

SensitiveSegment

Instances of this object type are used to associate a PCBdatabase instance with the Segment instances that represent the segments to which the PCB is sensitive.

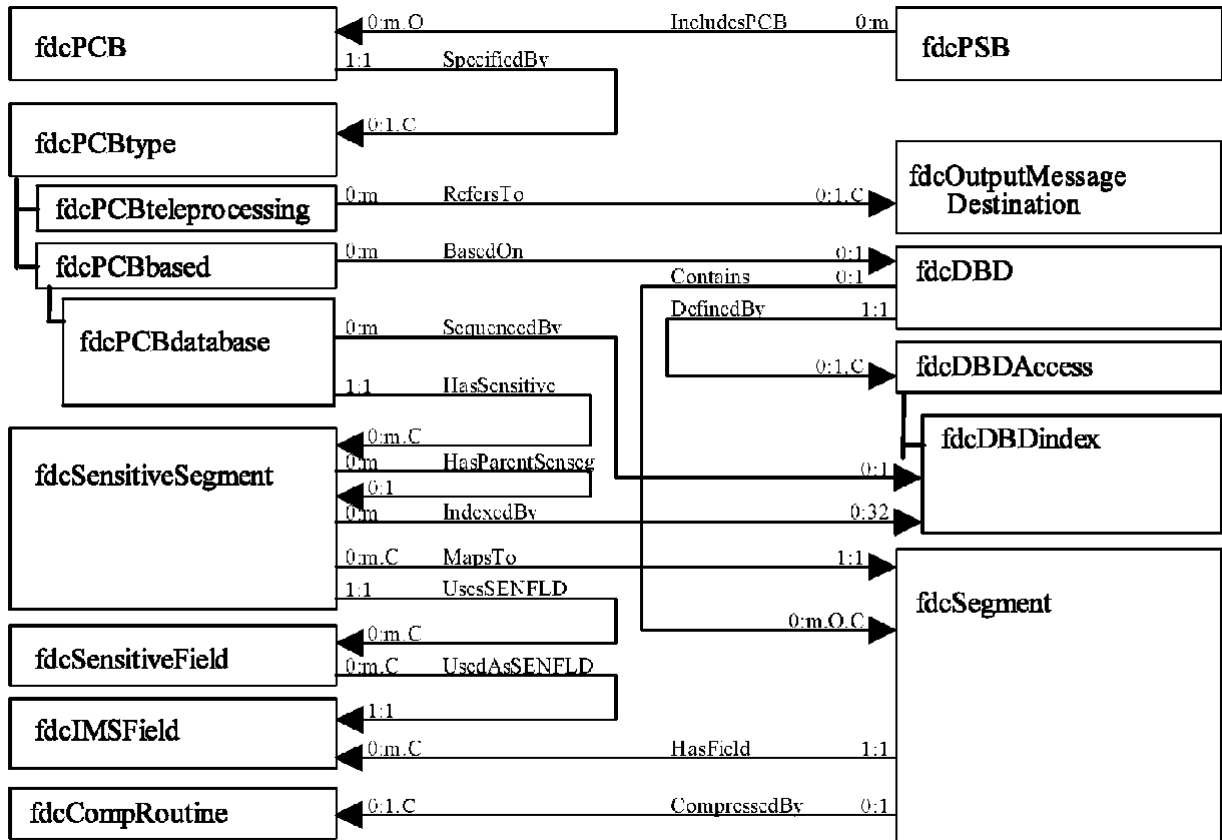


Figure 6. Category Class-Relationships Diagram for PSB area

HLL Global and Local Data Definitions

Category Name

HLL Global and Local Data Definitions

Description

The object classes in this section are members of one of two sub hierarchies of the Information Model. One set (subclasses of DataDefinition) defines the detailed data types for data components, whether local to one particular use or globally shareable (reusable). The other set (subclasses of SharedDataDefinition) attaches globally unique names to data type definitions thus making the definitions shareable.

Class Name

Description

Array An object of this class represents a ***shareable named*** array.

ArrayDef

An instance of this class represents the definition of one dimension of an array. Multi-dimensional array is represented using the relationship ArrayDef_HasElement_DataDef and the related classes.

DataDefinition

This is an abstract superclass of objects that provide non-shareable data type definitions. These data type definitions identify the characteristics of a data item. See its subclasses for details.

DataElement

An object of this class represents a ***shareable named*** data element.

DataStructure

An object of this class represents a ***shareable named*** data structure.

EnumeratedList

An object of this class represents a ***shareable named*** enumerated list.

EnumerationDef

An instance of this class represents a C ***enum*** data type or a PL/I ordinal data type.

Pointer

An object of this class represents a ***shareable named*** pointer.

PointerDef

An instance of this object class represents a high level language pointer type.

SharedDataDefinition

This object class is an abstract superclass which defines a representation of a ***shareable named*** data definition. See its subclasses for details.

SimpleTypeDef

An instance of this class represents a simple type definition for an elementary data item. This simple type definition can be used in more than one technology, such as relational database, IMS, and high level languages.

StructureDef

An instance of this class represents the structure type definition of an HLL record, an IMS Segment, and/or a VisualGen Record.

For COBOL, this object represents the definition of a data item that contains subordinate data items.

For PL/I, this object represents the definition of a major structure or a minor structure.

For C/C++, this object represents:

- A data type of struct with a tag name.
- A data type of struct without a tag name.

TechnologyObject

This is the base class for all technology objects. All named objects that represent technology-based concepts/components are subclasses of TechnologyObject.

Union An object of this class represents a **shareable named** HLL union.

UnionDef

An instance of this object class represents C union data type, or PL/I UNION attribute.

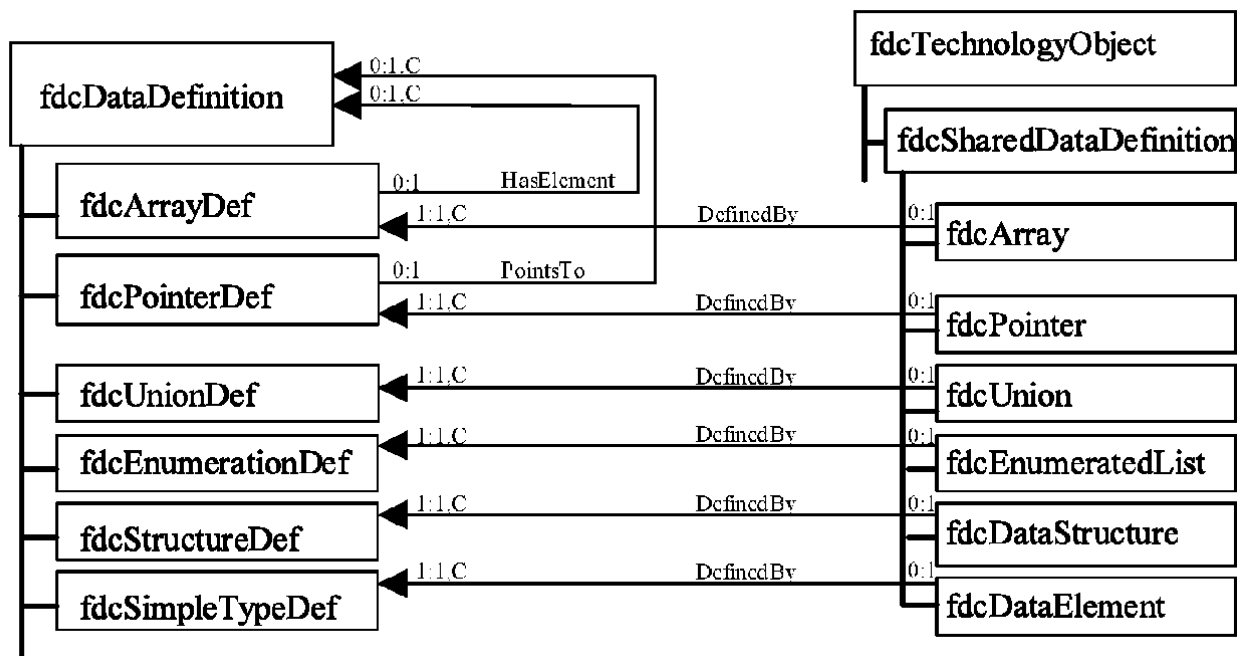


Figure 7. Category Class-Relationships Diagram for HLL Global and Local Data Definitions

HLL Include Components

Category Name

HLL Include Components

Description

The object classes in this section represent those elements that serve as components of structures or records included into high level language source files.

Class Name

Description

DataComponent

This is an abstract superclass of objects that can be members of a data structure definition. This class is needed to support PLI INCLUDE and COBOL COPY statement within a data structure. See the subclasses for details.

DataItem

An instance of this object class represents a storage location mapped by a high level language (HLL) data declaration.

FunctionDeclaration

An instance of this object class represents a function or procedure interface. For PL/I, it represents an ENTRY data declaration. For C, it represents a function prototype.

IncludeComponent

This is an abstract superclass of all objects that can be components of an include file definition. This class is needed to maintain the order of the components within the definition. See the subclasses for details.

IncludedSourceDef

An instance of this object class represents a file referenced in a COBOL COPY, PL/I Include, or C #include statement contained in a source program. Each instance may be included multiple times and may contain nested include files.

IncludeStatement

An instance of this class represents a COBOL COPY, PL/I INCLUDE, or C #include statement.

SharedDataDefinition

This object class is an abstract superclass which defines a representation of a **shareable named** data definition. See its subclasses for details.

TypeDefReference

An instance of this class represents the include component that is a C/C++ **typedef** or PL/I **alias** declaration. This class is needed to maintain the relative position of a type definition within an include file.

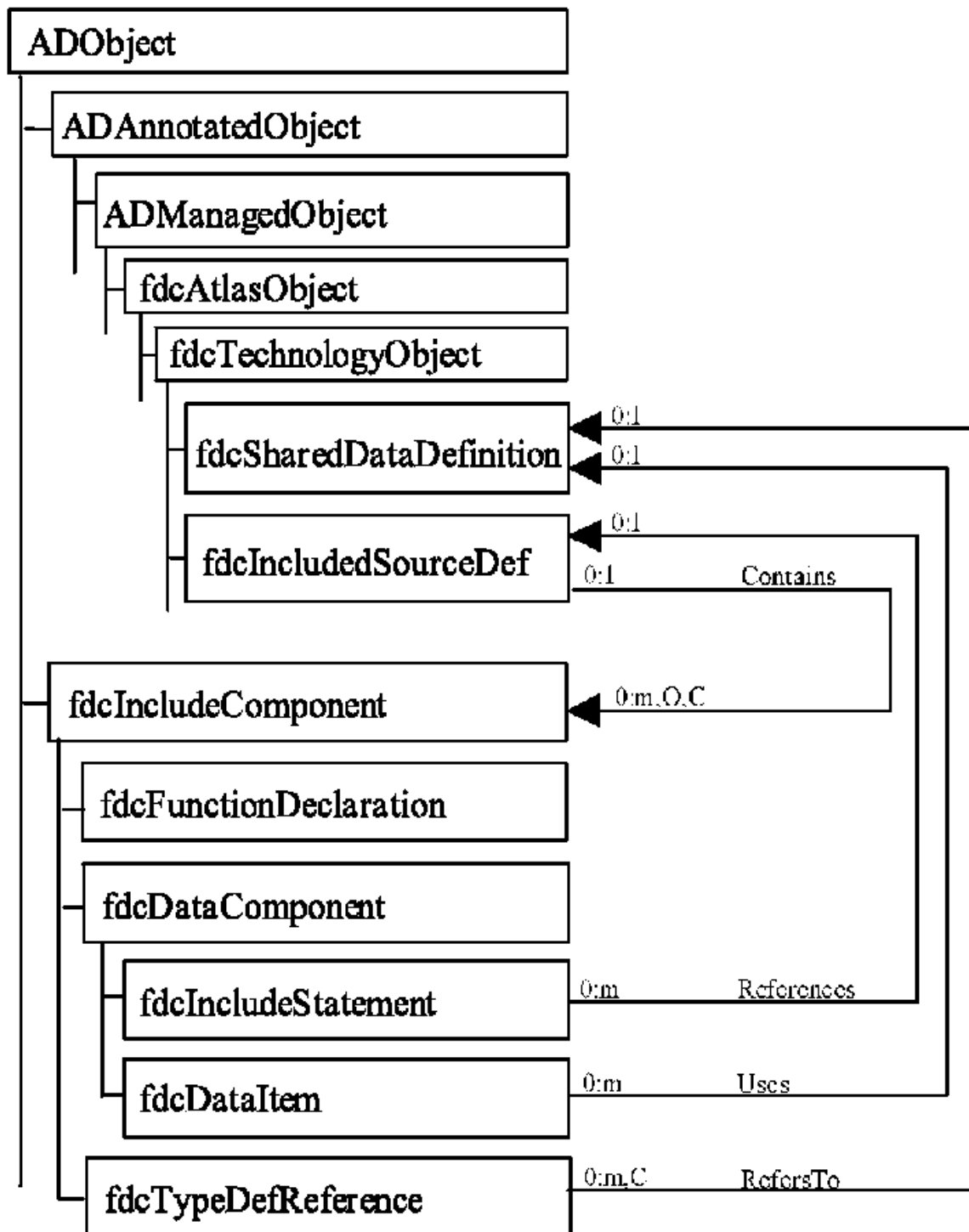


Figure 8. Category Class-Relationships Diagram for HLL Include Components

HLL Sub-Structure, Array, and Union

Category Name

HLL Sub-Structure, Array, and Union

Description

The objects in this section represent composite data definitions that may be simple or complex structures, arrays, or unions of elemental or other complex data definitions.

Class Name

Description

Array An object of this class represents a *shareable named* array.

ArrayDef

An instance of this class represents the definition of one dimension of an array. Multi-dimensional array is represented using the relationship ArrayDef_HasElement_DataDef and the related classes.

DataComponent

This is an abstract superclass of objects that can be members of a data structure definition. This class is needed to support PLI INCLUDE and COBOL COPY statement within a data structure. See the subclasses for details.

DataDefinition

This is an abstract superclass of objects that provide non-shareable data type definitions. These data type definitions identify the characteristics of a data item. See its subclasses for details.

DataItem

An instance of this object class represents a storage location mapped by a high level language (HLL) data declaration.

DataStructure

An object of this class represents a *shareable named* data structure.

SharedDataDefinition

This object class is an abstract superclass which defines a representation of a *shareable named* data definition. Please see its subclasses for details.

StructureDef

An instance of this class represents the structure type definition of an HLL record, an IMS Segment, and/or a VisualGen Record.

For COBOL, this object represents the definition of a data item that contains subordinate data items.

For PL/I, this object represents the definition of a major structure or a minor structure.

For C/C++, this object represents:

- A data type of struct with a tag name.
- A data type of struct without a tag name.

TypeDefReference

An instance of this class represents the include component that is a C/C++ *typedef* or PL/I *alias* declaration. This class is needed to maintain the relative position of a type definition within an include file.

Union An object of this class represents a *shareable named* HLL union.

UnionDef

An instance of this object class represents C union data type, or PL/I UNION attribute.

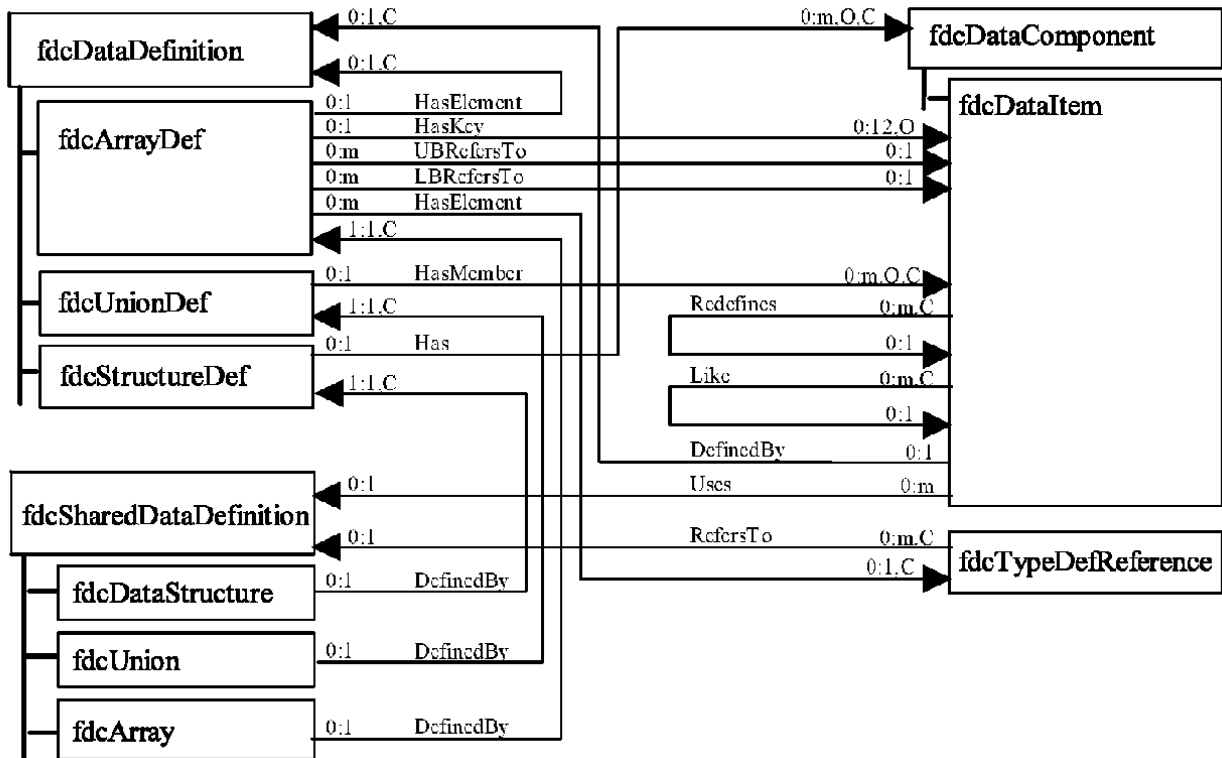


Figure 9. Category Class-Relationships Diagram for HLL Structure, Array, and Union

HLL Elementary Data Item

Category Name

HLL Elementary Data Item

Description

Objects in this section define elemental HLL data items. The data items defined may be either local or global data items. They will have the characteristics of a simple, unconstrained, unstructured element, an unstructured element with an enumerated list of allowed values, or a HLL pointer element.

Class Name

Description

CobolDataDefinition

An instance of this class represents the COBOL-specific data definitions for a data item.

DataDefinition

This is an abstract superclass of objects that provide non-shareable data type definitions. These data type definitions identify the characteristics of a data item. Please see its subclasses for details.

DataElement

An object of this class represents a ***shareable named*** data element.

DataElementAlias

An instance of this class represents an alias name for a ***DataElement*** object.

Short name: DEAlias

DataItem

An instance of this object class represents a storage location mapped by a high level language (HLL) data declaration.

EnumeratedList

An object of this class represents a ***shareable named*** enumerated list.

EnumerationDef

An instance of this class represents a C ***enum*** data type or a PL/I ordinal data type.

PLIDataDefinition

An instance of this object class represents PL/I-specific data definitions for a data item.

Pointer

An object of this class represents a ***shareable named*** pointer.

PointerDef

An instance of this object class represents a high level language pointer type.

SDDAlias

This object class defines a representation of an alias for a ***shareable named*** data definition.

This is a superclass, which serves mainly as the target of the relationships from other classes. Please see its subclasses for details.

Short name: SDDAlias

SharedDataDefinition

This object class is an abstract superclass which defines a representation of a ***shareable named*** data definition. Please see its subclasses for details.

SimpleTypeDef

An instance of this class represents a simple type definition for an elementary data item. This simple type definition can be used in more than one technology, such as relational database, IMS, and high level languages.

TypeDefReference

An instance of this class represents the include component that is a C/C++ ***typedef*** or PL/I ***alias*** declaration. This class is needed to maintain the relative position of a type definition within an include file.

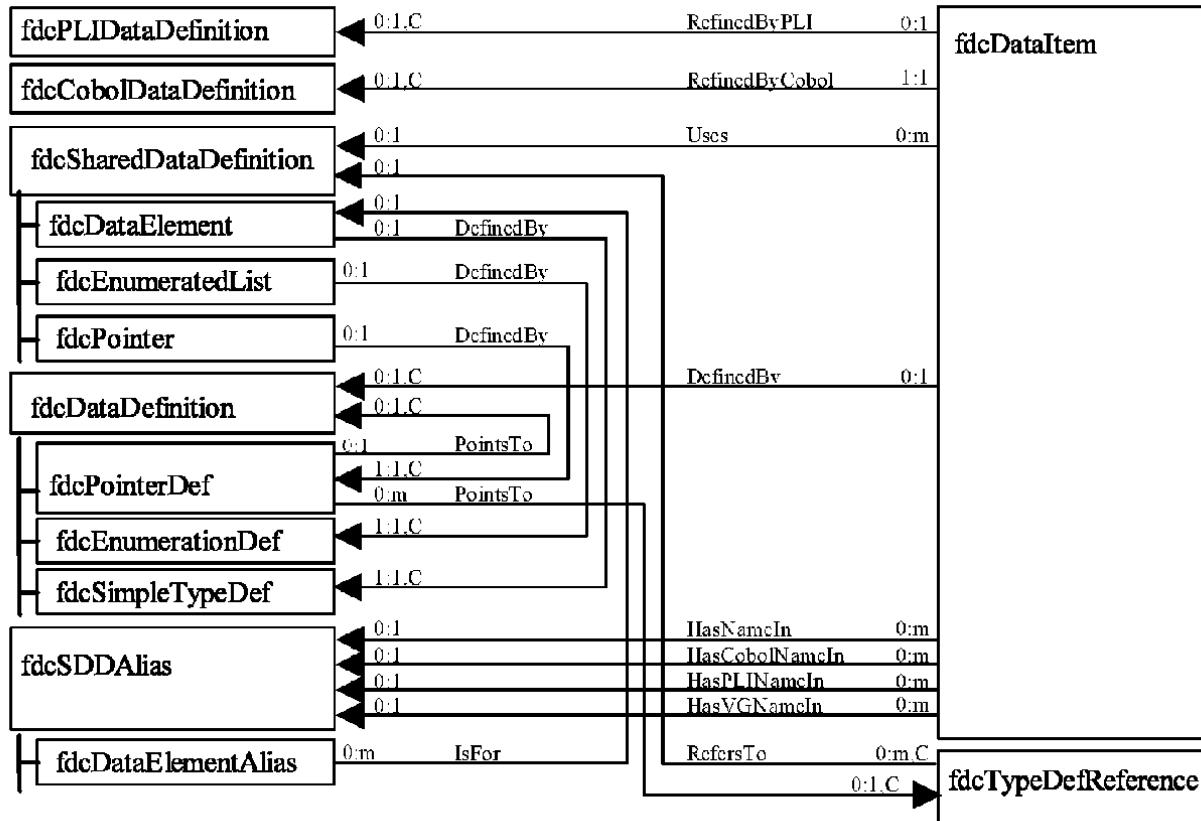


Figure 10. Category Class-Relationships Diagram for HLL Elementary Data Item

Extensions for COBOL and PL/I

Category Name

Extensions for COBOL

Description

This category includes classes and relationships that are applicable specifically to COBOL, classes and relationships that are applicable specifically to PL/I, classes and relationships to describe a PL/I ENTRY data declaration or C function prototype, and classes and relationships to describe a PL/I Alias or a C or C++ typedef. These technology specific classes and relationships are in addition to those in the Shared Data Definitions category.

Class Name

Description

ADAlias

This object class defines a representation of an alias name of a named object in the repository. Please see its subclasses for details.

CFunctionPrototype

An instance of this class represents a C function prototype.

CobolDataDefinition

An instance of this class represents the COBOL-specific data definitions for a data item.

CobolSpecialDataName

An instance of this object class represents either a COBOL level-66 or level-88 data names.

Short name: CblSpNm

DataComponent

This is an abstract superclass of objects that can be members of a data structure definition. This class is needed to support PLI INCLUDE and COBOL COPY statement within a data struture. Please see the subclasses for details.

Dataterm

An instance of this object class represents a storage location mapped by a high level language (HLL) data declaration.

DataComponent

This is an abstract superclass of objects that can be members of a data structure definition. This class is needed to support PLI INCLUDE and COBOL COPY statement within a data struture. Please see the subclasses for details.

DataDefinition

This is an abstract superclass of objects that provide non-shareable data type definitions. These data type definitions identify the characteristics of a data item. Please see its subclasses for details.

Dataterm

An instance of this object class represents a storage location mapped by a high level language (HLL) data declaration.

FunctionDeclaration

An instance of this object class represents represents a function or procedure interface. For PL/I, it represents an ENTRY data declaration. For C, it represents a function prototype.

IncludeComponent

This is an abstract superclass of all objects that can be components of an include file definition. This class is needed to maintain the order of the components within the definition. Please see the subclasses for details.

Offset An instance of this object class represents PL/I OFFSET data attribute.

PLIArea

An instance of this object class represents PL/I AREA data attribute.

PLIDataDefinition

An instance of this object class represents PL/I-specific data definitions for a data item.

PLIEntryDataDCL

An instance of this object class represents PL/I ENTRY data declaration.

SDDAlias

This is an abstract superclass object class which defines a representation of an alias for a **shareable named** data definition. Please see its subclasses for details.

SharedDataDefinition

This object class is an abstract superclass which defines a representation of a **shareable named** data definition. Please see its subclasses for details.

TypeDefReference

An instance of this class represents the include component that is a C/C++ **typedef** or PL/I **alias** declaration. This class is needed to maintain the relative position of a type definition within an include file.

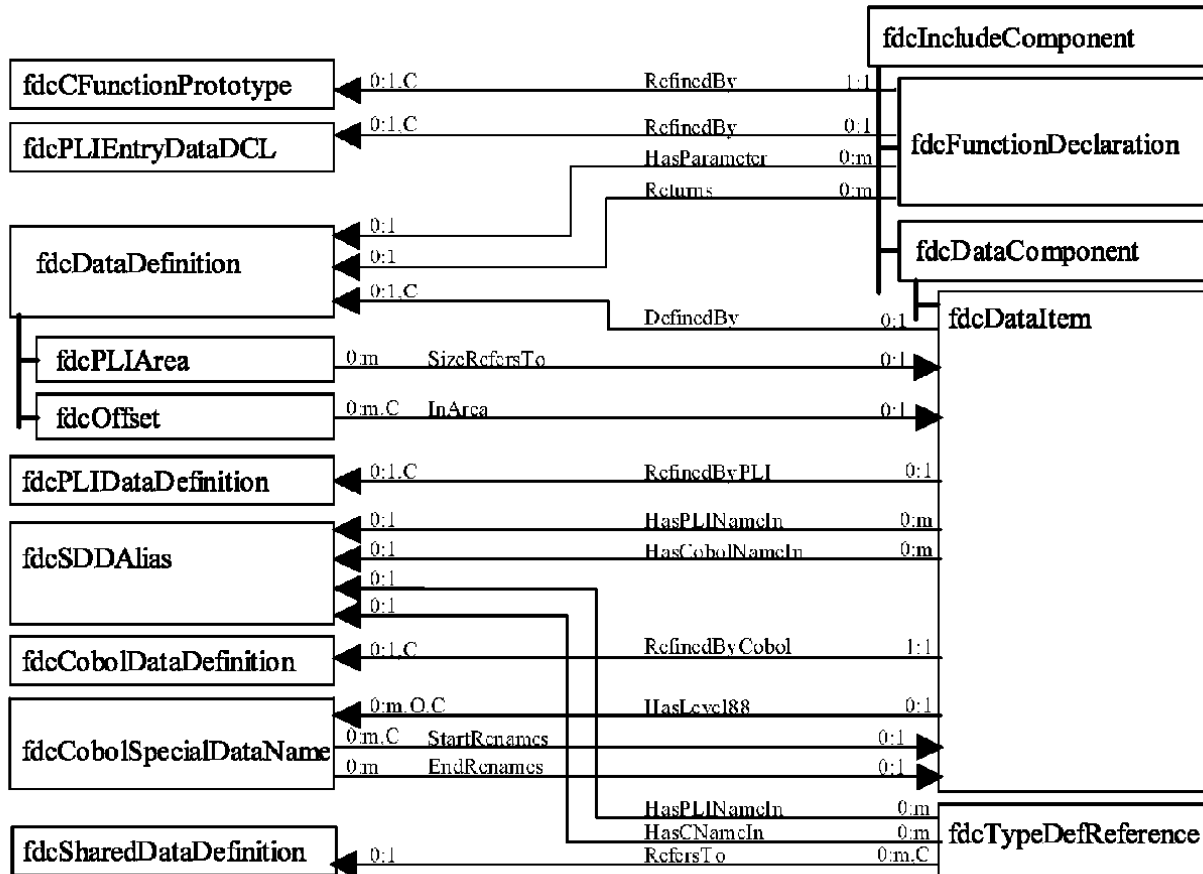


Figure 11. Category Class-Relationships Diagram for Extensions for COBOL and PL/I

VisualAge Requirements Tool

The Requirements Model is designed to help business users determine and specify requirements for business operations.

An operation represents a system either automated or manual that interacts with its environment through business transactions contained in the operation.

The Requirements model contains three components:

- Business facts
- Business information

- Business rules

Business Facts

A business fact is information that must be provided for a transaction to be complete and unambiguous.

Facts provide the terminology needed to describe transactions and information. They are also used in the textual contents of rules because rules are about business objects and their representations, expressed by facts. Facts are not contained in an operation. It is important that all operations within the same enterprise have access to and manage facts and their definitions globally.

Facts can include other facts to allow a single reference to a group of facts, thus offering the possibility of abstracting into aggregate concepts.

Business Information

Business information represents information stored about the business objects that the operation handles. Examples of stored information are:

- Customer information
- Order information
- Policy information

Information is also contained in an operation and has the ability to include facts, which in this case represent the data kept about the business object.

Information elements can have dependencies on each other. For example an Order may depend on Parts, to indicate that an order makes reference to a product.

Business Rules

Business rules are also contained in an operation. Business rules can be static or dynamic. Static rules are needed to define either transactions or information elements. They express properties that are always true about their targets. Dynamic rules are triggered by a transaction. It is the dynamic rules responsibility to explain the behavior caused by that transaction. The behavior is explained in textual form in the rule, with emphasis on what output transactions it initiates, what information elements it interrogates and modifies, and what other rules it uses.

Chapter 2. Global Model

Object Class: ADAnnotatedObject

Inherits From
ADObject

Description
An instance of this object class represents an object that can be associated with annotation text.

Public Class
Yes

Extends
None

Class Extent
None

Cache DLL Name
EWSWGLO

Attributes

Name	Description
------	-------------

pAnnotationText	This attribute specifies an instance of the relationship AnnotObj_Has_AnnotTxt.
------------------------	---

Data Type	relationship to ADAnnotationText
------------------	----------------------------------

Target Class	ADAnnotationText
---------------------	------------------

Target Attribute	pAnnotatedObject
-------------------------	------------------

Cardinality	1..n
--------------------	------

Flags	PRIMARY, CONTROLLING
--------------	----------------------

Cache View Methods

None

Object Class: TCPart

Inherits From
ADAnnotatedObject

Description
This class defines a representation of all user objects that are initially introduced by TeamConnection. This is a superclass of all named objects. Only subclasses of TCPart can serve as the root class for a view. Only subclasses of TCPart can be identified as participants, inputs, or outputs of a Build process. Being a named object, any TCPart subclass instance may

be accessed and manipulated directly through its unique name without navigating through existing relationship instances.

Public Class

Yes

Extends

None

Class Extent

None

Cache DLL Name

EWSWGLO

Attributes

Name	Description
------	-------------

adName	
---------------	--

This attribute specifies the fully-qualified name of this TcPart.

Data Type	
------------------	--

string

Flags	NODATA, USERSETTER
--------------	--------------------

component	
------------------	--

This attribute specifies the component associated with this TcPart.

Data Type	
------------------	--

string

Flags	TCINTERNAL
--------------	------------

lastUpdateTimeStamp	
----------------------------	--

This attribute specifies the timestamp for the latest update to this TcPart.

Data Type	
------------------	--

string

Flags	USERSETTER
--------------	------------

nuDropDate	
-------------------	--

This attribute specifies the deletion date of this TcPart.

Data Type	
------------------	--

string

Flags	TCINTERNAL
--------------	------------

nuAddDate	
------------------	--

This attribute specifies the creation date of this TcPart.

Data Type	
------------------	--

string

Flags	TCINTERNAL
--------------	------------

parms	This attribute specifies the build parameters for this TcPart.
--------------	--

Data Type	
------------------	--

string

parser	
---------------	--

This attribute specifies parsers supported for this TcPart.

Data Type
string

lockedBy
This attribute specifies the user ID of the user who has this TcPart locked.

Data Type
string

Flags TCINTERNAL

isLocked
This attribute specifies whether this TcPart is locked.

Data Type
integer

Valid Values
1 (TRUE) or 2 (FALSE)

Flags TCINTERNAL, USERSETTER

viewTypeList
This attribute specifies a list of views that this TcPart was retrieved by.

Data Type
RSCollection

Flags READONLY, TCINTERNAL

buildStatus
This attribute specifies the build status of this TcPart.

Data Type
string

deleteLog
This attribute specifies the name(s) of the Delete control log(s) used in the processing of cache objects.

Data Type
DSModelCollection

Flags READONLY, TCINTERNAL

bulkdata
This attribute specifies the contents of this TcPart.

Data Type
bindata

type This attribute specifies the subclass of TcPart that this TcPart belongs to.

Data Type
string

transformedBy
This attribute specifies a relationship to a build event that uses this TcPart.

Target Class
FHBTransformedByInfo

Target Attribute
plInput

Relationship Object
TransByInfo2Input

Cardinality

1..n

generatedBy

This attribute specifies a relationship to an object that this TcPart is dependent on.

Target Class

FHBGeneratesInfo

Target Attribute

pOutput

Relationship Object

GenInfo2Output

Cardinality

1..1

applications

This attribute points to an application that this TcPart is a component of.

Target Class

fdcApplicationCollector

Target Attribute

managedObjects

Relationship Object

fdcLinkApplColl2TcPart

Cardinality

m..n

dependsOn

This attribute specifies a relationship to a TcPart that this TcPart is dependent on.

Target Class

TcPart

Target Attribute

referencedBy

Relationship Object

ADLinkTcPartDependsOnTcPart

Cardinality

m..n

commonVersions

This attribute specifies versions that a TcPart is common to, and takes the form:

release1 release2...releasen|workarea1 workarea2...workarean

Data Type

string

Flags READONLY, TCINTERNAL, USERSETTER

Cache View Methods

Name Description**isNamedObject**

This method returns 1 if the object is a named object.

Return Type
integer

Valid Values
1 (TRUE) or 0 (FALSE)

Parameters
None

isViewType

This method returns 1 if the TCPart represents the indicated view type.

Return Type
integer

Valid Values
1 (TRUE) or 0 (FALSE)

Parameters

Name <Direction>Type

&strViewType
<Input > string

isPartialViewType

This method returns 1 if the TCPart represents the indicated view type BUT the entire view instance has not been retrieved into the cache.

Return Type
integer

Valid Values
1 (TRUE) or 0 (FALSE)

Parameters

Name <Direction>Type

&strViewType
<Input > string

list

This method returns a collection of handles to objects satisfying search criteria. The objects are partially represented in the cache for purposes of a query, although they currently reside as complete objects on the server.

Return Type
DSModelCollection *

Flags static

Parameters

Name <Direction>Type

viewTypeList
<Input > string

wildName
<Input > string

strVersion
<Input > string

listCache

This method returns a collection of handles to objects instantiated in the cache that satisfy search criteria.

Return Type
DSModelCollection *

Flags static

Parameters

Name	<Direction>	Type
------	-------------	------

viewList	<Input >	string
-----------------	----------	--------

wildName	<Input >	string
-----------------	----------	--------

strVersion	<Input >	string
-------------------	----------	--------

retrieve

This method retrieves objects identified by a view type into the cache.

Return Type
integer

Parameters

Name	<Direction>	Type
------	-------------	------

viewType	<Input >	string
-----------------	----------	--------

lock	<Input >	integer
-------------	----------	---------

refreshFlag	<Input >	integer
--------------------	----------	---------

forceOption	<Input >	integer
--------------------	----------	---------

strVersion	<Input >	string
-------------------	----------	--------

retrieveObjects

This method retrieves a collection objects identified by a multiple view types into the cache.

Return Type
DSModelCollection *

Flags Static

Parameters

Name	<Direction>	Type
------	-------------	------

list	<Input >	ObjectViewList *
-------------	----------	------------------

lock	<Input >	integer
-------------	----------	---------

refreshFlag	<Input >	integer
--------------------	----------	---------

forceOption	<Input >	integer
--------------------	----------	---------

strVersion	<Input >	string
-------------------	----------	--------

retrieveByName

This method retrieves objects identified by a name and a view type into the cache.

Return Type

DSModelCollection*

Flags Static

Parameters

Name <Direction>Type

viewList

<Input > string

wildName

<Input > string

lock <Input > integer

refreshFlag

<Input > Integer

forceOption

<Input > integer

strVersion

<Input > string

retrieveObjectsByName

This method retrieves a collection of objects identified by names and view types into the cache.

Return Type

DSModelCollection *

Flags Static

Parameters

Name <Direction>Type

list <Input > NameViewList *

lock <Input > integer

refreshFlag

<Input > integer

forceOption

<Input > integer

strVersion

<Input > string

store This method stores cache objects, identified by a view type into the repository.

Return Type

integer

Parameters

Name <Direction>Type

viewType

<Input > string

bCheckTimeStamp
 <Input > integer

forceOption
 <Input > integer

lockOption
 <Input > uINT4

comments
 <Input > string

strVersion
 <Input > string

storeObjects

This method stores multiple cache objects, identified by multiple view types, into the repository.

Return Type
 integer

Flags static

Parameters

Name	<Direction>	Type
list	<Input >	ObjectViewList *
bCheckTimeStamp	<Input >	integer
forceOption	<Input >	integer
lockOption	<Input >	uINT4
comments	<Input >	string
strVersion	<Input >	string

lock This method locks a named object in the repository.

Return Type
 void

Parameters

Name	<Direction>	Type
viewType	<Input >	string
forceOption	<Input >	integer
strVersion	<Input >	string

unlock

This method unlocks a named object in the repository.

Return Type
 void

Parameters

Name <Direction>**Type**

viewType

<Input > string

strVersion

<Input > string

fixCrLf This method makes adjustments to an object's bulk data to prepare it for storage on the file system. The default, CRLF_DEFAULT, adds carriage returns, line feeds, Control-Z characters, and converts tab characters to spaces, as needed.

Return Type

integer

Default

CRLF_DEFAULT

Valid Values

- CRLF_DEFAULT (CRLF_ADDCR | CRLF_CTRLZ | CRLF_EXPANDTAB for Intel, none of the above for UNIX)
- CRLF_ADDCR (adds carriage returns)
- CRLF_CTRLZ (adds end-of-file characters)
- CRLF_ADDTAB (replaces spaces with tabs)
- CRLF_EXPANDTAB (replaces tabs with spaces)

Parameters

Name <Direction>**Type**

flags <Input > integer

Object Class: ADAnnotationText

Inherits From

ADObject

Description

An instance of this object class represents a labeled string that forms the annotation text of an object.

This text string can have a label to indicate the type of the annotation text. The default label is "DESCRIPTION". Possible annotation text labels are:

- DESCRIPTION
- EXAMPLE
- NOTE

The label of an annotation text can be changed from one to another.

Short name: AnntnTxt

Public Class

Yes

Extends

None

Class Extent
None

Cache DLL Name
EWSWGLO

Attributes

Name	Description
pAnnotatedObject	This attribute specifies an instance of the relationship AnnotTxt_Describes_AnnotObj. Target Class ADAnnotatedObject Target Attribute pAnnotationText Relationship Object ADLinkADAnnotObj2AnnotTxt Cardinality n..1
contents	The contents of the annotation text. Data Type string
label	This attributes specifies the label (or type) of the annotation text. Data Type string Default DESCRIPTION

Cache View Methods

None

Object Class: fdcApplicationCollector

Inherits From
TCPart

Description
An instance of this class type is used to identify all of the components of an application or any other logical grouping through the relationship with TCPart.

Public Class
Yes

Extends
None

Class Extent
None

Cache DLL Name
EWSWGLO

Attributes

Name	Description
------	-------------

applicationFunctionAnalyst

This attribute specifies the function analyst for the given application.
--

Data Type string

Default null

applicationSystemAnalyst

This attribute specifies the system analyst for the given application.
--

Data Type string

Valid Values string character

Default null

managedObjects

This attribute points to the TcPart instances which serve as the components of this application.
--

Target Class TcPart

Target Attribute applications

Relationship Object fdcLinkApplColl2TcPart
--

Cardinality m..n

Cache View Methods

None

Relationship Class: fdcLinkApplColl2TcPart

Inherits From
ADLink

Description

An instance of this relationship represents the association between a managed object and the application collector.

Cache DLL Name
EWSWGLO

Attributes

Name	Description
------	-------------

source

Target Class
fdApplicationCollector

Target Attribute
managedObjects

Cardinality
m..n

Flags PRIMARY

target

Target Class
TCTPart

Target Attribute
applications

Cardinality
m..n

Cache View Methods

None

Relationship Class: ADLinkADAnnotObj2AnnotTxt

Inherits From
ADLink

Description
An instance of this relationship represents the association between a managed object and the annotation text.

Attributes

Name	Description
source	
Target Class	ADAnnotatedObject
Target Attribute	pAnnotationText
Cardinality	1..n
Flags	PRIMARY, CONTROLLING
target	
Target Class	ADAnnotationText
Target Attribute	pAnnotatedObject
Cardinality	n..1

Cache View Methods

None

Relationship Class: ADLinkTCPartDependsOnTCPart

Inherits From
ADLink

Description
An instance of this relationship represents the relationship (DependsOn) between two TCParts.

Cache DLL Name
EWSWGLO

Attributes

Name	Description
------	-------------

source

Target Class
TCPart

Target Attribute
dependsOn

Cardinality
m..n

Flags PRIMARY

target

Target Class
TCPart

Target Attribute
referencedBy

Cardinality
m..n

Cache View Methods

None

Chapter 3. Data Atlas Dictionary Model

Object Class: DSAtlasObject

Cache View Name

DSAtlasObject

Storage View Name

DSAtlasObject

Inherits From

TCPart

Description

This class defines a representation of all named objects initially introduced by DataAtlas and the common behaviors and functions supported for those named objects.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name

ewsgaob

DLL Name

atlmod

Attributes

Name	Description
------	-------------

contents	
-----------------	--

This attribute collects instances of the relationship AtlasObj_IsIn_Content.

Data Type

relationship to Content.

Valid Values

0 or more relationship instances

Cache View Methods

Name	Description
------	-------------

copy	This method copies named object in the repository.
-------------	--

Return Type

void

Parameters

Name	<Direction>	Type
------	-------------	------

*objType		
-----------------	--	--

<Input > char

aNewName
 <Input > DSUserObjectName

forceOption
 <Input > int

***comments**
 <Input > char

***strVersion**
 <Input > char

findNamedObj

This method finds named objects.

The purpose of this method is to find an object of the specified view type (strViewName) with the specified name (name) in either the TeamConnection Cache or Repository. Additionally, this method will invoke the _DSAtlasObject::retrieve() method (see below for a description) if told to retrieve it if it is found as indicated by the retrieveFlag. This method first looks in the cache for the object. If the object with the specified name is found in the cache, a new handle to that object is returned. If the object is not found in the cache, a _TCTPart::list is performed to find it in the persistent store. A new handle to the returned object is created and the DSModelCollection returned from the list is deleted. If indicated, _DSAtlasObject::retrieve is invoked. If the pointer returned is non-NULL, the handle represents an "extra" handle and must be cleaned up by the caller.

Return Type

DSAtlasObject *

Parameters

Name <Direction>Type

&strViewName
 <Input > char*

&version
 <Input > char*

&name
 <Input > DSUserObjectName

refreshFlag
 <Input > boolean

retrieveFlag
 <Input > boolean

retrieveLock
 <Input > boolean

findNamedObjects

This method finds named objects.

This method simply invokes the _TCTPart::list method using the input model collection. It is the caller's responsibility to clean up this collection.

Return Type

DSModelCollection *

Parameters

Name <Direction>Type

&aMask
<Input > TStringList
&version
<Input > char*
&wildName
<Input > DSUserObjectName

getConnectableObjectViaMappingTable

This method finds named objects.

Return Type

DSAtlasObject *

Parameters

Name <Direction>Type
***pParmList**
<Input > DSParmList
&strViewTypeName
<Input > string
&strAccessName
<Input > string
refreshFlag
<Input > boolean
retrieveFlag
<Input > boolean
retrieveLock
<Input > boolean

getReconcilableObjects

This method finds named objects.

Return Type

void

Parameters

Name <Direction>Type
****ppRecObjsColl**
<Input > DSCollection
***pParmList**
<Input > DSParmList

deleteFromRepository

This method deletes named objects in aDeleteList from the repository.

Return Type

void

Parameters

Name <Direction>Type
objType
<Input > DSUserObjectType
***deleteList**
<Input > DSModelCollection

forceOption
<Input > BOOL

comments
<Input > IString

strVersion
<Input > IString

getDeleteList

This method returns a list of objects within the scope of the view type indicated that is deleted if the user or tool invokes deleteFromRepository on a set or subset of objects returned from this method.

Return Type
DSModelCollection*

Parameters

Name <Direction>Type

objType
<Input > DSUserObjectType

hasTheSameNameAs

This method returns TRUE if its fullName is the same as that of the object passed in as a parameter; it returns FALSE otherwise.

Return Type
boolean

Parameters

Name <Direction>Type

&anSDD
<Input > DSNamedObject

reconcileViaMappingTable

This is a virtual API that must be overridden at the subclass level.

Return Type
void

Parameters

Name <Direction>Type

***pParmList**
<Input > DSParmList

retrieve

This method retrieves the named object's contents from the repository into the workarea in memory.

Return Type
void

Parameters

Name <Direction>Type

objType
<Input > DSUserObjectType

lock <Input > BOOL

refreshFlag
 <Input > BOOL

forceOption
 <Input > BOOL

strVersion
 <Input > IString

store This method stores the workarea contents (previously retrieved or newly created in memory) into the repository.

Return Type
 void

Parameters

Name	<Direction>	Type
objType	<Input >	DSUserObjectType
bCheckTimeStamp	<Input >	BOOL
forceOption	<Input >	BOOL
lockOption	<Input >	uINT4
comments	<Input >	IString
strVersion	<Input >	IString

Object Class: DSAlias

Cache View Name
 DSAlias

Storage View Name
 DSAlias

Inherits From
 DSAtlasObject

Description
 This object class defines a representation of an alias name of a DataAtlas object in the repository. See its subclasses for details.

Short name: Alias

Public Class
 Yes

Extends
 None

Class Extent
 None

Cache View File Name
 EWSWUIP

Storage View File Name

adals

DLL Name

atlmod

Attributes

Name Description**classNameContext**

This attribute contains the class name of the object that serves as the context for a name uniqueness constraint. For example, if a data structure member that uses a shared data element has a local name, this local name is considered as an alias of the shared data element. If this data name must be unique within a structure, the context would be "DSDataStructure".

Data Type

char*

Valid Values

null or a class name (with DS prefix)

Default

null

Cache View Methods

None

Object Class: DSDeviceObject

Cache View Name

DSDeviceObject

Storage View Name

DSDeviceObject

Inherits From

DSAtlasObject

Description

This is the base class for all device objects, such as Repository, Library, and RDB Catalog. These objects can be opened into a container view on the User Interface.

Short name: DevObj

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWUIP

Storage View File Name

ewsgdev

DLL Name
atlmod

Attributes

Name	Description
------	-------------

pContent	
-----------------	--

This attribute specifies an instance of the relationship DevObj_Has_Content.

Data Type	
------------------	--

relationship to Content.

Valid Values	
---------------------	--

0 or 1 relationship instance

Cache View Methods

Name	Description
------	-------------

add	This method adds one object to the folder content.
------------	--

Return Type	
--------------------	--

DSContent reference

Parameters	
-------------------	--

Name	<Direction>Type
-------------	------------------------------

&anObject	
----------------------	--

<Input > DSAtlasObject

add1	This method adds a collection of objects to the folder content.
-------------	---

Return Type	
--------------------	--

DSContent reference

Parameters	
-------------------	--

Name	<Direction>Type
-------------	------------------------------

&aColOfObjs	
------------------------	--

<Input > DSModelCollection

getObjectList	
----------------------	--

This method returns a list of objects in the content by combining two collections: one from the related Content to DSUIObject, and one from the related Content to NamedObject.

Return Type	
--------------------	--

DSModelCollection *

Parameters	
-------------------	--

None

remove	
---------------	--

This method removes an object from the folder.

Return Type	
--------------------	--

boolean

Parameters	
-------------------	--

Name	<Direction>Type
-------------	------------------------------

***anObject**
<Input > DSAtlasObject

Object Class: DSTechnologyObject

Cache View Name
DSTechnologyObject

Storage View Name
DSTechnologyObject

Inherits From
DSAtlasObject

Description
This is the base class for all DataAtlas technology objects. All named objects that represent technology-based concepts and components are subclasses of TechnologyObject.

Short name: TchObj

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWUIP

Storage View File Name
ewsgtch

DLL Name
atlmod

Attributes

None

Cache View Methods

Name	Description
------	-------------

createSourceFile	This method creates a file containing the generated source.
-------------------------	---

Return Type	void
--------------------	------

Parameters

Name	<Direction>	Type
------	-------------	------

*pParmList	<Input >	DSParmList
-------------------	----------	------------

createSourceFileFromWA	This method creates a file containing the generated source.
-------------------------------	---

Return Type

void

Parameters**Name** <Direction>Type***pParmList**

<Input > DSParmList

generateToLibrary

This method creates a file containing the generated source.

Return Type

void

Parameters**Name** <Direction>Type***pParmList**

<Input > DSParmList

reportOnValidation

This method validates the generated source code.

Return Type

void

Parameters**Name** <Direction>Type***pParmList**

<Input > DSParmList

validate

This method validates a technology object according to the corresponding technology syntax and semantics.

Return Type

void

Parameters**Name** <Direction>Type***pParmList**

<Input > DSParmList

Object Class: DSContent**Cache View Name**

DSContent

Storage View Name

DSContent

Inherits From

DSAtlasObject

Description

An instance of this class represents the collection of nested user objects within a device or technology object. This class is used and maintained by the DataAtlas User Interface to keep track of the nested objects to be displayed.

Short name: Content

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name

ewsgctn

DLL Name

atlmod

Attributes

Name	Description
------	-------------

namedObjects

This attribute collects instances of the relationship Content_Holds_AtlasObj.

This is the collection that the Atlas User Interface maintains to keep track of objects associated with a device or nested objects within a user object.

Data Type

relationship to AtlasObject

Valid Values

0 or more relationship instances

pDeviceObject

This attribute specifies an instance of the relationship Content_BelongsTo_DevObj.

Data Type

relationship to DeviceObject

Valid Values

0 or 1 relationship instance

Update Constraints

(document only)

This attribute and the pTechnologyObject are mutually exclusive.

Cache View Methods

Name	Description
------	-------------

add This method adds one object to the folder content.

Return Type

DSCContent *

Parameters

Name	<Direction>	Type
------	-------------	------

&anObject
<Input > DSAtlasObject

getModel

This method returns the pointer either to the DeviceObject or to the TechnologyObject that has this Content.

Return Type
DSModelObject*

Parameters
None

getObjectList

This method returns a list of objects in the content by combining two collections: one from the related Content to DSUIObject, and one from the related Content to NamedObject.

Return Type
DSModelCollection*

Parameters
None

remove

This method removes an object from the folder.

Return Type
boolean

Parameters
Name <Direction>Type
***anObject**
<Input > DSAtlasObject

Object Class: DSGenericFolder

Cache View Name
DSGenericFolder

Storage View Name
DSGenericFolder

Inherits From
DSDeviceObject

Description
An instance of this object class represents a folder containing other objects that belong to any grouping the user wants to access.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWUIP

Storage View File Name

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Object Class: DSWorkFolder

Cache View Name
DSWorkFolder

Storage View Name
DSWorkFolder

Inherits From
DSGenericFolder

Description
An instance of this object class represents a folder containing other objects that belong to any grouping the user wants to access.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWUIP

Storage View File Name

DLL Name
atlmod

Attributes

Name	Description
------	-------------

versionHistory	A list of versions created for an instance of work folder.
-----------------------	--

Data Type	char*
------------------	-------

Valid Values	null
---------------------	------

Default	null
----------------	------

searchHistory	A list of previous searches performed in this work folder.
----------------------	--

Data Type
char*

Valid Values
null

Default
null

Cache View Methods

None

Object Class: DSMainWindow

Cache View Name
DSMainWindow

Storage View Name
DSMainWindow

Inherits From
DSGenericFolder

Description
This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items...

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWUIP

Storage View File Name

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Object Class: DSRepositoryFolder

Cache View Name
DSRepositoryFolder

Storage View Name
DSRepositoryFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSProfileFolder

Cache View Name

DSProfileFolder

Storage View Name

DSProfileFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSTemplateFolder

Cache View Name

DSTemplateFolder

Storage View Name

DSTemplateFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSQueryFolder

Cache View Name

DSQueryFolder

Storage View Name

DSQueryFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSReportFolder**Cache View Name**

DSReportFolder

Storage View Name

DSReportFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSIMSMVSFolder

Cache View Name

DSIMSMVSFolder

Storage View Name

DSIMSMVSFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSApplicationMVSFolder

Cache View Name

DSApplicationMVSFolder

Storage View Name

DSApplicationMVSFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSApplicationOS2Folder**Cache View Name**

DSApplicationOS2Folder

Storage View Name

DSApplicationOS2Folder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSIMSOS2Folder

Cache View Name
DSIMSOS2Folder

Storage View Name
DSIMSOS2Folder

Inherits From
DSGenericFolder

Description
This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWUIP

Storage View File Name

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Object Class: DSCobolOS2Folder

Cache View Name
DSCobolOS2Folder

Storage View Name
DSCobolOS2Folder

Inherits From
DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSCobolMVSFolder**Cache View Name**

DSCobolMVSFolder

Storage View Name

DSCobolMVSFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSPLIOS2Folder

Cache View Name

DSPLIOS2Folder

Storage View Name

DSPLIOS2Folder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSPLIMVSFolder

Cache View Name

DSPLIMVSFolder

Storage View Name

DSPLIMVSFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSPLIMVSPDS**Cache View Name**

DSPLIMVSPDS

Storage View Name

DSPLIMVSPDS

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSCobolMVSPDS

Cache View Name

DSCobolMVSPDS

Storage View Name

DSCobolMVSPDS

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSApplicationMVSPDS

Cache View Name

DSApplicationMVSPDS

Storage View Name

DSApplicationMVSPDS

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSIMSMVSPDS**Cache View Name**

DSIMSMVSPDS

Storage View Name

DSIMSMVSPDS

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSPLIOS2Directory

Cache View Name

DSPLIOS2Directory

Storage View Name

DSPLIOS2Directory

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSCobolOS2Directory

Cache View Name

DSCobolOS2Directory

Storage View Name

DSCobolOS2Directory

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSApplicationOS2Directory**Cache View Name**

DSApplicationOS2Directory

Storage View Name

DSApplicationOS2Directory

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSIMSOS2Directory

Cache View Name

DSIMSOS2Directory

Storage View Name

DSIMSOS2Directory

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSCobolOS2File

Cache View Name

DSCobolOS2File

Storage View Name

DSCobolOS2File

Inherits From

DSAtlasObject

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSCobolMVSMember

Cache View Name

DSCobolMVSMember

Storage View Name

DSCobolMVSMember

Inherits From

DSAtlasObject

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSPLIOS2File

Cache View Name

DSPLIOS2File

Storage View Name

DSPLIOS2File

Inherits From

DSAtlasObject

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSPLIMVSMember

Cache View Name

DSPLIMVSMember

Storage View Name

DSPLIMVSMember

Inherits From

DSAtlasObject

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSApplicationOS2File**Cache View Name**

DSApplicationOS2File

Storage View Name

DSApplicationOS2File

Inherits From

DSAtlasObject

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSApplicationMVSMember

Cache View Name

DSApplicationMVSMember

Storage View Name

DSApplicationMVSMember

Inherits From

DSAtlasObject

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSIMSOS2File

Cache View Name

DSIMSOS2File

Storage View Name

DSIMSOS2File

Inherits From

DSAtlasObject

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSIMSMVSMember

Cache View Name

DSIMSMVSMember

Storage View Name

DSIMSMVSMember

Inherits From

DSAtlasObject

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSVisualQueryFolder

Cache View Name

DSVisualQueryFolder

Storage View Name

DSVisualQueryFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSSQLQueryFolder

Cache View Name

DSSQLQueryFolder

Storage View Name

DSSQLQueryFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSVisualQueryObject**Cache View Name**

DSVisualQueryObject

Storage View Name

DSVisualQueryObject

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSSQLQueryObject

Cache View Name

DSSQLQueryObject

Storage View Name

DSSQLQueryObject

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSSQLQueryResultsObject

Cache View Name

DSSQLQueryResultsObject

Storage View Name

DSSQLQueryResultsObject

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSVisualQueryResultsFolder

Cache View Name

DSVisualQueryResultsFolder

Storage View Name

DSVisualQueryResultsFolder

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Object Class: DSDB2SubSystem

Cache View Name

DSDB2SubSystem

Storage View Name

DSDB2SubSystem

Inherits From

DSGenericFolder

Description

This class is provided mainly for DataAtlas GUI to use as a mechanism to associate object specific actions or menu items.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWUIP

Storage View File Name**DLL Name**

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDevObj2Content

Cache View Name

DSLinkDevObj2Content

Storage View Name

DSLinkDevObj2Content

Description

An instance of this relationship represents the association between a Device Object and the User Interface Content that holds the nested user objects.

Source Information:**Class Name**

DSDeviceObject

Attribute

pContent

Target Information:**Class Name**

DSContent

Attribute

pDeviceObject

Forward Mapping Semantics:**Verb** Has**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** BelongsTo**Cardinality**

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWUIP

Storage View File Name

ewsgl05

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkContent2AtlasObj**Cache View Name**

DSLinkContent2AtlasObj

Storage View Name

DSLinkContent2AtlasObj

Description

An instance of this relationship represents the association between a User Interface Content and the nested user object.

Source Information:**Class Name**

DSContent

Attribute

namedObjects

Target Information:**Class Name**

DSAtlasObject

Attribute

contents

Forward Mapping Semantics:**Verb** Holds**Cardinality**

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsIn**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWUIP

Storage View File Name

ewsgl07

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Chapter 4. VisualAge Generator Model

Object Class: VGOBJECT

Cache View Name

VGOBJECT

Inherits From

TCPart

Description

This is an abstract superclass of all VisualAge Generator objects included in a VisualAge Generator library MSL. This class is needed to maintain name uniqueness across all VisualAge Generator part types in a release. Please see subclasses for details.

Short name: VGOBJECT

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EZE20VGX

Storage View File Name

ezevc01

DLL Name

atlmod

Attributes

Name	Description
------	-------------

formatLvl	
------------------	--

This attribute represents the format level of a VisualAge Generator part.

Data Type

unsigned long

pVGReferences

This attribute specifies an instance of the relationship
VGOBJECT_VGReferences_VGOBJECT

Data Type

relationship to VGOBJECT

Valid Values

TRUE/FALSE

pVGReferencedBy

This attribute specifies an instance of the relationship
VGOBJECT_VGReferencedBy_VGOBJECT

Data Type

relationship to VGOBJECT

Valid Values

O or M relationship instance

adName

This attribute is inherited from TCGPart.

Data Type

inherited

Index ordered, no_duplicates

Cache View Methods

Name Description**VGObjectInit**

This instance method overrides init.

Return Type

void

Parameters

None

Object Class: VGApplication

Cache View Name

VGApplication

Inherits From

VGObject

Description

This object class defines a representation of a VisualAge Generator Application part

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic (A-Z).
- The remaining characters are alphanumeric (A-Z, 0-9).
- The name cannot contain blanks or have an EZE prefix.
- The name cannot be a DBCS name.
- Maximum length of 7 characters

Short name: VGAppl

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EZE20VGX

Storage View File Name

ezevc02

DLL Name

atlmod

Attributes

Name Description**memberData**

This attribute represents data describing the application in a binary format understood by VisualAge Generator Developer.

Data Type

sequence<octet>

adName

This attribute is inherited from TcPart.

Data Type

inherited

Index ordered, no_duplicates

Cache View Methods

None

Object Class: VGDataItem

Cache View Name

VGDataItem

Inherits From

VGObject

Description

This object class defines a representation of a VisualAge Generator data item part. An instance of this class is linked to an instance of Shared Data Element via the TcPart_DependsOn_TcPart relationship.

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic or national (A-Z, \$, #, @).
- The remaining characters are alphanumeric or national characters, hyphens, or underscores (A-Z, 0-9, \$, #, @, -, _).
- The name cannot contain blanks or have an EZE prefix.
- The name can be a DBCS name up to 15 DBCS characters long with no embedded blanks.

Short name: VGDataItem**Public Class**

Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EZE20VGX

Storage View File Name
ezevc03

DLL Name
atlmod

Attributes

Name	Description
pDEAliasAsVGName	This attribute specifies an instance of the relationship VGDataItem_HasVGNameIn_DEAlias.
Data Type	relationship to DataElementAlias
Valid Values	0 or 1 relationship instance
adName	This attribute is inherited from TCPart.
Data Type	inherited
Index	ordered, no_duplicates

Cache View Methods

Name	Description
associateWithDataElement	This instance method adds a TCPart_DependsOn_TCPart relationship between the VisualAge Generator Data Item and the Shared Data Element on which it depends.
Return Type	DSDataElement*
Parameters	
Name <Direction>Type	
pAssocDataElement	<Input > DSDataElement
versionString	<Input > string
createSharedDataItem	This instance method creates a new DataElement instance with the same name as the VGDataItem if an existing DataElement instance was not provided as input. The current VGDataItem instance is then related to the new or input DataElement via the TCPart_DependsOn_TCPart relationship.

Return Type

void

Parameters

Name <Direction>Type

***pAssocDataElement**

<Input > DSDDataElement*

versionString

<Input > string

getAssocDataElementName

This instance method returns the name of the Shared Data Element on which this VisualAge Generator Data Item is dependant.

Return Type

string

Parameters

None

setDataElementAlias

This instance method creates a DataElementAlias if the VGDataltem has a different name from the DataElement.

Return Type

void

Parameters

Name <Direction>Type

***pAssocDataElement**

<Input > DSDDataElement*

versionString

<Input > string

Object Class: VGGUI

Cache View Name

VGGUI

Inherits From

VGObject

Description

This object class defines a representation of a VisualAge Generator GUI part.

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic.
- The remaining characters are alphanumeric (A-Z, 0-9).
- The name cannot contain blanks or have an EZE prefix.
- The name cannot be a DBCS name.
- Length of 8

Short name: VGGUI

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EZE20VGX

Storage View File Name

ezevc04

DLL Name

atlmod

Attributes

Name Description

memberData

This attribute represents data describing the GUI in a binary format understood by VisualAge Generator Developer.

Data Type

sequence<octet>

adName

This attribute is inherited from TCGPart.

Data Type

inherited

Index ordered, no_duplicates

Cache View Methods

None

Object Class: VGProcess

Cache View Name

VGProcess

Inherits From
VGObject

Description
This object class defines a representation of a VisualAge Generator Process part.

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic or national (A-Z, \$, #, @).
- The remaining characters are alphanumeric or national characters, hyphens, or underscores (A-Z, 0-9, \$, #, @, -, _).
- The name cannot contain blanks or have an EZE prefix.
- The name can be a DBCS name up to 8 DBCS characters long with no embedded blanks.
- Length of 18

Short name: VGProc

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EZE20VGX

Storage View File Name
ezevc05

DLL Name
atlmod

Attributes

Name	Description
------	-------------

memberData	This attribute represents data describing the process in a binary format understood by VisualAge Generator Developer.
-------------------	---

Data Type	sequence<octet>
------------------	-----------------

adName	This attribute is inherited from TCGPart.
---------------	---

Data Type	inherited
------------------	-----------

Index	ordered, no_duplicates
--------------	------------------------

Cache View Methods

None

Object Class: VGPSB

Cache View Name
VGPSB

Inherits From
VGOBJECT

Description
This object class defines a representation of a VisualAge Generator PSB part. If the VisualAge Generator PSB is shared with a DSPSB, then the instance of this class is linked to an instance of DSPSB via the TcPart_DependsOn_TcPart relationship.

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic or national (A-Z, \$, #, @).
- The remaining characters are alphanumeric or national characters (A-Z, 0-9, \$, #, @).
- The name cannot contain blanks or have an EZE prefix.
- The name cannot be a DBCS name.
- Length of 8

Short name: VGPSB

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EZE20VGX

Storage View File Name
ezevc06

DLL Name
atlmod

Attributes

Name	Description
------	-------------

memberData	
-------------------	--

	This attribute represents data describing the PSB in a binary format understood by VisualAge Generator Developer.
--	---

Data Type
sequence<octet>

adName
This attribute is inherited from TcPart.

Data Type
inherited

Index ordered, no_duplicates

Cache View Methods

Name **Description**

associateWithDSPSB

This instance method adds a DependsOn relationship between the VGPSB and the DSPSB on which it depends.

Return Type
DSPSB*

Parameters

Name **<Direction>Type**

***pAssocDSPSB**
<Input > DSPSB*

getAssocIMSPSBName

This instance method returns the name of the DSPSB on which this VisualAge Generator PSB is dependent. If this VisualAge Generator PSB is not dependent on a DSPSB, then a null string is returned.

Return Type
void

Parameters

Name **<Direction>Type**

***pIMSPSBName**
<Input > string

Object Class: VGRecord

Cache View Name
VGRecord

Inherits From
VGObject

Description

This object class defines a representation of a VisualAge Generator record part.

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic or national (A-Z, \$, #, @).
- The remaining characters are alphanumeric or national characters, hyphens, or underscores (A-Z, 0-9, \$, #, @, -, _).
- The name cannot contain blanks or have an EZE prefix.
- The name can be a DBCS name up to 8 DBCS characters long with no embedded blanks.
- Length of 18

Short name: VGRecord

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EZE20VGX

Storage View File Name

ezevc07

DLL Name

atlmod

Attributes

Name Description

memberData

This attribute represents data describing the record in a binary format understood by VisualAge Generator Developer.

Data Type

sequence<octet>

adName

This attribute is inherited from TcPart.

Data Type

inherited

Index ordered, no_duplicates

Cache View Methods

None

Object Class: VGStmtGroup

Cache View Name

VGStmtGroup

Inherits From

VGObject

Description

This object class defines a representation of a VisualAge Generator Statement Group part.

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic or national (A-Z, \$, #, @).
- The remaining characters are alphanumeric or national characters, hyphens, or underscores (A-Z, 0-9, \$, #, @, -, _).
- The name cannot contain blanks or have an EZE prefix.
- The name can be a DBCS name up to 8 DBCS characters long with no embedded blanks.
- Length of 18

Short name: VGStGrp

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EZE20VGX

Storage View File Name

ezevc08

DLL Name

atlmod

Attributes

Name Description**memberData**

This attribute represents data describing the statement group in a binary format understood by VisualAge Generator Developer.

Data Type

sequence<octet>

adName

This attribute is inherited from TCPart.

Data Type
inherited
Index ordered, no_duplicates

Cache View Methods

None

Object Class: VGTable

Cache View Name
VGTable

Inherits From
VGObject

Description
This object class defines a representation of a VisualAge Generator Table part.

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic (A-Z).
- The remaining characters are alphanumeric (A-Z, 0-9).
- The name cannot contain blanks or have an EZE prefix.
- The name cannot be a DBCS name.
- The name cannot end with a zero.
- Length of 7

Short name: VGTable

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EZE20VGX

Storage View File Name
ezevc09

DLL Name
atlmod

Attributes

Name	Description
------	-------------

memberData

This attribute represents data describing the table in a binary format understood by VisualAge Generator Developer.

Data Type

sequence<octet>

adName

This attribute is inherited from TCTPart.

Data Type

inherited

Index ordered, no_duplicates

Cache View Methods

None

Object Class: VGMap

Cache View Name

VGMap

Inherits From

VGObject

Description

This object class defines a representation of a VisualAge Generator Map part.

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic or national (A-Z, \$, #, @).
- The remaining characters are alphanumeric or national characters (A-Z, 0-9, \$, #, @).
- The name cannot contain blanks or have an EZE prefix.
- The name cannot be a DBCS name.
- Length of 8

Short name: VGMap

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EZE20VGX

Storage View File Name

ezevc10

DLL Name
atlmod

Attributes

Name	Description
------	-------------

memberData	
-------------------	--

This attribute represents data describing the map in a binary format understood by VisualAge Generator Developer.

Data Type	
------------------	--

sequence<octet>

adName	
---------------	--

This attribute is inherited from TcPart.

Data Type	
------------------	--

inherited

Index	ordered, no_duplicates
--------------	------------------------

Cache View Methods

None

Object Class: VGMapGroup

Cache View Name	
------------------------	--

VGMapGroup

Inherits From	
----------------------	--

VGObject

Description	
--------------------	--

This object class defines a representation of a VisualAge Generator Map Group part.

Constraints

Values for the adName attribute must adhere to the following rules.

- The first character is alphabetic (A-Z).
- The remaining characters are alphanumeric (A-Z, 0-9).
- The name cannot contain blanks or have an EZE prefix.
- The name cannot be a DBCS name.
- Length of 6

Short name: VGMapGrp

Public Class	
---------------------	--

Yes

Extends	
----------------	--

None

Class Extent	
---------------------	--

Versioned

Cache View File Name
EZE20VGX

Storage View File Name
ezevc11

DLL Name
atlmod

Attributes

Name **Description**

memberData

This attribute represents data describing the map group in a binary format understood by VisualAge Generator Developer.

Data Type
sequence<octet>

adName

This attribute is inherited from TCGPart.

Data Type
inherited

Index ordered, no_duplicates

Cache View Methods

None

Object Class: VGDataDef

Cache View Name
VGDataDef

Inherits From
VGObject

Description

An instance of this class represents the VisualAge Generator unique data needed to be able to describe General and Numeric edit characteristics.

Short name: VGDataDef

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EZE20VGX

Storage View File Name
ezevc12

DLL Name
atlmod

Attributes

Name	Description
------	-------------

pSimpleTypeDef	
-----------------------	--

This attribute specifies an instance of the relationship SmpIType_RefinedBy_VGDatDef.

Data Type	
------------------	--

relationship to SimpleTypeDef

Valid Values	
---------------------	--

0 or 1 relationship instance

genInvalidMsg	
----------------------	--

This attribute specifies the error message number to use when an application user enters data that is not compatible with the data type defined for the data element.

Data Type	
------------------	--

short

Valid Values	
---------------------	--

Valid values are in the range -9999 to 9999.

Default	
----------------	--

0

genMinInputMsg	
-----------------------	--

This attribute specifies the error message number to use when an application user does not enter the minimum number of characters required for a data element that has the minimum input edit specified.

Data Type	
------------------	--

short

Valid Values	
---------------------	--

Valid values are in the range -9999 to 9999.

Default	
----------------	--

0

genReqMsg	
------------------	--

This attribute specifies the error message number to use when an application user does not enter data in a field for which the input required edit was specified.

Data Type	
------------------	--

short

Valid Values	
---------------------	--

Valid values are in the range -9999 to 9999.

Default	
----------------	--

0

genEditMsg	
-------------------	--

This attribute specifies the error message number to use when an application user enters data that fails a modulus check or table edit check.

Data Type	
------------------	--

short

Valid Values

Valid values are in the range -9999 to 9999.

Default

0

genMinInput

This attribute specifies the minimum number of characters the application user must enter if the application user types data in the field.

Data Type

char

Valid Values

Valid values are in the range -9999 to 9999.

Default

0

genFill

This attribute specifies the character which will fill unused positions in a field which is displayed or printed on a map.

Data Type

char

Valid Values

A fill character can be an alphanumeric character, a blank, or a null (binary zeroes). For DBCS data, only blanks and null characters are allowed. For mixed data, only SBCS characters are allowed.

genJustify

This attribute specifies the alignment of data in a field when it is displayed on a map.

Data Type

char

Valid Values

Valid values are Left, Right, and None.

Default

For Character data, left justification.

For Numeric data, right justification.

Update Constraints**(document only)**

Right justification is required for numeric fields that also have a decimal position or sign edit specified.

genRoutine

This attribute specifies the name of a routine or edit table to be used for special editing of data that is entered by the application user in a variable field.

Data Type

string<19>

Valid Values

The following is a list of valid values:

- EZEC10
- EZEC11

- Statement group name.
- Match valid table name.
- Match invalid table name.
- Range match table name.

Default

Blank

genDateEdit

This attribute specifies the format for date edits in a VisualAge Generator map field or a data item.

Data Type

string<11>

Valid Values

The following is a list of valid values:

- SYSGREGRN.
- SYSJULIAN.
- YY, MM, and DD in any order separated by a single-byte character other than M, Y, or D.
- YYYY, MM, and DD in any order separated by a single-byte character other than M, Y, or D.
- YY and DDD in any order separated by a character other than Y or D.
- YYYY and DDD in any order separated by a character other than Y or D.

Default

Blank

Update Constraints

(document only)

Type must NOT be DBCS, MIX, or HEX. Length must be long enough to hold the characters plus the separators. Decimals, currency symbol, numeric separator, and sign must not be specified.

genInputReq

This attribute specifies that the VisualAge Generator application user must enter data in the VisualAge Generator map field.

Data Type

boolean

Valid Values

TRUE/FALSE

Default

FALSE

genHexEdit

This attribute specifies that the VisualAge Generator application user is required to enter only hexadecimal characters in the map variable field.

Data Type

boolean

Valid Values

TRUE/FALSE

Default

FALSE

Update Constraints

(document only)

For types of CHAR or HEX only

genFold

This attribute specifies that the alphabetic characters in a field are changed to uppercase when a map containing the field is entered.

Data Type

boolean

Valid Values

TRUE/FALSE

Default

FALSE

Update Constraints

(document only)

Not available for types DBCS, NUM, NUMC, PACK, PACF, BIN.

genSoSiSpace

This attribute specifies that the mixed data entered in a field on an ASCII device should be checked to ensure that the data, when converted to the mainframe SO/SI format, will still fit in a field of the same length.

Data Type

boolean

Valid Values

TRUE/FALSE

Default

TRUE

Update Constraints

(document only)

For Mixed type only.

numRecFieldLen

The value of this attribute is the recommended field length that will be needed for a map variable field in order to display the largest numeric value for the data item, including any editing characteristics.

Data Type

unsigned short

Update Constraints

(document only)

This is not a user input. VisualAge Generator specifies it according to the settings of the length, decimals, currency symbol, numeric separator, and sign values.

numRangeMsg

This attribute specifies the error message number to use when a VisualAge Generator application user enters data that fails the minimum or maximum value check.

Data Type

short

Valid Values

Valid values are in the range -9999 to 9999.

Default

0

numSign

This attribute specifies the position of the sign when data is displayed in the VisualAge Generator map variable field.

Data Type

char

Valid Values

L (leading), T (trailing), or null (none)

Default

L (leading)

Update Constraints**(document only)**

If leading or trailing, genJustify must be set to Right. If leading or trailing, genDateEdit should be null.

numMin

This attribute specifies the smallest number a VisualAge Generator application user can enter into the variable field.

Data Type

string<21>

Valid Values

Numeric values with optional decimal point and leading sign.

Number of digits, ignoring the sign, currency symbol, numeric separator, decimal point, should be less than or equal to the length.

Default

0

Update Constraints**(document only)**

Must be less than or equal to numMax.

numMax

This attribute specifies the largest number a VisualAge Generator application user can enter into the variable field.

Data Type

string<21>

Valid Values

Numeric values with optional decimal point and leading sign.

Number of digits, ignoring the sign, currency symbol, numeric separator, decimal point, should be less than or equal to the length.

Default

0

Update Constraints

(document only)

Must be greater than or equal to numMin.

numCurrSym

This attribute specifies that a currency symbol should be displayed or accepted as input in a VisualAge Generator map variable field.

Data Type

boolean

Valid Values

TRUE/FALSE

Default

FALSE

Update Constraints

(document only)

If ON, genDateEdit must be blank.

numNumSep

This attribute specifies that a numeric separator should be displayed or accepted as input in a VisualAge Generator map variable field.

Data Type

boolean

Valid Values

TRUE/FALSE

Default

FALSE

Update Constraints

(document only)

If ON, genDateEdit must be blank.

numZeroEdit

This attribute specifies how zero values are displayed in numeric fields. If the value is TRUE, then a zero value is displayed as the number zero; otherwise, a zero value is displayed as a character blank.

Data Type

boolean

Valid Values

TRUE/FALSE

Default

FALSE

Cache View Methods

None

Relationship Class: DSLinkSmpType2VGDatDef

Cache View Name

DSLinkSmpType2VGDatDef

Description

An instance of this relationship represents the association between a SimpleTypeDef and a VGDataDef

Source Information:**Class Name**

DSSimpleTypeDef

Attribute

pVGDataDef

Target Information:**Class Name**

VGDataDef

Attribute

pSimpleTypeDef

Forward Mapping Semantics:

Verb RefinedBy

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Refines

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EZE20VGX

Storage View File Name

ezevl01

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkVGDatltn2DEAlias

Cache View Name

DSLLinkVGDatltn2DEAlias

Description

An instance of this relationship represents the association between a Shared Data Element and its VisualAge Generator name.

Source Information:

Class Name

VGDataItem

Attribute

pDEAliasAsVGName

Target Information:

Class Name

DSDataElementAlias

Attribute

VGLocalNames

Forward Mapping Semantics:

Verb HasVGNameIn

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb IsVGNameFor

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EZE20VGX

Storage View File Name

ezevl02

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkVGOBJECT2VGOBJECT

Cache View Name

DSLinkVGOBJECT2VGOBJECT

Description

An instance of this relationship is used so that object SQL queries can return a result set of VGOBJECTs that are instances of different VisualAge Generator part types.

Source Information:**Class Name**

VGOBJECT

Attribute

pVGReferences

Target Information:**Class Name**

VGOBJECT

Attribute

pVGReferencedBy

Forward Mapping Semantics:

Verb VGReferences

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb VGReferencedBy

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EZE20VGX

Storage View File Name

ezevl03

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Chapter 5. Conceptual Data Modeler Model

Object Class: DSEntityType

Cache View File Name

cdmwpcm_

Storage View File Name

cdm004

DLL Name

cdmDLL

Public Class

Yes

Storage View Name

DSEntityType

Short Name

Ent

Inherits From

ADManagedObject

Extends

None

Class Extent

Versioned

Description

This abstract class defines a representation of the common properties of Fundamental Entity Types (entity types in the strict sense of E/R modeling), and of Associated Entity Types which are needed for the relational implementation of complex relationships.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Object Class: DSAssociatedEntityType

Cache View File Name

cdmwpcm_

Storage View File Name

cdm001

DLL Name

cdmDLL

Public Class

Yes

Storage View Name

DSAssociatedEntityType

Short Name

AssEnt

Inherits From

DSEntityType

Extends

None

Class Extent

Versioned

Description

An instance of this object class represents a table that is needed for the relational implementation of complex relationships. Relationships are considered complex if they carry attributes, are many-to-many, or are n-ary. Associated Entity Type is not directly modeled by a data modeler in a data model, but its existence is derived from the properties of modeled relationships.

An example of an instance of this object class is a table that would be needed for the implementation of a relationship "people invest in stock x-amount of dollars, where "x-amount of dollars" is an attribute of the relationship "invest in".

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Object Class: DSAttribute

Cache View File Name

cdmwpcm_

Storage View File Name

cdm002

DLL Name

cdmDLL

Public Class

Yes

Storage View Name

DSAttribute

Short Name

Attr

Inherits From

ADAnnotatedObject

Extends

None

Class Extent

Versioned

Description

An instance of this object class represents an attribute of an entity type or relationship type in E/R modeling.

An example of an instance of this object class is "Name" as it applies to an instance of Entity Type "Person".

Attributes

Name Description

name This attribute associates a name with the instance of object class Attribute, which identifies it to the tool user.

Data Type

string <64>

Update Constraints**Function****Description****cdm_AttrNameUnique (cdmatrun)**

This constraint ensures that the attributes characterizing one instance of object class Entity Type have all different names.

targetTechnologyName

This attribute can be used by a tool to deposit the name under which the construct generated from an instance of this entity type would be accessed in the target data base.

Data Type

string <19>

isPrimary

This attribute indicates whether or not the instance of object class Attribute is to be used as the primary key in a relational implementation of the instance of object class Entity Type, which it characterizes; or if it contributes to the primary key if the the key is composite.

If any of the attributes contributing to the primary key are simultaneously also foreign keys and these foreign keys refer to what are attributes constituting the primary key of one and the same (but other) instance of Entity Type, then this (the original) instance of Entity Type is an "attributive", "owned", or "weak" entity type; which tools may choose to represent differently.

Data Type

boolean

isNotNull

This attribute specifies whether or not the instance of object class Attribute, when materialized in a database, must or need not contain a value.

Data Type

boolean

Cache View Methods

None

Storage View Methods

None

Constraints

None

Object Class: DSDataElement

Cache View File Name

ewswsdd

Storage View File Name

ewssc05

DLL Name

atlmod

Public Class

Yes

Storage View Name

DSDataElement

Short Name

DataElmn

Inherits From

ADManagedObject

Extends

None

Class Extent

Versioned

Description

An object of this class represents a shareable named data element.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Object Class: DSDataModel

Cache View File Name

cdmwpcm_

Storage View File Name

cdm003

DLL Name

cdmDLL

Public Class

Yes

Storage View Name

DSDataModel

Short Name

Model

Inherits From

ADManagedObject

Extends

None

Class Extent

Versioned

Description

An instance of this object class represents a set of entity types and relationship types that collectively form a data model.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Object Class: DSAggregate

Cache View File Name

cdmwpcm_

Storage View File Name

cdm018

DLL Name

cdmDLL

Public Class

Yes

Storage View Name

DSAggregate

Short Name

Aggr

Inherits From

ADManagedObject

Extends

None

Class Extent

Versioned

Description

This object class provides the means to partition a Data Model instance into: non-overlapping sets of Entity Type, Relationship Type, and Relationship Type Bundle instances; or, in a recursive manner, into sets of such sets.

Each instance of this object class represents one such set.

Relationship Type and Relationship Type Bundle instances are further constrained.

Relationship Type instances must belong to an Aggregate instance which contains at least one of the linked to Entity Type instances.

Relationship Type Bundle instances must belong to an aggregate which contains the Entity Type instance that is source of the bundle; or which contains at least one of the relationships interrelated by the bundle.

Attributes

Name	Description
------	-------------

targetTechnologyName

This attribute can be used by a tool to deposit the name under which the construct generated from an instance of this entity type would be accessed in the target data base.

Data Type

string <19>

Cache View Methods

None

Storage View Methods

None

Constraints

None

Object Class: DSFundamentalEntityType**Cache View File Name**

cdmwpcm_

Storage View File Name

cdm005

DLL Name

cdmDLL

Public Class

Yes

Storage View Name

DSFundamentalEntityType

Short Name

FundEnt

Inherits From

DSEntityType

Extends

None

Class Extent

Versioned

Description

An instance of this object class represents an entity type as modeled in a data model.

An example of an instance of this object class is "Person".

Attributes**Name Description**

role This attribute describes the role instances of this entity type are to play in

the data model, which if properly enforced, can be used by a tool to generate specific forms of referential constraints in the target data base.

Data Type

enum fdcDescriptionType

Valid Values

Value	Description
-------	-------------

fundamental	fundamental entity type
--------------------	-------------------------

attributive	attributive entity type
--------------------	-------------------------

associative	associative entity type
--------------------	-------------------------

targetTechnologyName

This attribute can be used by a tool to deposit the name under which the construct generated from an instance of this entity type would be accessed in the target data base.

Data Type

string <19>

rank This attribute describes the importance instances of this entity type have with respect to structuring the model.

Data Type

enum fdcEnumEntityRank

Valid Values

Value	Description
-------	-------------

noRank	of no distinguishing rank
---------------	---------------------------

focal	of central importance
--------------	-----------------------

Cache View Methods

None

Storage View Methods

None

Constraints

None

Object Class: DSRelationshipBundle

Cache View File Name

cdmwpcm_

Storage View File Name

cdm006

DLL Name
cdmDLL

Public Class
Yes

Storage View Name
DSRelationshipBundle

Short Name
RelBundle

Inherits From
ADManagedObject

Extends
None

Class Extent
Versioned

Description

An instance of this object class represents relationships which have their instances constrained through mutual dependencies expressed on the type level. Participation in a bundle requires that the relationships have one entity type in common (the source of the relationship bundle) and that they all have a minimum cardinality of zero on the links to the non-source entity types; in other words, that the mapping from the source entity type to the entity types singled out through a participating relationship need not necessarily be instantiated (otherwise, mandatory instantiation would override the inter-relationship dependencies).

Relationship bundles have two possible interpretations: as either a bundle of relationships with dependencies between them, as described above; or, when maximum cardinalities are 1 on all links, the bundle may mean a specialization/generalization or subtype/supertype, these considered synonymous with, and being subsumed under the more general term "categorization". The intended meaning is captured in the "semantic" attribute below; it will further reflect in how bundles are represented by tools on the glass.

Attributes

Name Description

semantic

This attribute records the intended meaning (per the intention of the tool user) of a relationship bundle as: either a set of relationships with constraints between them; or as a categorization in which the source entity type is "specialized" in terms of the target entity types of the relationships in the bundle (a generalization when seen in the reverse directions), or as a subtyping of the source entity type considered supertype.

In the case of mutually constrained relationships the attribute "name" inherited from ADManagedObject will serve a special purpose: it will contain an integer number. This number is to be associated with all participating relationships and singles them out as belonging to the bundle. The number is unique only with respect to bundles emerging from one and the same source entity type.

Data Type

enum fdcRelBundleTypes

Valid Values

Value	Description
-------	-------------

categorization	Relationships have a maximum cardinality of one, and are intended to express specialization / generalization, subtype / supertype etc. concepts.
-----------------------	--

interdependentRels	Relationships are constrained as to how many of them may be instantiated simultaneously at any one point in time.
---------------------------	---

Default

interdependentRels

Update Constraints

Function	Description
----------	-------------

cdmCatConstraints (cdmcatcs)	This constraint ensures that categorization type relationship bundles do not have their interrelationshipsInstantiated constraint set to noneOrAll or everyOne.
-------------------------------------	---

relationshipsInstantiated

This attribute defines how many of the bundle member relationships may be, or must be, instantiated when seen from the source entity type.

Data Type

enum fdcInterRelConstraints

Valid Values

Value	Description
-------	-------------

zeroOrOne	Instantiation is optional. But if one relationship is instantiated it excludes instantiation of all others.
------------------	---

preciselyOne	Instantiation is mandatory. Instantiation of one relationship excludes instantiation of all others.
---------------------	---

zeroOrMore	Instantiation is optional. Any number of relationships may be instantiated.
-------------------	---

This is effectively no constraint at all. A bundle of this type, however, may serve a documentation purpose in the data model.

oneOrMore	Instantiation is mandatory. At least one relationship must be instantiated.
------------------	---

noneOrAll	Instantiation is optional. But if one relationship is instantiated, all must be instantiated.
------------------	---

everyOne	Instantiation is mandatory and pertains to all relationships.
-----------------	---

This is effectively a constraint that can be achieved otherwise (by making minimum cardinalities of relationships all one). A bundle, so constrained, may however serve a documentation purpose in the data model.

Default

preciselyOne

Update Constraints

Function

Description

cdmCatConstraints (cmdcatcs)

This constraint ensures that categorization type relationship bundles do not have their interrelationshipsInstantiated constraint set to noneOrAll or everyOne.

Cache View Methods

None

Storage View Methods

None

Constraints

None

Object Class: DSRelationshipType

Cache View File Name

cdmwpcm_

Storage View File Name

cdm007

DLL Name

cdmDLL

Public Class

Yes

Storage View Name

DSRelationshipType

Short Name

Rel

Inherits From

ADManagedObject

Extends

None

Class Extent

Versioned

Description

An instance of this object class represents a relationship as modeled in an E/R data model.

Attributes**Name Description****targetTechnologyName**

This attribute can be used by a tool to deposit the name under which the construct generated from an instance of this entity type would be accessed in the target data base.

Data Type

string <19>

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSEnt_characterizedBy_Attr
Cache View File Name

cdmwpcm_

Storage View File Name

cdm008

DLL Name

cdmDLL

Storage View Name

DSEnt_characterizedBy_Attr

Source

Source Information:

Class Name

DSEntityType

Attribute

attributes

Target Target Information:**Class Name**

DSAttribute

Attribute

entityType

Forward Mapping

Forward Mapping Semantics:

Verb characterizedBy

Cardinality
0..m

Ordered
Yes

Controlling
Yes

Inverse Mapping
Inverse Mapping Semantics:

Verb characterizes

Cardinality
1..1

Ordered
No

Controlling
No

Description

An instance of this relationship class represents the characterization of an entity type by an attribute (as part of the modeling of the entity type).

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSRel_classOfConceptsIn_Model

Cache View File Name
cdmwpcm_

Storage View File Name
cdm009

DLL Name
cdmDLL

Storage View Name
DSRel_classOfConceptsIn_Model

Source
Source Information:

	Class Name	DSRelationshipType
	Attribute	dataModel
Target	Target Information:	
	Class Name	DSDataModel
	Attribute	relationshipTypes
	Forward Mapping	
	Forward Mapping Semantics:	
	Verb	classOfConceptsIn
	Cardinality	1..1
	Ordered	No
	Controlling	No
	Inverse Mapping	
	Inverse Mapping Semantics:	
	Verb	contains
	Cardinality	0..m
	Ordered	No
	Controlling	No
	Description	
	An instance of this relationship class represents the contribution a modeled relationship type makes to a data model.	

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSEnt_classOfThingsIn_Model

Cache View File Name

cdmwpcm_

Storage View File Name

cdm010

DLL Name

cdmDLL

Storage View Name

DSEnt_classOfThingsIn_Model

Source

Source Information:

Class Name

DSEntityType

Attribute

dataModel

Target Target Information:**Class Name**

DSDataModel

Attribute

entityTypes

Forward Mapping

Forward Mapping Semantics:

Verb classOfThingsIn

Cardinality

1..1

Ordered

No

Controlling

No

Inverse Mapping

Inverse Mapping Semantics:

Verb contains

Cardinality

0..m

Ordered

No

Controlling

No

Description

An instance of this relationship class represents the contribution a modeled entity type makes to a data model.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSAggr_classOfDomainsIn_Model

Cache View File Name

cdmwpcm_

Storage View File Name

cdm019

DLL Name

cdmDLL

Storage View Name

DSAggregate

Source

Source Information:

Class Name

DSAggregate

Attribute

dataModel

Target Target Information:

Class Name

DSDataModel

Attribute

aggregates

Forward Mapping

Forward Mapping Semantics:

Verb classOfDomainsIn

Cardinality

1..1

Ordered

No

Controlling

No

Inverse Mapping

Inverse Mapping Semantics:

Verb partitionedInto

Cardinality
0..m

Ordered
No

Controlling
No

Description

An instance of this relationship class identifies a component of a partitioning of a data model in terms of aggregates.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSEnt_atHomeln_Aggr

Cache View File Name
cdmwpcm_

Storage View File Name
cdm020

DLL Name
cdmDLL

Storage View Name
DSEnt_atHomeln_Aggr

Source
Source Information:

Class Name
DSEntityType

Attribute
aggregate

Target Target Information:

Class Name
DSAggregate

Attribute
entityTypes

Forward Mapping
Forward Mapping Semantics:

Verb atHomeln

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping
Inverse Mapping Semantics:

Verb owns

Cardinality
0..m

Ordered
No

Controlling
No

Description
An instance of this relationship class is the set builder construct for an Aggregate instance.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSRel_atHomeln_Aggr

Cache View File Name
cdmwpcm_

Storage View File Name
cdm021

DLL Name
cdmDLL

Storage View Name

DSRel_atHomeIn_Aggr

Source

Source Information:

Class Name

DSRelationshipType

Attribute

aggregate

Target Target Information:**Class Name**

DSAggregate

Attribute

relationshipTypes

Forward Mapping

Forward Mapping Semantics:

Verb atHomeIn**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping

Inverse Mapping Semantics:

Verb owns**Cardinality**

0..m

Ordered

No

Controlling

No

Description

An instance of this relationship class identifies the aggregate to which a relationship belongs.

Attributese

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSAggr_isTOCfor_Aggr

Cache View File Name

cdmwpcm_

Storage View File Name

cdm023

DLL Name

cdmDLL

Storage View Name

DSAggr_isTOCfor_Aggr

Source

Source Information:

Class Name

DSAggregate

Attribute

containedAggregates

Target Target Information:

Class Name

DSAggregate

Attribute

containingAggregate

Forward Mapping

Forward Mapping Semantics:

Verb isTOCfor

Cardinality

0..m

Ordered

Yes

Controlling

No

Inverse Mapping

Inverse Mapping Semantics:

Verb subordinateTo

Cardinality

0..1

Ordered

No

Controlling

No

Description

An instance of this relationship class represents the hierarchical table of contents like tree of aggregates expressing containment.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSRelBundle_atHomeln_Aggr**Cache View File Name**

cdmwpcm_

Storage View File Name

cdm022

DLL Name

cdmDLL

Storage View Name

DSRelBundle_atHomeln_Aggr

Source

Source Information:

Class Name

DSRelationshipBundle

Attribute

aggregate

Target Target Information:**Class Name**

DSAggregate

Attribute

relationshipBundles

Forward Mapping

Forward Mapping Semantics:

Verb atHomeln

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping

Inverse Mapping Semantics:

Verb owns**Cardinality**

0..m

Ordered

No

Controlling

No

Description

An instance of this relationship class identifies the aggregate to which a relationship belongs.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSKeyPairing**Cache View File Name**

cdmwpcm_

Storage View File Name

cdm012

DLL Name

cdmDLL

Storage View Name

DSKeyPairing

Source

Source Information:

Class Name

DSAttribute

Attribute

primaryKey

Target Target Information:

Class Name
DSAttribute

Attribute
foreignKeys

Forward Mapping
Forward Mapping Semantics:

Verb foreignKeyRefersToPrimaryKey

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping
Inverse Mapping Semantics:

Verb primaryKeyReflectsInForeignKey

Cardinality
0..m

Ordered
No

Controlling
No

Description
An instance of this relationship class ties together two uses of attributes as primary and foreign key respectively, implementing relationally a relationship type; or, if keys are composite, contributing to the implementation.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSRel_enactedBy_KeyPairing

Cache View File Name
cdmwpcm_

Storage View File Name

cdm011

DLL Name

cdmDLL

Storage View Name

DSRel_enactedBy_KeyPairing

Source

Source Information:

Class Name

DSRelationshipType

Attribute

keyPairings

Target Target Information:**Class Name**

DSKeyPairing

Attribute

relationshipType

Forward Mapping

Forward Mapping Semantics:

Verb enactedBy**Cardinality**

0..m

Ordered

No

Controlling

Yes

Inverse Mapping

Inverse Mapping Semantics:

Verb enacts**Cardinality**

1..1

Ordered

No

Controlling

No

Description

An instance of this relationship class represents a relational implementation of a relationship type through the pairing of foreign and primary keys; or it contributes to the implementation if keys are composite.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSRel_ifNeedsTable_AssEnt

Cache View File Name

cdmwpcm_

Storage View File Name

cdm013

DLL Name

cdmDLL

Storage View Name

DSRel_ifNeedsTable_AssEnt

Source

Source Information:

Class Name

DSRelationshipType

Attribute

associatedEntityType

Target

Target Information:

Class Name

DSAssociatedEntityType

Attribute

relationshipType

Forward Mapping

Forward Mapping Semantics:

Verb ifNeedsTable

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping

Inverse Mapping Semantics:

Verb for

Cardinality

1..1

Ordered

No

Controlling

No

Description

An instance of this relationship class represents the fact that a relationship type in a relational implementation needs a table for its implementation and identifies which the associated entity type is from which this table will be generated.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSRelBundle_interrelates_Rel**Cache View File Name**

cdmwpcm_

Storage View File Name

cdm014

DLL Name

cdmDLL

Storage View Name

DSRelBundle_interrelates_Rel

Source

Source Information:

Class Name

DSRelationshipBundle

Attribute

relationshipTypes

Target Target Information:**Class Name**

DSRelationshipType

Attribute
relationshipBundles

Forward Mapping
Forward Mapping Semantics:

Verb interrelates

Cardinality
2..m

Ordered
No

Controlling
No

Inverse Mapping
Inverse Mapping Semantics:

Verb interrelatedBy

Cardinality
0..m

Ordered
No

Controlling
No

Description
An instance of this relationship class represents the association of a relationship bundle with the relationship types that are its members.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSFundEnt_isSourceOf_RelBundle

Cache View File Name
cdmwpcm_

Storage View File Name
cdm015

DLL Name
cdmDLL

Storage View Name

DSFundEnt_isSourceOf_RelBundle

Source

Source Information:

Class Name

DSFundamentalEntityType

Attribute

relationshipBundles

Target Target Information:**Class Name**

DSRelationshipBundle

Attribute

fundamentalEntityType

Forward Mapping

Forward Mapping Semantics:

Verb isSourceOf**Cardinality**

0..m

Ordered

No

Controlling

No

Inverse Mapping

Inverse Mapping Semantics:

Verb hasAsSource**Cardinality**

1..1

Ordered

No

Controlling

No

Description

An instance of this relationship class pinpoints the entity type which a bundle of relationship types have as common source.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Relationship Class: DSRel_linksTo_Ent

Cache View File Name

cdmwpcm_

Storage View File Name

cdm016

DLL Name

cdmDLL

Storage View Name

DSRel_linksTo_Ent

Source

Source Information:

Class Name

DSRelationshipType

Attribute

entityTypes

Target Target Information:

Class Name

DSEntityType

Attribute

relationshipTypes

Forward Mapping

Forward Mapping Semantics:

Verb linksTo

Cardinality

2..m

Ordered

No

Controlling

No

Inverse Mapping

Inverse Mapping Semantics:

Verb linkedInto

Cardinality

0..m

Ordered

No

Controlling

No

Description

An instance of this relationship class represents a link (an arm) of a

relationship from a virtual or visible center of a relationship (the "hub" of the relationship) to an entity type that is involved in the relationship.

Attributes

name This attribute names the mapping defined through the Source Class relationship type of this instance of "links to": the mapping in the direction from the cartesian product of all other entity types involved in the relationship type to the one that is instance of the Target Class entity type of "links to". Or the attribute "name" may describe the role the Target Class entity type plays in the Source Class relationship type.

Data Type

string <64>

minimumCardinality

This attribute defines how many images (instances of the Target Class entity type of "links to") each instance of the mapping (as defined above) must at least have. Note that a minimum cardinality of zero means that the mapping is defined only partially over its domain (the cartesian product of all but the Target Class entity type involved in the Source Class relationship type).

Data Type

enum minCardinalityType

Valid Values

Value	Description
-------	-------------

zero	There exist objects of the source entity type which are not mapped to objects of the target entity type
-------------	---

ones	Each object of the source entity type has at least one image it is mapped to in the target entity type.
-------------	---

maximumCardinality

This attribute defines how many images (instances of the Target Class entity type, also referred to as the co-domain of the mapping) each element of the domain (see above) may have at most.

Data Type

enum maxCardinalityType

Valid Values

Value	Description
-------	-------------

one	Each object of the source entity type, if it has mapped, has exactly one image in the target entity type.
------------	---

m	Each object of the source entity type may have undetermined many images it is mapped to in the target entity type.
----------	--

Cache View Methods

None

Storage View Methods

None

Constraints

Create Constraints

Function

Description

cdmSameDataModel (cdmsmmmod)

This constraint ensures that links connect to entity types only that are in the same data model as the relationship type.

Relationship Class: DSAttr_ofType_DataElemn

Cache View File Name

cdmwpcm_

Storage View File Name

cdm017

DLL Name

cdmDLL

Storage View Name

DSAttr_ofType_DataElemn

Source

Source Information:

Class Name

DSAttribute

Attribute

dataElement

Target Target Information:

Class Name

DSDataElement

Attribute

theAttribute

Forward Mapping

Forward Mapping Semantics:

Verb ofType

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping

Inverse Mapping Semantics:

Verb declares

Cardinality

0..m

Ordered

No

Controlling
No

Description

An instance of this relationship class represents the declaration of an attribute with a semantic data type.

Attributes

None

Cache View Methods

None

Storage View Methods

None

Constraints

None

Chapter 6. Shareable Data Definitions Model

Object Class: DSIncludeComponent

Cache View Name

DSIncludeComponent

Storage View Name

DSIncludeComponent PARENTLINE

Description

This is an abstract superclass of all objects that can be components of an include file. This class is needed to maintain the order of the components within the include file. Please see the subclasses for details.

Short name: IncldCmp

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWSDD

Storage View File Name

ewssc01

DLL Name

atlmod

Attributes

Name	Description
------	-------------

pIncludedSource	
------------------------	--

This attribute specifies an instance of the relationship IncldCmp_IsMemberOf_InclSrcDef.

Data Type

relationship to IncludedSourceDef

Valid Values

0 or 1 relationship instance

Cache View Methods

None

Object Class: DSDataDefinition

Cache View Name

DSDataDefinition

Storage View Name

DSDataDefinition PARENTLINE

Description

This is an abstract superclass of objects that provide non-shareable data type definitions. These data type definitions identify the characteristics of a data item. Please see its subclasses for details.

Short name: DataDef

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWSDD

Storage View File Name

ewssc03

DLL Name

atlmod

Attributes

Name	Description
dataItems	This attribute specifies an instance of the relationship DataDef_Defines_DataItem.
Data Type	relationship to DataItem
Valid Values	0 or 1 relationship instance
isConstant	This attribute indicates whether or not this object is a constant. For PL/I, this attribute represents the VALUE keyword. For C/C++, this attribute represents the const keyword.
Data Type	boolean
Valid Values	FALSE, TRUE
Default	FALSE
pArrayDef	This attribute specifies an instance of the relationship DataDef_IsElementOf_ArrayDef.
Data Type	relationship to ArrayDef
Valid Values	0 or 1 relationship instance

pParmTypeFunction

This attribute specifies an instance of the relationship
DataDef_IsParmTypeOf_FnctnDcl

Data Type

relationship to FunctionDeclaration

Valid Values

0 or 1 relationship instance

pPointerDef

This attribute specifies an instance of the relationship
DataDef_isTypeOf_PtrDef.

Data Type

relationship to PointerDef

Valid Values

0 or 1 relationship instance

pReturnTypeFunction

This attribute specifies an instance of the relationship
DataDef_IsReturnTypeOf_FnctnDcl

Data Type

relationship to FunctionDeclaration

Valid Values

0 or 1 relationship instance

Cache View Methods

Name	Description
size	This method returns the size that a data item of type identified by the attribute typeCode occupies in memory. The unit is bytes, except for DE_BIT_STR typeCode whose unit is bits.
Return Type	ulNT4
Parameters	None
DSDDataDefinitionInit	This method performs all necessary initializations (such as setting the default values for attributes) for this object. It is invoked from the constructor.
Return Type	void
Parameters	None

Object Class: DSSimpleTypeDef

Cache View Name

DSSimpleTypeDef

Storage View Name

DSSimpleTypeDef

Inherits From

DSDataDefinition

Description

An instance of this class represents a simple type definition for an elementary data item. This simple type definition can be used in more than one technology, such as relational database, IMS, and high level languages.

Constraints

1. An object of this class can be the target of only one of the following relationships:
 - *DataElmn_DefinedBy_SmplType*
 - *DataItm_DefinedBy_DataDef*

Short name: SmplType**Public Class**

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWSDD

Storage View File Name

ewssc09

DLL Name

atlmod

Attributes

Name Description**asBitData**

This attribute specifies a flag indicating whether or not the character string is treated as bit (binary) data.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

genericPicture

This attribute specifies a string representing a language-independent picture.

Data Type

char<30>

Valid Values

a generic picture string or *null*. Please see the appendix for details on the generic picture string.

Default

null

isComplex

This attribute specifies a flag indicating whether or not a numeric data is real or complex number.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

isSigned

This attribute specifies a flag indicating whether the numeric data type represented by this object is signed or is unsigned.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

TRUE

pForLocalColDef

This attribute specifies an instance of the relationship SmpIType_ForLocalColDef_ClmnDfnt.

Data Type

relationship to ColumnDefinition

Valid Values

0 or 1 relationship instance

precision

This attribute specifies an integer representing the total number of bits or digits of a binary or decimal numeric data.

This number is used to derive the size of a numeric data type.

Data Type

unsigned short

Valid Values

non negative integer

Default

0

pSharedElement

This attribute specifies an instance of the relationship SmpIType_Defines_DataElmn.

Data Type

relationship to DataElement

Valid Values

0 or 1 relationship instance

pSizeVariable

This attribute specifies an instance of the relationship
SmplType_SizeRefersTo_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or 1 relationship instance

pVGDataDef

This attribute specifies an instance of the relationship
SmplType_RefinedBy_VGDataDef.

Data Type

relationship to VGDataDef

Valid Values

0 or 1 relationship instance

scale This attribute specifies the position of the binary point or decimal point.

Data Type

short

Default

0

Update Constraints

(document only)

scale < precision

strLength

This attribute specifies an integer representing the number of bytes
specified in the definition of a character or graphic string data type. For
graphic string, the actual size occupied in storage is twice the strLength.

Data Type

unsigned long

Valid Values

non negative integer

Default

0

strVaryingFlag

This attribute specifies a flag indicating whether or not a string is of variable
length.

Data Type

DSEnum::VarStr

Valid Values

- VARSTR_NON_VARYING
- VARSTR_VARYING
- VARSTR_VARYINGZ

typeCode

This attribute specifies the data type that this object represents.

Data Type

DSEnum::ElementTypeCode

Valid Values

- DE_FIXED_BIN
- DE_FIXED_PACKED_DEC
- DE_FIXED_ZONED_DEC
- DE_FLOAT_BIN
- DE_FLOAT_DEC
- DE_BIT_STR
- DE_CHAR_STR
- DE_GRAPHIC_STR
- DE_MIXED_STR
- DE_DATE_STR
- DE_TIME_STR
- DE_TIMESTAMP
- DE_INDEX
- DE_UNDEFINED

vgEvenPackSQL

This attribute specifies that the element is to be treated as if it were of even length when interfacing with SQL or DB2.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

Update Constraints**(document only)**

This attribute must be FALSE unless the typeCode is DE_FIXED_PACKED_DEC.

vgSignFormat

This attribute specifies whether the original VisualGen data type was NUM instead of NUMC or PACK instead of PACKF. If the typeCode is DE_FIXED_PACKED_DEC and the value of this attribute is FALSE, then the VisualGen data type is PACKF. If the typeCode is DE_FIXED_ZONED_DEC and the value of this attribute is FALSE, then the VisualGen data type is NUMC.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

TRUE

isLOB This attribute specifies a flag indicating whether or not the character string

is treated as a Large Object. The asBitData flag will designate the LOB as a BLOB. Graphic character strings with this attribute TRUE will be DCLOB strings.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

Cache View Methods

Name	Description
------	-------------

asCobolClauses

This method returns a fdCobolClauses object, which contains all COBOL clauses that are equivalent to the typeCode and the related attributes. The sender (caller) is responsible for deleting the returned object.

Return Type

DS CobolClauses *

Parameters

None

asIMSFieldType

This method returns one character representing the IMS field type that is equivalent to the typeCode. The sender (caller) is responsible for deleting the returned object.

Return Type

char

Parameters

None

asPLIDataAttributes

This method returns a DSPLIDataAttributes object, which contains all PL/I data attributes that are equivalent to the typeCode and the related attributes. The sender (caller) is responsible for deleting the returned object.

Return Type

DSPLIDataAttributes

Parameters

None

asRDBDataAttributes

This method returns a fdRDBDataAttributes object, which contains all RDB data attributes that are equivalent to the typeCode and the related attributes. The sender (caller) is responsible for deleting the returned object.

Return Type

DSRDBDataAttributes *

Parameters

None

asVGDataAttributes

This instance method returns the SimpleTypeDef attributes precision, scale, strLength, and vgEvenPackSQL as VisualGen data attributes dependent upon the SimpleTypeDef typeCode.

Return Type

void

Parameters

Name <Direction>Type

decimals

<Output> UCHAR

length <Output> ULONG

bytes <Output> USHORT

evenSQL

<Output> boolean

asVGDataType

This instance method returns the SimpleTypeDef type code as a VisualGen data type.

Return Type

int

Parameters

None

cobolPic2Generic

This method converts a COBOL picture string to a generic picture. The sender (caller) is responsible for deleting the returned object.

Return Type

char *

Parameters

Name <Direction>Type

&sCblPic

<Input > char*

signClause

<Input > dsEnumCobolSignClause

DSSimpleTypeDefInit

This method initializes the default values for all attributes of this object.

This method is invoked from the constructor.

Return Type

void

Parameters

None

fromCobolClauses

This method converts COBOL clauses into the appropriate attributes of this class.

Return Type

void

Parameters

Name <Direction>Type

cbiClauses

<Input > DSCobolClauses

fromIMSFieldAttributes

This method converts IMS field definition (type and length) into the appropriate attributes of this class.

Return Type

void

Parameters

Name <Direction>Type

imsType

<Input > char

length <Input > int

fromPLIDataAttributes

This method converts PL/I data attributes into the appropriate attributes of this class.

Return Type

void

Parameters

Name <Direction>Type

pliType

<Input > DSPLIDataAttributes

fromRDBDataAttributes

This method converts RDB data attributes into the appropriate attributes of this class.

Return Type

void

Parameters

Name <Direction>Type

rdbType

<Input > DSRDBDataAttributes

fromVGDataAttributes

This method sets the SimpleTypeDef values for strLength, precision, scale, and typeCode based on the input VisualGen data type characteristics.

Return Type

void

Parameters

Name <Direction>Type

decimals

<Input > UCHAR

length <Input > ULONG

bytes <Input > USHORT

vgType
<Input > int

genericPic2Cobol

This method converts a generic picture string to a COBOL picture.

The return codes are as follows:

0: successful translation
100: memory allocation error
1001: K not supported in COBOL
1002: P not valid in COBOL with V in string
1003: V not valid in COBOL with P in string
1004: Y not supported in COBOL
1005: + not supported in COBOL
1006: I not supported in COBOL
1007: R not supported in COBOL
1008: T in middle of string not valid in COBOL
1003: F not valid in COBOL

Return Type
char *

Parameters

Name <Direction>**Type**

&sGenPic
<Input > char*

sCbIPic
<Output> string

signClause
<Output> dsEnumCobolSignClause

genericPic2PLI

This method converts a generic picture string to a PL/I picture.

The return codes are as follows:

0: successful translation
100: memory allocation error
1001: O not valid in PL/I
1002: / may not be valid in PL/I
1003: B may not be valid in PL/I
1004: G may not be valid in PL/I
1005: P may not be valid in PL/I

Return Type
char*

Parameters

Name <Direction>**Type**

&sGenPic
<Input > string

sPLiPic
<Output> string

pliPic2Generic

This method converts a PL/I picture string to a generic picture. The sender (caller) is responsible for deleting the returned object.

Return Type

char*

Parameters

Name <Direction>**Type**

&sPliPic

<Input > string

PLIDataAttributesAsString

This method returns a string containing PL/I data attributes represented by this object.

Return Type

char*

Parameters

None

size This method returns the size that a data item of type identified by the attribute typeCode occupies in memory. The unit is bytes, except for DE_BIT_STR typeCode whose unit is bits.

Return Type

unsigned long

Parameters

None

typeCodeAsString

This method returns a string representing the typeCode attribute. The sender (caller) is responsible for deleting the returned object.

Return Type

string

Parameters

None

Object Class: DSStructureDef

Cache View Name

DSStructureDef

Storage View Name

DSStructureDef

Inherits From

DSDataDefinition

Description

An instance of this class represents the structure type definition of an HLL record, an IMS Segment, and/or a VisualGen Record.

For COBOL, this object represents the definition of a data item that contains subordinate data items.

For PL/I, this object represent the definition of a major structure or a minor structure.

For C/C++, this object represents:

- A data type of struct with a tag name.
- A data type of struct without a tag name.

Constraints

1. An object of this class can be the target of only one of the following relationships:
 - ***DataStrc_DefinedBy_StrctDef***
 - ***Dataltm_DefinedBy_DataDef***
2. If an instance of this class is used by an IMS Segment, the following conditions must be true:
 - a. The target of each StrctDef_Has_DataCmp relationship instance is an instance of Dataltem.
 - b. From each of the above mentioned Dataltem, only one of the following relationships is allowed:
 - ***Dataltm_Uses_SDD***
If the target of this relationship is an instance of DataStructure, condition #1 is applied recursively.
 - ***Dataltm_DefinedBy_DataDef***
If the target of this relationship is an instance of StructureDef, condition #1 is applied recursively.
 - ***Dataltm_Redefines_Dataltm***
The data item that is the source of this relationship is not used in generating IMS Field.

Short name: StrctDef

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWSDD

Storage View File Name

ewssc10

DLL Name

atlmod

Attributes

Name Description

dataMembers

This attribute collects instances of the relationship StrctDef_Has_DataCmp.

Data Type

relationship to DataComponent

Valid Values

0 or more relationship instances

pSharedStructure

This attribute specifies an instance of the relationship StrctDef_Defines_DataStrc.

Data Type

relationship to DataStructure

Valid Values

0 or 1 relationship instance

Cache View Methods

Name	Description												
asCobolSource	This method generates, into the file specified in the parameter list, the COBOL source code represented by this data structure. The sender (caller) is responsible for deleting the returned object.												
Return Type	char*												
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th><Direction></th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>&aParmList</td> <td><Input ></td> <td>DSHLLParmList</td> </tr> <tr> <td>parentLevel</td> <td><Input ></td> <td>uINT2</td> </tr> <tr> <td>previousLevelPos</td> <td><Input ></td> <td>uINT2</td> </tr> </tbody> </table>	Name	<Direction>	Type	&aParmList	<Input >	DSHLLParmList	parentLevel	<Input >	uINT2	previousLevelPos	<Input >	uINT2
Name	<Direction>	Type											
&aParmList	<Input >	DSHLLParmList											
parentLevel	<Input >	uINT2											
previousLevelPos	<Input >	uINT2											
asPliSource	This method generates, into the file specified in the parameter list, the PL/I source code represented by this data structure. The sender (caller) is responsible for deleting the returned object.												
Return Type	DSICollection*												
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th><Direction></th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>&aParmList</td> <td><Input ></td> <td>DSHLLParmList</td> </tr> </tbody> </table>	Name	<Direction>	Type	&aParmList	<Input >	DSHLLParmList						
Name	<Direction>	Type											
&aParmList	<Input >	DSHLLParmList											

Object Class: DSSharedDataDefinition

Cache View Name

DSSharedDataDefinition

Storage View Name

DSSharedDataDefinition

Inherits From

DSTechnologyObject

Description

This object class is an abstract superclass which defines a representation of a ***shareable named*** data definition. Please see its subclasses for details.

Short name: SDD

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWSDD

Storage View File Name

ewssc02

DLL Name

atlmod

Attributes

Name Description**dataItems**

This attribute collects instances of the relationship SDD_UsedBy_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or more relationship instances

typeDefReferences

This attribute collects instances of the relationship SDD_UsedBy_TDRef.

Data Type

relationship to TypeDefReference

Valid Values

0 or more relationship instances

pArrayDef

This attribute collects instances of the relationship ArrayDef_HasElement_SharedDataDef

Data Type

relationship to ArrayDef

Valid Values

0 or more relationship instances

Cache View Methods

None

Object Class: DSDataElement

Cache View Name

DSDataElement

Storage View Name
DSDataElement

Inherits From
DSSharedDataDefinition

Description
An object of this class represents a ***shareable named*** data element.
Short name: DataElmn

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWSDD

Storage View File Name
ewssc05

DLL Name
atlmod

Attributes

Name	Description
aliases	This attribute collects instances of the relationship DataElmn_AKA_DEAlias.
Data Type	relationship to DataElementAlias
Valid Values	0 or more relationship instances
pForColumnDef	This attribute specifies an instance of the relationship DataElmn_UsedBy_ColDef.
Data Type	relationship to ColumnDefinition
Valid Values	0 or 1 relationship instance
pSimpleTypeDef	This attribute specifies an instance of the relationship DataElmn_DefinedBy_SmplType.
Data Type	relationship to SimpleTypeDef
Valid Values	0 or 1 relationship instance
theAttribute	This attribute specifies an instance of the relationship Attr_ofType_DataElmn

Data Type

relationship to Attribute

Valid Values

0 or 1 relationship instance

Cache View Methods**Name Description****sample**

This method is a sample method.

Return Type

DSDataElement *

Parameters

None

setFullName

This method sets the fullName attribute.

Return Type

boolean

Parameters**Name <Direction>Type****&fullName**

<Input > char*

setFullName1

This method sets the fullName attribute.

Return Type

boolean

Parameters**Name <Direction>Type****&fullName**

<Input > char*

prefixBaseSep

<Input > char

baseSep

<Input > char

setFullName2

This method sets the fullName attribute.

Return Type

boolean

Parameters**Name <Direction>Type****&prefix**

<Input > char*

&accessName

<Input > char*

&variation
<Input > char*

&revision
<Input > char*

renameAliases
This method renames any aliases associated with this DE

Return Type
void

Parameters
None

Object Class: DSDataStructure

Cache View Name
DSDataStructure

Storage View Name
DSDataStructure

Inherits From
DSSharedDataDefinition

Description
An object of this class represents a *shareable named* data structure.
Short name: DataStrc

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWSDD

Storage View File Name
ewssc06

DLL Name
atlmod

Attributes

Name	Description
aliases	This attribute collects instances of the relationship DataStrc_AKA_DSAlias.
Data Type	relationship to DataStructureAlias
Valid Values	0 or more relationship instances
imsAlternateSegments	This attribute collects instances of the relationship DataStrc_IsAlternateFor_SEGM.

Data Type

relationship to SEGM

Valid Values

0 or more relationship instances

imsMappedSegments

This attribute collects instances of the relationship DataStrc_Maps_SEGM.

Data Type

relationship to SEGM

Valid Values

0 or more relationship instances

pStructureDef

This attribute specifies an instance of the relationship DataStrc_DefinedBy_StrctDef.

Data Type

relationship to StructureDef

Valid Values

0 or 1 relationship instance

Cache View Methods

Name Description**sample**

This method is a sample method.

Return Type

DSDataStructure *

Parameters

None

setFullName

This method sets the fullName attribute.

Return Type

boolean

Parameters**Name** <Direction>Type**&fullName**

<Input > char*

setFullName1

This method sets the fullName attribute.

Return Type

boolean

Parameters**Name** <Direction>Type**&fullName**

<Input > IString

prefixBaseSep

<Input > char

baseSep
 <Input > char

setFullName2
 This method sets the fullName attribute.

Return Type
 boolean

Parameters

Name	<Direction>	Type
&prefix	<Input >	char*
&accessName	<Input >	char*
&variation	<Input >	char*
&revision	<Input >	char*

renameAliases
 This method renames any aliases associated with this DS

Return Type
 void

Parameters
 None

Object Class: DSDataComponent

Cache View Name
 DSDataComponent

Storage View Name
 DSDataComponent

Inherits From
 DSIncludeComponent

Description
 This is an abstract superclass of objects that can be members of a data structure definition. This class is needed to support PLI %INCLUDE and COBOL COPY statement within a data struture. Please see the subclasses for details.

Short name: DataCmp

Public Class
 Yes

Extends
 None

Class Extent
 None

Cache View File Name
 EWSWSDD

Storage View File Name

ewssc07

DLL Name

atlmod

Attributes

Name Description**pStructureDef**

This attribute specifies an instance of the relationship StrctDef_Has_DataCmp.

Data Type

relationship to StructureDef

Valid Values

0 or 1 relationship instance

Cache View Methods

None

Object Class: DSDataltem

Cache View Name

DSDataltem

Storage View Name

DSDataltem

Inherits From

DSDataComponent

Description

An instance of this object class represents a storage location mapped by a high level language (HLL) data declaration.

The **data type** of a data item is defined by the target of one of the following mutually exclusive relationships:

- **Dataltem_DefinedBy_DataDef**
- **Dataltem_IsLike_Dataltem**

This relationship represents the PL/I attribute **LIKE**

- **Dataltem_Uses_SDD**

This relationship represents the usage of a shareable named data definition in the repository.

The **data name** can be found as follows:

1. If the **Dataltem_Uses_SDD** relationship exists, the target object's name is the default data name. See the **pSharedDataDefinition** attribute below.
2. If the **Dataltem_HasNameIn_SDDAlias** relationship exists, the target object's name, which serves as a language-independent name, overrides the above default data name. See the **pSDDAliasAsLocalName** attribute below.
3. For language-specific name:

- If the ***Dataltm_HasCobolNameIn_SDDAlias*** relationship exists, the target object's name, which serves as a COBOL data name, overrides the above language-independent name. See the ***pSDDAliasAsCobolName*** attribute below.
 - If the ***Dataltm_HasPLINameIn_SDDAlias*** relationship exists, the target object's name, which serves as a PL/I data name, overrides the above language-independent name. See the ***pSDDAliasAsPLIName*** attribute below.
4. If the ***Dataltm_Uses_SDD*** relationship does **not** exist, the following attributes provide the data name for this local data item:
- ***defaultName***: language-independent name, which is overridden by the language-specific attributes if they contain values.
 - ***cobolName***: data name in COBOL
 - ***pliName***: data name in PL/I
 - ***vgName***: data name in VisualGen's ESF

Constraint

These attributes should be set to null if the ***Dataltm_Uses_SDD*** relationship exists.

A SharedDataDefinition object, which is the target of the ***Dataltm_Uses_SDD*** relationship, requires a name that is unique among all SharedDataDefinition objects. Tools are encouraged to use the ***Dataltm_Uses_SDD*** relationship instead of the ***Dataltm_DefinedBy_DataDef*** relationship, to increase the level of data sharing in the repository. The ***Dataltm_DefinedBy_DataDef*** relationship is provided to satisfy a requirement from tools (or their users) that need to separate "local and global data". Examples of these top level data items are:

- COBOL level 01 data item that is a structure.
- PL/I level 01 data item that is a structure (or major structure variable)
- C structure variable

COBOL and PL/I level 01 data items that are not structures and COBOL level 77 data items are stored as objects of DataItem class.

Constraints

- If the target of the ***Dataltm_Uses_SDD*** relationship is a ***DataElement***, and if the following relationships exist, their targets must be instances of ***DataElementAlias***
 - ***Dataltm_HasNameIn_SDDAlias*** relationship
 - ***Dataltm_HasCobolNameIn_SDDAlias*** relationship
 - ***Dataltm_HasPLINameIn_SDDAlias*** relationship
 - ***IMSfld_HasFldNameIn_SDDAlias*** relationship
- If the target of the ***Dataltm_Uses_SDD*** relationship is a ***DataStructure***, and if the following relationships exist, their targets must be instances of ***DataStructureAlias***
 - ***Dataltm_HasNameIn_SDDAlias*** relationship
 - ***Dataltm_HasCobolNameIn_SDDAlias*** relationship
 - ***Dataltm_HasPLINameIn_SDDAlias*** relationship
 - ***IMSfld_HasFldNameIn_SDDAlias*** relationship

Short name: Dataltm

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWSDD

Storage View File Name

ewssc08

DLL Name

atlmod

Attributes

Name Description

alignmentFlag

This attribute specifies a flag indicating whether or not data items are to be aligned. This attribute represents COBOL SYNCHRONIZED clause, and PL/I ALIGNED/UNALIGNED key word.

- **ALIGNED_YES** : For PL/I, **ALIGNED** attribute is specified; for COBOL, **SYNCHRONIZED** clause is specified.
- **ALIGNED_NO** : For PL/I, **UNALIGNED** attribute is specified; for COBOL, **SYNCHRONIZED** clause is NOT specified.
- **ALIGNED_UNSPEC** : Not specified

Data Type

DSEnum::Alignment

Valid Values

- **ALIGNED_YES**

- ALIGNED_NO
- ALIGNED_UNSPEC

Default

ALIGNED_UNSPEC

areaOffsets

This attribute collects instances of the relationship
DataItem_AreaContains_Offset.

Data Type

relationship to Offset

Valid Values

0 or more relationship instances

cobolDefinition

This attribute specifies an instance of the relationship
DataItem_RefinedBy_CblDef.

Data Type

relationship to CobolDataDefinition

Valid Values

0 or 1 relationship instance

cobolName

This attribute specifies a data name in COBOL. The value of this attribute, if not null, overrides the **defaultName** attribute.

Data Type

char<30>

Update Constraints

SDD_00060002

This attribute must be null if the relationship
DataItem_Uses_SDD exists.

comment

This attribute specifies the comments on this data item.

Data Type

char*

defaultName

This attribute specifies a language-independent data name. The value of this attribute is overridden by that of the language-specific data name.

Data Type

char<32>

Update Constraints

SDD_00020002

This attribute must be null if the relationship
DataItem_Uses_SDD exists.

endRenamingItems

This attribute collects instances of the relationship
DataItem_IsEndRenamedBy_CblSpNm.

Data Type

relationship CobolSpecialDataName

Valid Values

0 or more relationship instances

fieldsForIMS

This attribute collects instances of the relationship
DataItem_IsUsedAs_IMSfld.

Data Type

relationship to IMSField

Valid Values

0 or more relationship instances

independentDCLFlag

This attribute indicates one of the following:

- INDEPENDENT_NOT : This data item is part of a structure.
- INDEPENDENT_LEVEL01 : This data item is declared as an independent elementary item with level 01.
- INDEPENDENT_LEVEL77 : This data item is declared as an independent elementary item with level 77.
- INDEPENDENT_LEVEL00 : This data item is declared as an independent elementary item without a level number.

Data Type

DSEnum::IndependentDCLFlag

Valid Values

- INDEPENDENT_NOT
- INDEPENDENT_LEVEL01
- INDEPENDENT_LEVEL77
- INDEPENDENT_LEVEL00

Default

INDEPENDENT_NOT

initialValue

This attribute specifies a string representing the initial value of a data item.

Data Type

char*

level88Items

This attribute collects instances of the relationship
DataItem_HasLevel88_CblSpNm.

Data Type

relationship to CobolSpecialDataName

Valid Values

0 or more relationship instances

pArea This attribute collects instances of the relationship
DataItem_DefinesSizeOf_PLIArea.

Data Type

relationship to PLIArea

Valid Values

0 or more relationship instances

pDataDefinition

This attribute specifies an instance of the relationship
DataItem_DefinedBy_DataDef.

Data Type

relationship to DataDefinition

Valid Values

0 or 1 relationship instance

pHasKeyRelSource

This attribute specifies an instance of the relationship
DataItem_IsKeyOf_ArrayDef.

Data Type

relationship to ArrayDef

Valid Values

0 or 1 relationship instance

pIsLikeRelSource

This attribute collects instances of the relationship
DataItem_StructureIsUseIn_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or more relationship instances

pIsLikeRelTarget

This attribute specifies an instance of the relationship
DataItem_IsLike_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or 1 relationship instance

pLBRefersToRelSource

This attribute collects instances of the relationship
DataItem_IsLBOf_ArrayDef

Data Type

relationship to ArrayDef

Valid Values

0 or more relationship instances

pliDefinition

This attribute specifies an instance of the relationship
DataItem_RefinedBy_PLIDef

Data Type

relationship to PLIDataDefinition

Valid Values

0 or 1 relationship instance

pliName

This attribute specifies a data name in PL/I. The value of this attribute, if not null, overrides the **defaultName** attribute.

Data Type

char<32>

Update Constraints**SDD_00060002**

This attribute must be null if the relationship *Dataitm_Uses_SDD* exists.

pRedefinesRelSource

This attribute collects instances of the relationship Dataitm_RedefinedBy_Dataitm.

Data Type

relationship to Dataitem

Valid Values

0 or more relationship instances

pRedefinesRelTarget

This attribute specifies an instance of the relationship Dataitm_Redefines_Dataitm.

Data Type

relationship to Dataitem

Valid Values

0 or 1 relationship instance

pSDDAliasAsCobolName

This attribute specifies an instance of the relationship Dataitm_HasCobolNameIn_SDDAlias.

Data Type

relationship to SDDAlias

Valid Values

0 or 1 relationship instance

pSDDAliasAsLocalName

This attribute specifies an instance of the relationship Dataitm_HasNameIn_SDDAlias.

Data Type

relationship to SDDAlias

Valid Values

0 or 1 relationship instance

pSDDAliasAsPLIName

This attribute specifies an instance of the relationship Dataitm_HasPLINameIn_SDDAias.

Data Type

relationship to SDDAlias

Valid Values

0 or 1 relationship instance

pSDDAliasAsVGName

This attribute specifies an instance of the relationship Dataitm_HasVGNameIn_SDDAlias.

Data Type

relationship to SDDAlias

Valid Values

0 or 1 relationship instance

pSharedDataDef

This attribute specifies an instance of the relationship `DataItem_Uses_SDD`.

Data Type

relationship to `SharedDataDefinition`

Valid Values

0 or 1 relationship instance

pSizeRefersToRelSource

This attribute specifies an instance of the relationship `DataItem_DefinesSizeOf_SimpleType`.

Data Type

relationship to `SimpleTypeDef`

Valid Values

0 or 1 relationship instance

pStorageBasedRelSource

This attribute collects instances of the relationship `DataItem_IsStorageBasedOf_DataItem`.

Data Type

relationship to `DataItem`

Valid Values

0 or more relationship instances

pStorageBasedRelTarget

This attribute specifies an instance of the relationship `DataItem_StorageBasedOn_DataItem`.

Data Type

relationship to `DataItem`

Valid Values

0 or more relationship instances

pUBRefersToRelSource

This attribute collects instances of the relationship `DataItem_IsUBOf_ArrayDef`

Data Type

relationship to `ArrayDef`

Valid Values

0 or more relationship instances

pUnion

This attribute specifies an instance of the relationship `DataItem_IsMemberOf_UnionDef`

Data Type

relationship to `UnionDef`

Valid Values

0 or 1 relationship instance

scope This attribute specifies the scope of a data item.

Data Type

`DSEnum::Scope`

Valid Values

- SCOPE_INTERNAL
- SCOPE_EXTERNAL
- SCOPE_UNSPEC

startRenamingItems

This attribute collects instances of the relationship
`Dataitm_IsStartRenamedBy_CblSpNm`.

Data Type

relationship to `CobolSpecialDataName`

Valid Values

0 or more relationship instances

vgName

This attribute specifies a data name in VisualGen ESF. The value of this attribute, if not null, overrides the **defaultName** attribute.

Data Type

char<30>

Update Constraints**SDD_00060002**

This attribute must be null if the relationship
Dataitm_Uses_SDD exists.

offsetFromLevel01

This attribute specifies the byte offset from the outermost top level data item. The value of this attribute is used to calculate the START position of a field within an IMS Segment. This attribute is for the Atlas OOWA only and will be filled in by the ReCalculate function.

Data Type

long

level66DataItems

This attribute collects instances of the relationship
`Dataitm_HasLevel66_CblSpNm`.

Data Type

relationship to `CobolSpecialDataName`

Valid Values

0 or more relationship instances

isExternal

This attribute specifies a flag indicating whether or not this data item is declared as internal or external.

This attribute represents COBOL **EXTERNAL** clause, PL/I **EXTERNAL** scope attribute, and C **extern** key word.

Data Type

boolean

Valid Values

FALSE, TRUE

Default

FALSE

fieldsSharedByIMS

This attribute points to the DSIMSField instances connected to a Segment that is mapped by the DataStructure containing this DataItem.

This attribute collects instances of the relationship
DataItem_IsUsedAs_IMSfld.

Data Type

relationship to IMSField

Valid Values

0 or more relationship instances

Cache View Methods

Name Description**asCobolSource**

This method generates, into the file specified in the parameter list, the COBOL source code represented by this data item.

Return Type

char*

Parameters

Name <Direction>**Type**

&aParmList

<Input > DSHLLParmList

parentLevel

<Input > uINT2

previousLevelPos

<Input > uINT2

asPliSource

This method generates, into the file specified in the parameter list, the PL/I source code represented by this data item.

Return Type

DSICollection*

Parameters

Name <Direction>**Type**

&aParmList

<Input > DSHLLParmList

dataTypeAsEnum

This method returns a dsEnumDataDef representing the data type of this data item. The sender (caller) is responsible for deleting the returned object.

Return Type

dsEnumDataDef

Parameters

None

dataTypeAsPliAttributes

This method returns a DSPLIDataAttributes representing the data type of this data item. The sender (caller) is responsible for deleting the returned object.

Return Type
DSPLIDataAttributes

Parameters
None

dataTypeAsString

This method returns a string representing the data type of this data item. The sender (caller) is responsible for deleting the returned object.

Return Type
char*

Parameters
None

DSDataltemInit

This method initializes the default values for all attributes of this object. This method is invoked from the constructor.

Return Type
void

Parameters
None

getDataName

This method returns the data name of this data item. The sender (caller) is responsible for deleting the returned object.

Return Type
char*

Parameters

- Name** <Direction>Type
- language**
<Input > dsEnumLanguage
- *pParmList**
<Input > DSParmList

isCobolDataItem

This method returns TRUE if this data item has COBOL-specific data definitions, or returns FALSE otherwise.

Return Type
boolean

Parameters

- Name** <Direction>Type
- &pParmList**
<Input > DSParmList

isIMSField

This method returns TRUE if this data item has IMS-specific data definitions, or returns FALSE otherwise.

Return Type
boolean

Parameters
None

isPLIDataItem

This method returns TRUE if this data item has PLI-specific data definitions, or returns FALSE otherwise.

Return Type

boolean

Parameters

Name <Direction>Type

&pParmList

<Input > DSParmList

mxRelationshipAsEnum

This method returns an enum indicating the existence of one of the mutually exclusive relationships.

Return Type

dsEnumDataItemType

Parameters

None

setDataName

This method sets the data name of this data item.

Return Type

void

Parameters

Name <Direction>Type

&newName

<Input > char*

language

<Input > dsEnumLanguage

setDataName1

This method sets the data name of this data item The sender (caller) is responsible for deleting the returned object.

Return Type

void

Parameters

Name <Direction>Type

&newName

<Input > char*

language

<Input > dsEnumLanguage

***pParmList**

<Input > DSParmList

setIMSAlias

This method sets the alias name of this (IMS) data item The sender (caller) is responsible for deleting the returned object.

Return Type

void

Parameters

Name <Direction>Type

&newName
<Input > char*

***pParmList**
<Input > DSParmList

storageTypeAsString

This method returns the storage type of the data item as string The sender (caller) is responsible for deleting the returned object.

Return Type
char*

Parameters

Name <Direction>Type

***pParmList**
<Input > DSParmList

makeGlobal

This method converts the Dataltem into a global item by connecting it to an existing or new DataElement/DataStructure.

Return Type
char*

Parameters

Name <Direction>Type

****ppRecObjectColl**
<Input > DSCollection

***pParmList**
<Input > DSParmList

***pReconciledObject**
<Input > DSAtlasObject

***targetClass**
<Input > char

***targetName**
<Input > char

getReconcilableObjects

This method finds all the reconcilable objects (DSDataltems) in a DSIncludedSourceDef and puts them into a collection. It also puts DADataltems related to a DSHLLRecord into the collection. Note: This method is recursive.

Return Type
void

Parameters

Name <Direction>Type

****ppRecObjectColl**
<Input > DSCollection

***pParmList**
<Input > DSParmList

asCobolClauses

This method generates COBOL clauses that define this DataItem.

Return Type
DSCobolClauses *

Parameters
None

createUniqueDSName

This method creates a unique DS name, given a DI name

Return Type
char*

Parameters
Name <Direction>Type
&name
<Input > DSUserObjectName

Object Class: DSSDDAlias

Cache View Name
DSSDDAlias

Storage View Name
DSSDDAlias

Inherits From
DSAlias

Description
This is an abstract superclass object class which defines a representation of an alias for a **shareable named** data definition. Please see its subclasses for details.

Short name: SDDAlias

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWSDD

Storage View File Name
ewssc04

DLL Name
atlmod

Attributes

Name	Description
------	-------------

DICobolNames

This attribute collects instances of the relationship
SDDAlias_IsCobolNameFor_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or more relationship instances

DILocalNames

This attribute collects instances of the relationship
SDDAlias_IsNameFor_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or more relationship instances

DIPLINames

This attribute collects instances of the relationship
SDDAlias_IsPLINameFor_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or more relationship instances

DIVGNames

This attribute collects instances of the relationship
SDDAlias_IsVGNameFor_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or more relationship instances

fieldNames

This attribute collects instances of the relationship
SDDAlias_IsFldNameFor_IMSFld.

Data Type

relationship to IMSField

Valid Values

0 or more relationship instances

TDRNames

This attribute collects instances of the relationship
TDRRef_HasCNameIn_SDDAlias.

Data Type

relationship to SDDAlias

Valid Values

0 or more relationship instances

TDRPLINames

This attribute collects instances of the relationship
TDRRef_HasPLINameIn_SDDAlias

Data Type

relationship to SDDAlias

Valid Values

0 or more relationship instances

Cache View Methods

None

Object Class: DSDataElementAlias

Cache View Name

DSDataElementAlias

Storage View Name

DSDataElementAlias

Inherits From

DSSDDAlias

Description

An instance of this class represents an alias name for a ***DataElement*** object.

Short name: DEALias**Public Class**

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWSDD

Storage View File Name

ewssc11

DLL Name

atlmod

Attributes

Name Description**pDataElement**

This attribute specifies an instance of the relationship DEALias_IsFor_DataElmn.

Data Type

relationship to DataElement

Valid Values

0 or 1 relationship instance

pForColumnDef

This attribute collects instances of the relationship DEALias_ForColumnDef_ClmnDfnt.

Data Type
relationship to ColumnDefinition

Valid Values
0 or more relationship instances

VGLocalNames
This attribute collects instances of the relationship
DEAlias_IsVGNameFor_VGDatItm.

Data Type
relationship to VGDataItem

Valid Values
0 or more relationship instances

Cache View Methods

Name **Description**

IsIndependent

This method is used to determine whether or not the subordinate object (the "this" object) is still dependent upon the superior object so the existence of the dependsOn relationship is still necessary. If not, FALSE is returned; otherwise, TRUE is returned

Return Type
int

Parameters

Name **<Direction>Type**
&pSuperiorObject
<Input > DSTechnologyObject*

Object Class: DSDataStructureAlias

Cache View Name
DSDataStructureAlias

Storage View Name
DSDataStructureAlias

Inherits From
DSSDDAlias

Description
An instance of this class represents an alias name for a ***DataStructure*** object.

Short name: DSAlias

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWSDD

Storage View File Name

ewssc12

DLL Name

atlmod

Attributes

Name Description**pDataStructure**

This attribute specifies an instance of the relationship DSAlias_IsFor_DataStrc.

Data Type

relationship to DataStructure

Valid Values

0 or 1 relationship instance

segmNames

This attribute collects instances of the relationship DSAlias_IsIMSNameFor_SEGM.

Data Type

relationship to Segment

Valid Values

0 or more relationship instances

Cache View Methods

Name Description**IsIndependent**

This method is used to determine whether or not the subordinate object (the "this" object) is still dependent upon the superior object so the existence of the dependsOn relationship is still necessary. If not, FALSE is returned; otherwise, TRUE is returned

Return Type

int

Parameters**Name** <Direction>Type**&pSuperiorObject**

<Input > DSTechnologyObject*

Relationship Class: DSLinkDataElmn2SmpIType

Cache View Name

DSLinkDataElmn2SmpIType

Storage View Name

DSLinkDataElmn2SmpIType

Description

An instance of this relationship represents the association between a shareable data element and its simple type definition.

Source Information:

Class Name
DSDataElement

Attribute
pSimpleTypeDef

Target Information:

Class Name
DSSimpleTypeDef

Attribute
pSharedElement

Forward Mapping Semantics:

Verb DefinedBy

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Defines

Cardinality
0..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWSDD

Storage View File Name
ewssl01

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataI2m2SDD

Cache View Name
DSLinkDataI2m2SDD

Storage View Name
DSLinkDataI2m2SDD

Description

An instance of this relationship represents the association between a data item and the shareable data definition.

Source Information:**Class Name**

DSDatalItem

Attribute

pSharedDataDef

Target Information:**Class Name**

DSSharedDataDefinition

Attribute

dataItems

Forward Mapping Semantics:

Verb Uses

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb UsedBy

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWSDD

Storage View File Name

ewssl06

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataI2m2DataDef

Cache View Name

DSLinkDataI2m2DataDef

Storage View Name

DSLinkDataI2m2DataDef

Description

An instance of this relationship represents the association between a data item and its type definition.

Source Information:**Class Name**

DSDataItem

Attribute

pDataDefinition

Target Information:**Class Name**

DSDataDefinition

Attribute

dataItems

Forward Mapping Semantics:

Verb DefinedBy

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Defines

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWSDD

Storage View File Name

ewssl05

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataItemLikeItem

Cache View Name

DSLinkDataItemLikeItem

Storage View Name

DSLinkDataItemLikeItem

Description

An instance of this relationship represents the association between a data item and the structured data item whose subordinate data items are to be used in the former data item.

The target of this relationships represents

- The data item specified after the PL/I LIKE attribute
- The data item specified after the VisualGen ALTSPEC attribute

Source Information:**Class Name**

DSDataItem

Attribute

plsLikeRelTarget

Target Information:**Class Name**

DSDataItem

Attribute

plsLikeRelSource

Forward Mapping Semantics:

Verb IsLike

Cardinality

0..1

Ordered

No

Controlling

Yes

Inverse Mapping Semantics:

Verb StructureIsUseIn

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWSDD

Storage View File Name

ewssl07

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataItrmRedefinesItem**Cache View Name**

DSLinkDataItrmRedefinesItem

Storage View Name

DSLinkDataItrmRedefinesItem

Description

An instance of this relationship represents the COBOL REDEFINES clause, the PL/I DEFINED attribute, or ESF REDEFINES keyword.

Source Information:**Class Name**

DSDataItrm

Attribute

pRedefinesRelTarget

Target Information:**Class Name**

DSDataItrm

Attribute

pRedefinesRelSource

Forward Mapping Semantics:**Verb** Redefines**Cardinality**

0..1

Ordered

No

Controlling

Yes

Inverse Mapping Semantics:**Verb** RedefinedBy**Cardinality**

0..m

Ordered

No

Controlling
No

Cache View File Name
EWSWSDD

Storage View File Name
ewssl08

DLL Name
atlmod

Attributes

Name	Description
------	-------------

position	
-----------------	--

This attribute specifies the expression of specified after the POSITION keyword in a PL/I DEFINED attribute.

This attribute is applicable to PL/I only.

Data Type
char*

Valid Values
a string containing non-negative integer

Default
"0"

Cache View Methods

Name	Description
------	-------------

DSLLinkDataItemRedefinesItemInit	
---	--

This method performs all necessary initializations (such as setting the default values for attributes) for this object. It is invoked from the constructor.

Return Type
void

Parameters
None

Relationship Class: DSLinkDEAlias2DataElmn

Cache View Name
DSLLinkDEAlias2DataElmn

Storage View Name
DSLLinkDEAlias2DataElmn

Description
An instance of this relationship represents the association between a DataElement and its alias.

Source Information:

Class Name
DSDataElementAlias

Attribute
pDataElement

Target Information:

Class Name
DSDataElement

Attribute
aliases

Forward Mapping Semantics:

Verb IsFor

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb AKA

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWSDD

Storage View File Name
ewssl02

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDSAlias2DataStrc

Cache View Name
DSLinkDSAlias2DataStrc

Storage View Name
DSLinkDSAlias2DataStrc

Description

An instance of this relationship represents the association between a DataStructure and its alias.

Source Information:**Class Name**

DSDataStructureAlias

Attribute

pDataStructure

Target Information:**Class Name**

DSDataStructure

Attribute

aliases

Forward Mapping Semantics:

Verb IsFor

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb AKA

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWSDD

Storage View File Name

ewssl04

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataStrc2StrctDef

Cache View Name

DSLinkDataStrc2StrctDef

Storage View Name

DSLinkDataStrc2StrctDef

Description

An instance of this relationship represents the association between a shareable data structure and its structure type definition.

Source Information:**Class Name**

DSDataStructure

Attribute

pStructureDef

Target Information:**Class Name**

DSStructureDef

Attribute

pSharedStructure

Forward Mapping Semantics:

Verb DefinedBy

Cardinality

1..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Defines

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWSDD

Storage View File Name

ewssl03

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataItm2SDDAlias

Cache View Name

DSLinkDataItm2SDDAlias

Storage View Name

DSLinkDataItm2SDDAlias

Description

An instance of this relationship represents the association between a DataItem and its language-independent name.

Source Information:**Class Name**

DSDataItem

Attribute

pSDDAliasAsLocalName

Target Information:**Class Name**

DSSDDAlias

Attribute

DILocalNames

Forward Mapping Semantics:

Verb HasNameIn

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb IsNameFor

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWSDD

Storage View File Name

ewssl10

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataItn2SDDAlias1

Cache View Name

DSLinkDataItn2SDDAlias1

Storage View Name

DSLinkDataItn2SDDAlias1

Description

An instance of this relationship represents the association between a DataItem and its COBOL name.

Source Information:

Class Name

DSDataItem

Attribute

pSDDAliasAsCobolName

Target Information:

Class Name

DSSDDAlias

Attribute

DICobolNames

Forward Mapping Semantics:

Verb HasCobolNameIn

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb IsCobolNameFor

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWSDD

Storage View File Name

ewssl11

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataItn2SDDAlias2**Cache View Name**

DSLLinkDataItn2SDDAlias2

Storage View Name

DSLLinkDataItn2SDDAlias2

Description

An instance of this relationship represents the association between a DataItn and its PL/I name.

Source Information:**Class Name**

DSDataItn

Attribute

pSDDAliasAsPLIName

Target Information:**Class Name**

DSSDDAlias

Attribute

DIPLINames

Forward Mapping Semantics:**Verb** HasPLINameIn**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsPLINameFor**Cardinality**

0..m

Ordered

No

Controlling
No

Cache View File Name
EWSWSDD

Storage View File Name
ewssl2

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataItn2SDDAlias3

Cache View Name
DSLinkDataItn2SDDAlias3

Storage View Name
DSLinkDataItn2SDDAlias3

Description
An instance of this relationship represents the association between a DataItn and its VisualGen name.

Source Information:

Class Name
DSDataItn

Attribute
pSDDAliasAsVGName

Target Information:

Class Name
DSSDDAlias

Attribute
DIVGNames

Forward Mapping Semantics:

Verb HasVGNameIn

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsVGNameFor

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWSDD

Storage View File Name
ewssl14

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkSmpTypeSizeVariable

Cache View Name
DSLinkSmpTypeSizeVariable

Storage View Name
DSLinkSmpTypeSizeVariable

Description
An instance of this relationship represents the association between a PL/I character string size and the data item that defines the actual size of the string.

Source Information:

Class Name
DSSimpleTypeDef

Attribute
pSizeVariable

Target Information:

Class Name
DSDataItem

Attribute
pSizeRefersToRelSource

Forward Mapping Semantics:

Verb SizeRefersTo

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb DefinesSizeOf

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWSDD

Storage View File Name
ewssl15

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkStrctDefComponent

Cache View Name
DSLinkStrctDefComponent

Storage View Name
DSLinkStrctDefComponent

Description

An instance of this relationship represents the association between a structure definition and the data component that is directly contained within the structure. of the following classes:

- IncludeStatement
- DataItem.

If the data item is a substructure, its data components are not the target of this relationship.

Source Information:

Class Name
DSStructureDef

Attribute
dataMembers

Target Information:

Class Name
DSDataComponent

Attribute
pStructureDef

Forward Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
Yes

Controlling
No

Inverse Mapping Semantics:

Verb IsMemberOf

Cardinality
0..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWSDD

Storage View File Name
ewssl09

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataItemBASEDItem

Cache View Name
DSLinkDataItemBASEDItem

Storage View Name
DSLinkDataItemBASEDItem

Description
An instance of this relationship represents the association between a data item and the data item that is specified after the PL/I BASED attribute.

Source Information:

Class Name
DSDataItem

Attribute

pStorageBasedRelTarget

Target Information:**Class Name**

DSDatalItem

Attribute

pStorageBasedRelSource

Forward Mapping Semantics:

Verb StorageBasedOn

Cardinality

0..1

Ordered

No

Controlling

Yes

Inverse Mapping Semantics:

Verb IsStorageBasedOf

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWSDD

Storage View File Name

ewssl13

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Chapter 7. Information Management System (IMS) Model

Object Class: DSDBD

Cache View Name

DSDBD

Storage View Name

DSDBD

Inherits From

DSTechnologyObject

Description

An instance of this object class represents an IMS Data Base description, which is the Root entity for a DBD object. DBD is used as an abbreviation in relationship names that involve this class. The VERSION keyword in the IMS DBD macro is filled from the descriptiveName attribute that is inherited from NamedObject.

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWIMS

Storage View File Name

ewsid1

DLL Name

atImod

Attributes

Name	Description
------	-------------

basedPCBs	This points to a set of instances of PCBbased that are related to this DBD user object by the PCB_BasedOn_DBD relationship.
------------------	---

Data Type	relationship to PCBbased
------------------	--------------------------

Valid Values	0 or more instances
---------------------	---------------------

Update Constraints	(document only)
---------------------------	-----------------

When dliAccess="INDEX", no PCBbased instance should be based on it.

datasets	This points to imsDataset or imsArea relationship instances. Keywords for DATASET and AREA macro statements are stored as attributes on these relationship instances.
-----------------	---

Data Type

relationship to DDNAME

Valid Values

0-10 imsDatasets or 0-240 imsAreas

Update Constraints**(document only)**

When dliAccess="MSDB" or "LOGICAL", there will be no instances of this relationship.

When dliAccess="HIDAM" or "HDAM", there can be up to 10 instances of this relationship.

When dliAccess="DEDB", there can be up to 240 instances of this relationship.

For other values in dliAccess, there can be only 1 instance of this relationship.

dliAccess

This attribute points to the access method and access-specific characteristics of the DBD user object.

Data Type

relationship to DBDaccess

Valid Values

0 or 1 relationship instance

Update Constraints**(document only)**

A DBD instance without an instance of this relationship is incomplete.

exits This attribute points through the DBD_HasDBDexit_PropExit relationship to DataPropagationExit instances. The DataPropagationExit maps to the EXIT keyword and parameters.

Data type

relationship to DataPropagationExit

Visibility

protected

Valid values

0-255 pointer instances

Constraints

This relationship is valid when dliAccess="DEDB", "HDAM", "HISAM", or "HIDAM".

segments

This points to an ordered sequence of Segment instances that are connected by the DBD_ContainsSegment_SEGM relationship. Each Segment instance exists in exactly one DBD instance.

Data Type

relationship to Segment

Valid Values

0-255 Segment instances

Update Constraints

(document only)

If dliAccess="GSAM", there can be 0 Segment instances. If dliAccess="MSDB", "INDEX", "SHSAM", "SHISAM", there can be 1 Segment instance. If dliAccess="DEDB", there can be 127 Segment instances. If dliAccess="HDAM", "HIDAM", "HISAM", "HSAM", "LOGICAL", there can be 255 Segment instances.

versionString

This is a 255-character string that is generated with the VERSION keyword to serve as a descriptive label on the DBD.

Data Type

char*

Valid Values

any printable string or null

Default

null

Cache View Methods

Name	Description
------	-------------

getName

This method returns the 8-character IMS-valid name for the database description.

Return Type

string

Parameters

None

getReconcilableObjects

This method finds all the reconcilable objects (DSDataItems) in a DSDBD (DSDataItems attached to DSFields attached to DSSegments) and puts them into a collection.

Return Type

void

Parameters

Name	<Direction>	Type
------	-------------	------

****ppRecObjectColl**

<Input > DSCollection

***pParmList**

<Input > DSParmList

getRootSegment

This instance method returns the first Segment instance in this DBD. This method is introduced so that code that is concerned only with the root segment can avoid the collection of Segments returned by _get_segments.

Return Type

Segment reference

Parameters

None

isUsed

This instance method returns a TRUE when this DBD user object is used by a PCB or another DBD.

Return Type

boolean

Parameters

None

namedSegment

This instance method returns the segment instance used in the DBD with the specified name.

Return Type

DSSegment *

Parameters

Name <Direction>Type

name <Input > string

rerootSegmentTree

This instance method returns a segment hierarchy tree using the specified segment as the root. This method is used to present segments in the correct order for a PCB using a sequencing DBD, or for a logical DBD using two source segments to cross a logical branch to another physical DBD.

Return Type

Segment collection

Parameters

Name <Direction>Type

&newRoot

<Input > Segment

validateDBD

This instance method validates the DBD, DATASET, AREA, SEGM, FIELD, LCHILD, and XDFLD macro statements with reference to DBDGEN. The validation messages are returned in a formatted string.

Return Type

boolean

Parameters

Name <Direction>Type

messages

<Output> string

writeDBDasSource

This instance method writes the DBD information as a formatted string. The annotateFlag parameter determines whether existing annotation text is included as comments.

Return Type

string

Parameters

Name <Direction>Type

annotateFlag
 <Input > boolean

setFullName
 This method sets the fullName attribute.

Return Type
 boolean

Parameters

Name <Direction>Type

&fullName
 <Input > IString

setFullName1
 This method sets the fullName attribute.

Return Type
 boolean

Parameters

Name <Direction>Type

&fullName
 <Input > IString

prefixBaseSep
 <Input > char

baseSep
 <Input > char

setFullName2
 This method sets the fullName attribute.

Return Type
 boolean

Parameters

Name <Direction>Type

&prefix
 <Input > IString

&accessName
 <Input > IString

&variation
 <Input > IString

&revision
 <Input > IString

renameAliases
 This method renames any aliases associated with this DE

Return Type
 void

Parameters
 None

Object Class: DSDBDaccess

Cache View Name
DSDBDaccess

Storage View Name
DSDBDaccess

Inherits From
DSTechnologyObject

Description
An instance of this class holds the access method of the DBD and access-specific attributes of a DBD user object. Instances of this DBDaccess object class will be used by DBDs with access methods of GSAM, HSAM, HISAM, SHISAM, SHSAM, and LOGICAL. DBDs with other access methods (MSDB, INDEX, HIDAM, DEDB, HDAM) will use instances of the subclasses of this object class.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsida

DLL Name
atlmod

Attributes

Name	Description
------	-------------

containingDBD	This points to DBD instance that uses this instance of DBDaccess.
----------------------	---

Data Type	relationship to DBD
------------------	---------------------

Valid Values	1 relationship instance
---------------------	-------------------------

Update Constraints	(document only) This instance must belong to a DBD
---------------------------	---

dliAccess	This attribute holds the access method of the DBD.
------------------	--

Data Type	char*
------------------	-------

Valid Values	<ul style="list-style-type: none">• "GSAM"• "HSAM"
---------------------	---

- "SHSAM"
- "HISAM"
- "SHISAM"
- "LOGICAL"
- "DEDB"
- "HDAM"
- "HIDAM"
- "INDEX"
- "MSDB"
- null

Default

null

Update Constraints

(document only)

A DBD instance with null in this attribute is the root of an incomplete DBD user object.

osAccess

This attribute indicates whether the operating system access method for the DBD is "VSAM". It affects the string in the ACCESS keyword in the generated DBD when dliAccess="GSAM", "HDAM", or "HIDAM".

When dliAccess="GSAM", if osAccess=FALSE then
"ACCESS=(GSAM,BSAM)" else "ACCESS=(GSAM,VSAM)"

When dliAccess="HDAM" or "HIDAM" if osAccess=FALSE then
"ACCESS=(HDAM,OSAM)" or "ACCESS=(HIDAM,OSAM)" else
"ACCESS=(HDAM,VSAM)" or "ACCESS=(HIDAM,VSAM)"

Data type

boolean

Visibility

protected

Valid values

TRUE, FALSE

Default value

TRUE

passwordFlag

This attribute is a flag to indicate whether PASSWD=YES should be specified on the DBD macro.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

Update Constraints

(document only)

TRUE is valid only when osAccess = TRUE and
dliAccess="GSAM", "HDAM", "HIDAM", "HISAM", "INDEX"
or "SHISAM".

Cache View Methods

Name	Description
------	-------------

validateDBDtype	
------------------------	--

This instance method validates the DBD macro statement with reference to DBDGEN. The validation messages are returned in a formatted string.

Return Type

boolean

Parameters

Name	<Direction>	Type
------	-------------	------

messages		
-----------------	--	--

<Output> string

writeDBD	
-----------------	--

This instance method prepares the DBD, DATASET, AREA, SEGM, FIELD, LCHILD and XDFLD macro statements for DBDGEN.

Return Type

IMSmacroStatement reference

Parameters

None

Object Class: DSDBDdedb

Cache View Name

DSDBDdedb

Storage View Name

DSDBDdedb

Inherits From

DSDBDaccess

Description

An instance of this object class represents a DBD user object that has access=DEDB. A DEDB DBD must have a randomizing module relationship to be valid.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWIMS

Storage View File Name

ewsid2

DLL Name
atlmod

Attributes

Name	Description
------	-------------

rmName	
---------------	--

This attribute is the name of the executable module used to randomize the database.

Data Type
char*

Valid Values
MVS routine name

Update Constraints

(document only)

A DBD with dliAccess="DEDB" or "HDAM" must have a rmName to be valid.

stage	This attribute specifies whether or not the randomizer is a 1 or 2 stage process (default is 1).
--------------	--

Data Type
short

Valid Values
1 or 2

extendedCall	
---------------------	--

This attribute specifies whether or not the randomizer should use the extended call interface.

Data Type
boolean

Valid Values
TRUE, FALSE

Cache View Methods

None

Object Class: DSDBDhdam

Cache View Name
DSDBDhdam

Storage View Name
DSDBDhdam

Inherits From
DSDBDdedb

Description

An instance of this object class represents a DBD user object that has access=HDAM. These attributes apply to the randomizing module relationship that a valid HDAM DBD must have. DBDhdam inherits the randomizing module relationship from DBDdedb.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsid3

DLL Name
atlmod

Attributes

Name Description

relativeBlockNumber

The maximum relative block number that the user wishes to allow a randomizing module to produce for a database. This attributes determines the number of control intervals or blocks in the root-addressable area of a HDAM database. A value of 0 represents a null value.

Data Type

long

Valid Values

0 or 1-16777215

Default

0

rootAnchorPoints

The number of root anchor points desired in each control interval of block in the root addressable area. A value of 0 represents a null value.

Data Type

short

Valid Values

0 or 1-255

Default

1

rootMaxBytes

The maximum number of bytes of database record that can be stored in the root-addressable area in a series of inserts unbroken by a call to another database record. A value of 0 represents a null value.

Data Type

long

Valid Values

0 or 1-16777215

Default

0

Cache View Methods

None

Object Class: DSDBDhidam

Cache View Name
DSDBDhidam

Storage View Name
DSDBDhidam

Inherits From
DSDBDaccess

Description
An instance of this object class represents a DBD user object with an access method of HIDAM. A HIDAM DBD must have a primary index relationship to be valid. The relationship maps to an LCHILD statement under the root segment that has POINTER=INDX and no associated XDFLD statement.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsid4

DLL Name
atlmod

Attributes

Name	Description
------	-------------

primaryIndex	This attribute points to the DBDindex that represents the primary index of the HIDAM DBD user object. The DBDindex instance is connected by the DBD_HasPrimary_DBD relationship.
---------------------	--

Data Type
relationship to DBDindex

Valid Values
0 or 1 relationship instance

Update Constraints

(document only)

A DBD with dliAccess="HIDAM" must have an instance of this relationship to be valid.

rootSequenceField	This attribute points to the IMSField that represents the sequence field in
--------------------------	---

the root segment of the DBD user object. The DBDindex The information is used by the primary index DBD to generate the INDEX= keyword on the LCHILD statement.

Data Type

relationship to IMSField

Valid Values

0 or 1 relationship instance

Update Constraints

(document only)

A DBD with dliAccess="HIDAM" must have an instance of this relationship to be valid.

Cache View Methods

None

Object Class: DSDBDIndex

Cache View Name

DSDBDIndex

Storage View Name

DSDBDIndex

Inherits From

DSDBDaccess

Description

An instance of the DBDindex object class represents a DBD user object that can be used to index a HIDAM database or a segment in a HDAM, HIDAM or HISAM database. The indexing relationship maps to the LCHILD statement in the macro language description of an index DBD.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWIMS

Storage View File Name

ewsid5

DLL Name

atlmod

Attributes

Name Description

dosCompatibility

This attribute indicates whether the index DBD was created with DLI/DOS with a segment code as part of the prefix.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

indexSenseg

This attribute points to the SensitiveSegment relationship instances that are indexed by this DBDIndex instance through the SENSEG_IndexedBy_DBD relationship.

Data Type

relationship to SensitiveSegment

Valid Values

0 or more SensitiveSegment relationships

primaryIndexTarget

This attribute points to the DBDhidam instance that represents a HIDAM database whose root segment is being indexed by this DBD through the DBD_HasPrimary_DBD relationship.

Data Type

relationship to DBDhidam

Valid Values

0 or 1 relationship instance

Update Constraints**(document only)**

This cannot be specified at the same time as secondaryIndexTarget.

protect

This attribute is a flag for data integrity in index pointer segments.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

secondaryIndexTarget

This attribute points to a SecondaryIndex instance that connects this DBDIndex to a Segmentcomplex instance.

Data Type

relationship to SecondaryIndex

Valid Values

0 or 1 relationship instances

Update Constraints**(document only)**

This cannot be specified at the same time as primaryIndexTarget.

sequencedPCB

This attribute points to PCBdatabases that are related to this DBDindex through the PCB_SequencedBy_DBD relationship.

Data Type

relationship to PCBdatabase

Valid Values

0 to many PCBdatabase relationships

sharedDBD

This attribute points to the DBDindex instance that is implemented by the IMSDataset instance. The relationship is DBD_SharesWith_DBD.

Data Type

relationship to DBDindex

Valid Values

0 to 1 DBDindex relationship

Update Constraints**(document only)**

This DBDindex instance must have an instance of this relationship or a relationship to an IMSDataset to be valid.

sharingDBDs

This attribute collects DBDindex instances that share the IMSDataset instance with this DBDindex instance. The relationship from this side is DBD_SharedBy_DBD.

Data Type

relationship to DBDindex

Valid Values

0 to 16 DBDindex relationships

Update Constraints**(document only)**

All of the DBDindex instances must have the same-sized Segment with the sequence field in the same position.

Cache View Methods

Name	Description
getFieldName	This instance method returns either the name of the sequence field in the root segment of a DBDhidam instance that uses this DBDindex as a primary index or else the name in the SecondaryIndex associative entity.
Return Type	string
Parameters	None
getTargetDBD	This instance method returns the DBD instance that is indexed by this DBDindex. The DBD may be the DBDhidam instance that has a HasPrimary relationship to this DBDindex, or it may be the DBD that contains the SegmentComplex instance related through the SecondaryIndex associative entity.

Return Type
DBD reference

Parameters
None

getTargetSegment

This instance method returns the SegmentComplex instance that is indexed by this DBDindex. The SegmentComplex may be the root segment of the DBDhidam instance that has a HasPrimary relationship to this DBDindex, or it may be the SegmentComplex that is related through the SecondaryIndex associative entity.

Return Type
SegmentComplex reference

Parameters
None

setTarget

This instance method set the primaryIndexTarget or the secondaryIndexTarget, depending on the value in the first parameter. When the first parameter is TRUE, the third parameter is optional and the fourth parameter is ignored.

Return Type
void

Parameters

Name <Direction>Type

isPrimary
<Input > boolean

&targetDBD
<Input > DBD

&targetSegment
<Input > SegmentComplex

fieldName
<Input > string

Object Class: DSDBDmsdb

Cache View Name
DSDBDmsdb

Storage View Name
DSDBDmsdb

Inherits From
DSDBDaccess

Description
a DBD with access=MSDB (Mass Storage Data Base).

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsid6

DLL Name
atlmod

Attributes

Name	Description
------	-------------

msdbField

This attribute holds a search field name for a Mass Storage Data Base.

Data Type

char*

Valid Values

string of 1 to 8 alphanumeric characters or null

Default

null

Update Constraints

A string is required in msdbField when msdbType="FIXED", "TERM", or "DYNAMIC" for the DBD user object to be valid.

The string in msdbField must not be the same as the name on any FIELD statement in this DBD.

(document only)

msdbField must be null when msdbType="NO".

msdbType

This attribute specifies the type of Mass Storage Data Base. It may be NO (nonterminal-related without terminal-related keys which has key and sequence field as part of the segment), FIXED (terminal-related fixed), TERM (nonterminal-related with terminal-related keys), or DYNAMIC (terminal-related dynamic).

Data type

char*

Visibility

protected

Valid values

"NO", "TERM", "FIXED", "DYNAMIC", or null

Default value

null

Cache View Methods

None

Object Class: DSsegment

Cache View Name
DSsegment

Storage View Name
DSsegment

Inherits From
DSDBDaccess

Description
An instance of this object class (or of one of its sub-classes) represents a segment within a DBD user object. In relationship names that involve this class or one of its subclasses, SEGM is used as an abbreviation.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsis1

DLL Name
atlmod

Attributes

Name	Description
------	-------------

alternateStructures	This attribute collects the SEGM_HasAlternate_DataStrc relationship instances that connect this Segment instance to shareable data structures representing other descriptions of the data layout.
----------------------------	---

Data Type	relationship to DataStructure
------------------	-------------------------------

Valid Values	0 or more relationship instances
---------------------	----------------------------------

compRoutine	This attribute points to the CompRoutine instance that represents the compression routine for the segment through the SEGM_CompressedBy_Routine relationship.
--------------------	---

Data Type	relationship to CompRoutine
------------------	-----------------------------

Valid Values	0 or 1 relationship instance
---------------------	------------------------------

containingDBD	This attribute points to the DBD instance that is the root of the DBD user object containing this Segment instance. This segment is a target of the DBD_ContainsSegment_SEGM relationship.
----------------------	--

Data Type

relationship to DBD

Valid Values

1 relationship instance

Update Constraints**(document only)**

The ContainsSegment relationship is controlling on the Segment. If the relationship instance is deleted, the Segment instance will also be deleted.

exitFlag

This attribute is a flag to indicate whether a segment will use the data capture exits specified on the DBD. A value of FALSE maps to use of EXIT=NONE parameter on the SEGM macro. This flag has no meaning when exits points to any instances of PropagatedBy.

Data Type

boolean

Valid Values

FALSE, TRUE

Default

TRUE

exits This attribute points to the DataPropagationExit instances that are used by this segment in the DBD user object through the SEGM_UsesSEGMEExit_PropExit relationship.

Data Type

relationship to DataPropagationExit

Valid Values

0 or more relationship instances

fields This attribute points through the IMSField relationship to the Dataltem instances in the shareable data structure that are to be generated as FIELD macro statements under this segment in the DBD user object.

Data Type

relationship to Dataltem

Valid Values

0-255 relationship instances

Update Constraints**(document only)**

The maximum number of IMSFields in the DBD user object is 1000.

frequency

This attribute holds estimated number of times that this segment will occur for each occurrence of its physical parent.

Data Type

char*

Valid Values

0.01-16777215.0, null

Default

null

logicalSegments

This attribute points through the SEGM_IsSourceFor_SEGM relationship to the SegmentLogical instances for which this Segment instance serves as the physicalSegment.

Data Type

relationship to SegmentLogical

Valid Values

0 or more relationship instances

mappingStructure

This attribute points through the SEGM_MappedBy_DataStrc relationship to the shareable data structure user object that describes the physical data mapping for this segment.

Data Type

relationship to DataStructure

Valid Values

0 or 1 relationship instance

maxLength

This attribute holds the length of a fixed-length segment, or the maximum length of a variable length segment.

Data Type

short

Valid Values

1-32760, null

Default

null

minLength

This attribute holds the minimum length of a variable length segment.

Data Type

short

Valid Values

1-32760, null

Default

null

physicalChildren

This attribute points through the SEGM_HasPhysicalChild_SEGM relationship instances to the Segment instance that act as its hierarchical children.

Data Type

relationship to Segment

Valid Values

0-254 relationship instances

Update Constraints

(document only)

A Segment cannot be its own physical parent or physical child.

physicalParent

This attribute points through the SEGM_HasPhysicalParent_SEGM relationship to the Segment instance that acts as its hierarchical parent.

Data Type

relationship to Segment

Valid Values

0 or 1 relationship instance

Update Constraints

(document only)

The first segment in the DBD user object will have no pointer to a physical parent.

A Segment cannot be its own physical parent or physical child.

rules This attributes holds the value that indicates where to place new occurrences of this segment type in the physical database.

Data Type

char*

Valid Values

- "FIRST"
- "LAST"
- "HERE"
- null

Default

null

Update Constraints

(document only)

This attribute applies to user objects in which DBD.dliAccess="DEDB", "HDAM", "HIDAM", or "HISAM".

segmName

This attribute points to the DSALias instance that holds the IMS-valid name for the Data structure used by the segment.

Data type

relationship to DataStructureAlias

Visibility

protected

Valid values

0 or 1 relationship instance

sensitiveSegments

This attribute points through the SensitiveSegment relationship to PCBdatabase instances that are based on this DBD user object and are sensitive to this segment.

Data type

relationship to PCBdatabase

Visibility

protected

Valid values

0 or more relationship instances

Cache View Methods

Name	Description
------	-------------

getName

This method returns the 8-character IMS-valid name for the segment's use of a DataStructure.

Return Type

string

Parameters

None

getNormalFields

This method returns a collection of the non-sequence fields in the segment.

Return Type

DSModelCollection*

Parameters

None

getSequenceFields

This method returns a collection of the sequence fields in the segment.

Return Type

DSModelCollection*

Parameters

None

realLength

This instance method returns the length of the segment including the prefix area.

Return Type

uINT4

Parameters

None

validateSEGM

This instance method validates the SEGM and FIELD statements with reference to DBDGEN. Validation messages are returned as a formatted string.

Return Type

boolean

Parameters

Name <Direction>Type

messages

<Output> string

segmNumber

<Input > uINT2

accessMethod
<Input > string

writeSEGM
This instance method prepares the SEGM and FIELD statements.

Return Type
IMSMacroStatement reference

Parameters
None

Object Class: DSsegmentComplex

Cache View Name
DSsegmentComplex

Storage View Name
DSsegmentComplex

Inherits From
DSsegment

Description
An instance of this class represents a segment in a DBD that can have logical and index relationships. The DBD access method must be HDAM, HISAM, or HIDAM.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsis2

DLL Name
atlmod

Attributes

Name	Description
------	-------------

dataset	This attribute points to the IMSDataset relationship instance that is connected to this SegmentComplex instance by the SEGM_IsPartOf_DATASET relationship. The IMSDataset instance represents the DATASET macro statement under which the SEGM macro statement for this SegmentComplex instance should be placed in the generated DBD user object.
----------------	--

Data Type
relationship to IMSDataset

Valid Values
0 or 1 relationship instance

deleteFlag

This attribute holds the value used for the delete rule.

Data Type

char

Valid Values

- 'L'
- 'P'
- 'V'
- 'B'

Default

'L'

indices

This attribute points through SecondaryIndex instances to the DBDindex instances that serve as secondary indexes.

Data Type

relationship to SecondaryIndex

Valid Values

0 to 32 relationship instances

insertFlag

This attribute holds the value used for the insert rule.

Data Type

char

Valid Values

- 'L'
- 'P'
- 'V'

Default

'L'

logChildren

This attribute points through the SEGM_HasLogicalChild_SEGM relationship to the SegmentComplex instances that act as logical children.

Data Type

relationship to SegmentComplex

Valid Values

0-255 relationship instances

Update Constraints**(document only)**

A SegmentComplex cannot have logChildren and logParent specified at the same time.

logParent

This attribute points through the SEGM_HasLogicalParent_SEGM relationship to the SegmentComplex instance that is used as the logical parent

Data Type

relationship to SegmentComplex

Valid Values

0 or 1 relationship instance

Update Constraints**(document only)**

A SegmentComplex cannot have logChildren and logParent specified at the same time.

pairedLogChild

This attribute points through the SEGM_PairedWith_LCHILD relationship to the SEGM_HasLogicalChild_SEGM relationship instance with which this segment is paired. This relationship maps to the SOURCE keyword on the SEGM macro when this SegmentComplex represents a virtual logical child.

Data Type

relationship to HasLogicalChild

Valid Values

0 or 1 relationship instance

Update Constraints**(document only)**

A SegmentComplex cannot have logChildren and pairedLogChild specified at the same time.

replaceFlag

This attribute holds the value used for the replace rule.

Data Type

char

Valid Values

- 'L'
- 'P'
- 'V'

Default

'L'

segmPointer

This attribute holds the string used for pointer keyword value.

Data Type

char*

Valid Values

- "NOTWIN"
- "TWIN"
- "HIER"
- "TWINBWD"
- "HIERBWD"
- null

Default

null

sourcedIndex

This attribute points through SEGM_IsIndexSourceFor_XDFLD relationship to the SecondaryIndex instances for which this SegmentComplex instance serves as the index source segment.

Data Type
relationship to SecondaryIndex

Valid Values
0 or more relationship instances

Cache View Methods

None

Object Class: DSSegmentDEDB

Cache View Name
DSSegmentDEDB

Storage View Name
DSSegmentDEDB

Inherits From
DSSegment

Description
An instance of this object class acts as a Segment within a DBD user object that has access=DEDB.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsis3

DLL Name
atlmod

Attributes

Name	Description
directDependent	This attribute indicates whether the segment is direct dependent or sequential. A value of TRUE specifies use of "DIR" as the segment type on the generated DBD. A value of FALSE specifies use of "SEQ" on the generated DBD. This attribute is ignored for the root segment of the DBD user object.
Data Type	boolean
Valid Values	TRUE, FALSE
Default	TRUE

Update Constraints

(document only)

The second segment in the DBD user object may use either TRUE or FALSE.

TRUE is the only valid value for segments after the second segment.

subsetPointers

This attribute holds the number of subset pointers in a direct dependent segment.

Data Type

char

Valid Values

0 - 8

Default

0

Update Constraints

(document only)

This attribute is valid only when directDependent is TRUE.

Cache View Methods

Name	Description
------	-------------

validateSEGMdedb	
-------------------------	--

This instance method validates the SEGM and FIELD statements with reference to DBDGEN. Validation messages are returned as a formatted string.

Return Type

boolean

Parameters

Name	<Direction>	Type
------	-------------	------

messages		
-----------------	--	--

<Output> string

segmNumber		
-------------------	--	--

<Input > uINT2

accessMethod		
---------------------	--	--

<Input > string

ds_size		
----------------	--	--

<Input > uINT2

writeSEGM	
------------------	--

This instance method prepares the SEGM and FIELD statements.

Return Type

IMSmacroStatement reference

Parameters

None

Object Class: DSsegmentLogical

Cache View Name

DSsegmentLogical

Storage View Name

DSsegmentLogical

Inherits From

DSsegment

Description

An instance of this object class represents a segment in a DBD user object that has access method of LOGICAL. Segments in a logical DBD have relationships to segments in other DBDs instead of a MappedBy relationship to an dataStructure.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWIMS

Storage View File Name

ewsis4

DLL Name

atlmod

Attributes

Name	Description
------	-------------

segmentName	This attribute holds the 8-character name used for this logical segment instance in the DBD user object. A physical segment will get its name from an alias on the DataStructure.
--------------------	---

Data Type	char*
------------------	-------

Valid Values	first character alphabetic, rest of characters alphanumeric
---------------------	---

Update Constraints	(document only)
---------------------------	-----------------

This segment name must be unique among the segments in the DBD user object.

sourceSegments	This attribute points through SEGM_HasSource_SEGM relationships to Segment instances representing physical segments. The segment and DBD names from the target segment are used in the SOURCE keyword on the SEGM statement. The segment and DBD names from the second instance are used as the second part on the SOURCE keyword on the SEGM statement.
-----------------------	--

Data Type

relationship to Segment

Valid Values

0 to 2 relationship instances

Update Constraints**(document only)**

The target of the second relationship must be the virtual or real logical parent of the target of the first source relationship.

Cache View Methods

None

Object Class: DSIMSField

Cache View Name

DSIMSField

Storage View Name

DSIMSField

Inherits From

DSSegment

Description

An instance of this object type is used to associate an instance of Segment with a Dataltem instance that is contained in the DataStructure instance connected to the Segment and to provide the attributes that apply only to a Dataltem used by a Segment in a specific DBD. IMSfld is used in the names of relationship that connect to this relationship.

The Field name is stored under the Dataltem.

Source

Segment

Semantics

0..m

via Relationship

SEGM_Has_IMSfld

Target Dataltem**Semantics**

0..m

via Relationship

IMSfld_MakesUseOf_Dataltm

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name
EWSWIMS

Storage View File Name
ewsifl

DLL Name
atlmod

Attributes

Name	Description
------	-------------

fldName	
----------------	--

This attribute points to the SDDAlias instance that holds the IMS-valid name for the Data Item used by the field.

Data Type

relationship to SDDAlias

Valid Values

0 or 1 relationship instance

isDuplicateData	
------------------------	--

This attribute points to the XDFLD relationship instances that use this IMSField instance as a duplicate data field through the relationship XDFLD_HasDuplicateData_IMSfld.

Data Type

relationship to SecondaryIndex

Valid Values

0 or more relationship instances

isRootSequence	
-----------------------	--

This attribute points to the DBDhidam that uses this field as the sequence field in the root segment.

Data Type

relationship to DBDhidam

Valid Values

0 or 1 relationship instance

isSearch	
-----------------	--

This attribute points to the XDFLD relationship instances that use this IMSField instance as a search field through the relationship XDFLD_HasSearch_IMSfld.

Data Type

relationship to SecondaryIndex

Valid Values

0 or more relationship instances

isSubsequence	
----------------------	--

This attribute points to the XDFLD relationship instances that use this IMSField instance as a subsequence field through the relationship XDFLD_HasSubsequence_IMSfld.

Data Type

relationship to SecondaryIndex

Valid Values

0 or more relationship instances

senflds

This attribute connects this IMSField instance to the SENSEGs that are sensitive to them through the SensitiveField relationship.

Data Type

relationship to SensitiveField

Valid Values

0 or more relationship instances

sequenceField

This attribute is a flag to indicate whether the dataltem instance should be generated as a sequence field or not.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

sourceSegment

This attribute points to the Segment instance that uses this IMSField instance. This is a mandatory relationship.

Data Type

relationship to Segment

Valid Values

1 relationship instance

targetDataltem

This attribute points to the Dataltem instance used by this IMSField instance. This is a mandatory relationship.

Data Type

relationship to Dataltem

Valid Values

1 relationship instance

uniqueSequence

This attribute is a flag to indicate whether the dataltem instance should be generated as a unique sequence field.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

TRUE

Update Constraints

(document only)

This attribute is meaningless when sequence is not TRUE.

sharedDataltem

This attribute points to the Dataltem instance connected to a Shareable Data Structure that is used by this IMSField instance.

Data Type
relationship to Dataltem

Valid Values
1 relationship instance

Cache View Methods

Name	Description
------	-------------

getName	
----------------	--

This method returns the 8-character IMS-valid name for the field's use of a Dataltem.

Return Type
string

Parameters
None

validateFIELD	
----------------------	--

This instance method validates the FIELD macro statement with reference to DBDGEN. The validation messages are returned in a formatted string.

Return Type
boolean

Parameters

Name	<Direction>	Type
------	-------------	------

messages	<Output>	string
-----------------	----------	--------

accessMethod	<Input >	string
---------------------	----------	--------

writeFIELD	
-------------------	--

This instance method prepares the FIELD macro statement for DBDGEN.

Return Type
IMSmacroStatement reference

Parameters
None

Object Class: DSXDFLD

Cache View Name
DSXDFLD

Storage View Name
DSSecondaryIndex

Inherits From
DSSegment

Description

Instances of this object type are used to associate a SegmentComplex instance to the DBDindex instances that act as secondary indexes for the DBD user object (maps to combination of LCHILD and XDFLD statements). XDFLD is used as an abbreviation in the names of relationships that connect to this class.

Source
SegmentComplex

Semantics
0..1

Relationship
SEGMcmpx_HasIndex_XDFLD

Target DBDIndex

Semantics
0..32

Relationship
XDFLD_IsUseOf_DBDIndex

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsi2i

DLL Name
atlmod

Attributes

Name Description

constant

This attribute holds a character string that defines a one-byte self-defining term with which every index pointer segment in a particular secondary index is identified. It is used to identify index pointer segments for a specific secondary index when multiple secondary indexes reside in the same database.

Data Type

char*

ddata This attribute points to the IMSField instances that this HasSecondary instance uses as duplicate data fields. The relationship is XDFLD_HasDuplicateData_IMSfld.

Data Type

relationship to IMSField

Valid Values

0 to 5 IMSField relationships

exitRoutine

This attribute is the name of the executable module that suppresses creation of index pointer segments.

Data Type

char*

Valid Values

MVS routine name

indexSource

This attribute points to the SegmentComplex instance that this SecondaryIndex instance uses as the source of search, subsequence and duplicate data fields. The relationship is XDFLD_HasIndexSource_SEGM.

Data Type

relationship to SegmentComplex

Valid Values

0 or 1 SegmentComplex relationship

Update Constraints**(document only)**

The SegmentComplex must be the source of the SEGMcmpx_HasIndex_XDFLD relationship, or must be related to the source SegmentComplex through one or more Segment_HasPhysicalParent_Segment relationship instances.

nullValue

This attribute holds a character string that is a one-byte self-defining term. The creation of index pointer segments is suppressed when the specified value is contained in the search field of an index pointer segment.

Data Type

char*

search

This attribute points to the IMSField instances that this HasSecondary instance uses as search fields. The relationship is XDFLD_HasSearch_IMSfld.

Data Type

relationship to IMSField

Valid Values

0 to 5 IMSField relationships

Update Constraints**(document only)**

The generated XDFLD statement is not valid unless there is at least one SEARCH field specified.

sourceSEGMcmpx

This attribute points to the SegmentComplex instance that uses this SecondaryIndex instance. This is a mandatory relationship.

Data type

relationship to SegmentComplex

Valid values

1 relationship instance

subseq

This attribute points to the IMSField instances that this HasSecondary instance uses as subsequence fields. The relationship is XDFLD_HasSubsequence_IMSfld.

Data Type

relationship to IMSField

Valid Values

0 to 5 IMSField relationships

symbolic

This attribute is a flag to determine whether or not the concatenated keys of the index target segments are to be placed in the index pointer segments, instead of a direct pointer. TRUE maps to POINTER=SYMB in the LCHILD statements in both the index DBD and the indexed DBD. FALSE maps to POINTER=INDX in the LCHILD statement in the indexed DBD and to POINTER=SNGL in the LCHILD statement in the index DBD.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

targetDBDIndex

This attribute points to the DBDIndex instance that is used by this SecondaryIndex instance. This is a mandatory relationship.

Data Type

relationship to DBDIndex

Valid Values

1 relationship instance

xdfldName

This attribute holds the name of the indexed data field (maps to NAME keyword on XDFLD statement).

Data Type

char*

Valid Values

all characters alphanumeric

Update Constraints**(document only)**

Name must be unique among all the field names for the index target segment.

Cache View Methods

Name Description**validateINDEX**

This instance method validates the LCHILD and XDFLD macro statements with reference to DBDGEN. The validation messages are returned in a formatted string.

Return Type

boolean

Parameters

Name <Direction>Type

messages
<Output> string

writeLCHILD

This instance method prepares the LCHILD macro statement for the secondary INDEX DBD.

Return Type
IMSmacroStatement reference

Parameters
None

writeXDFLD

This instance method prepares the LCHILD and XDFLD macro statements for the secondary index relationship on a HIDAM, HDAM or HISAM DBD.

Return Type
IMSmacroStatement reference

Parameters
None

Object Class: DSCompRoutine

Cache View Name
DSCompRoutine

Storage View Name
DSCompRoutine

Inherits From
DSSegment

Description
An instance of this object type is used to describe the compression routine parameters for a segment.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsicr

DLL Name
atlmod

Attributes

Name	Description
------	-------------

dataOnly	A value of TRUE specifies that only the data fields in the segment may be compressed (maps to use of DATA keyword in COMPRTN parameter in SEGM macro statement).
-----------------	--

A value of FALSE specifies that both key fields and data fields in the segment can be compressed (maps to use of KEY keyword in COMPRTN parameter on SEGM macro statement).

Data Type

boolean

Valid Values

TRUE, FALSE

Default

TRUE

initialization

A value of TRUE specifies use of INIT keyword on COMPRTN parameter, which indicates that initialization and termination processing control is required by the routine.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

routine

This attribute is the name of the executable module used to compress the segment.

Data Type

char*

Valid Values

MVS routine name

segment

This attribute points to the segment instance that uses this compression routine through the SEGM_CompressedBy_Routine relationship.

Data Type

relationship to Segment

Valid Values

0 or 1 relationship instance

Cache View Methods

None

Object Class: DSExit

Cache View Name

DSExit

Storage View Name

DSDataPropagationExit

Inherits From

DSSegment

Description

An instance of this object type is used to hold the attributes that affect the relationship between a DBD or a Segment and the data capture routine (EXIT keyword on DBD or SEGM statement). It is possible for an instance of DataPropagationExit to be used by a DBD and more than one segment within the DBD.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWIMS

Storage View File Name

ewsidp

DLL Name

atlmod

Attributes

Name Description**capturedDBD**

This attribute points to the DBD instance that uses this data capture exit.

Data Type

relationship to DBD

Valid Values

0 or 1 relationship instance

capturedSegment

This attribute points to the Segment instance that uses this data capture exit.

Data Type

relationship to Segment

Valid Values

0 or 1 relationship instance

captureRoutine

This attribute is the name of the data capture exit.

Data Type

char*

Valid Values

MVS routine name

cascade

This attribute is a flag for CASCADE keyword. CASCADE indicates that the exit will be called for a segment whose parent segment is deleted.

Data Type

boolean

Valid Values
TRUE, FALSE

Default
TRUE

cascadeData

This attribute is a flag to pass the segment data to the exit for a cascade delete.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
TRUE

cascadeKey

This attribute is a flag to pass the segment key to the exit to identify the segment for a cascade delete.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
TRUE

cascadePath

This attribute is a flag to allow an application to separately access several segments for a cascade delete.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

exitData

This attribute is a flag to pass physical segment data to exit for update.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
TRUE

exitKey

This attribute is a flag to pass segment key to exit to identify the segment for update.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
TRUE

exitPath

This attribute is a flag to specify whether data in the hierarchical path needs to be passed to the exit.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

logFlag

This attribute is a flag to specify whether data capture control blocks and data will be written to the IMS system log.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

Cache View Methods

Name	Description
------	-------------

validateEXIT

This instance method validates the EXIT parameter with reference to DBDGEN. Validation messages are returned as a formatted string.

Return Type
boolean

Parameters

Name	<Direction>	Type
messages	<Output>	string

writeEXIT

This instance method prepares the EXIT keyword parameters, returning a string that does NOT include the EXIT= keyword.

Return Type
string

Parameters
None

Object Class: DSDDNAME

Cache View Name
DSDDNAME

Storage View Name
DSDDNAME

Inherits From
DSSegment

Description
An instance of this object class represents the DD name specified for a DATASET or AREA statement in a DBD.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsiddn

DLL Name
atlmod

Attributes

Name	Description
ddsName	This attribute specifies the name for a DD statement.
Data Type	char*
Valid Values	string character set
Default	null
implementedDBDs	This attribute points to the DBD instances that are implemented in the dataset with the given DDname.
Data Type	relationship to DBD
Valid Values	0 or more relationship instances
Update Constraints	(document only) None
overflowedDBDs	This attribute points to the IMSDataset instances that use the dataset with the given DDname as an overflow dataset for the DBD.
Data Type	relationship to IMSDataset
Valid Values	0 or more relationship instances

Update Constraints
(document only)
None

Cache View Methods

None

Object Class: DSArea

Cache View Name
DSArea

Storage View Name
DSIMSArea

Inherits From
DSSegment

Description
Instances of this object type are used to add attributes for the AREA macro statement to the relationship between a DBD instance and the instance of DDNAME that represents the DD1 attribute on the AREA statement.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsiar

DLL Name
atlmod

Attributes

Name	Description
------	-------------

areaSize	The value in this attribute represents the size of the control interval.
-----------------	--

Data Type	short
------------------	-------

Valid Values	<ul style="list-style-type: none">• 512• 1024• 2048• 4096• 8192• 12288
---------------------	---

- 16384
- 20480
- 24576
- 28672

Default

4096

collector

This attribute points to the relationship between DBD and DDNAME to which these attributes apply.

Data Type

relationship to ImplementedBy

Valid Values

1 relationship instance

root

This attribute holds total space allocated to root addressable part of area in terms of UOWs. A value of 0 represents a null value.

Data Type

short

Valid Values

0 or 2-32767

Update Constraints

The value in rootOverflow must be less than the value in uow.

rootOverflow

This attribute holds the amount of space reserved for independent overflow in terms of units of work. A value of 0 represents a null value.

Data type

short

Visibility

protected

Valid values

0 or 1-32767

Default value

Constraints

root and rootOverflow must be specified together.

The value in rootOverflow must be less than the value in root.

uow

This attribute holds the number of control intervals in a unit of work. A value of 0 represents a null value.

Data type

short

Visibility

protected

Valid values

0 or 2-32767

Default value

Constraints

uow and uowOverflow must be specified together.

The value in uowOverflow must be less than the value in uow.

uowOverflow

This attribute holds the number of control intervals in overflow section of a unit of work. A value of 0 represents a null value.

Data type

short

Visibility

protected

Valid values

0 or 1-32767

Default value**Constraints**

uow and uowOverflow must be specified together.

The value in uowOverflow must be less than the value in uow.

(document only)

root and rootOverflow must be specified together.

Cache View Methods

Name Description**validateAREA**

This instance method validates the AREA macro statements with reference to DBDGEN. The validation messages are returned in a formatted string.

Return Type

boolean

Parameters

Name <Direction>Type

messages

<Output> string

ddname

<Input > string

writeAREA

This instance method prepares the AREA macro statement for DBDGEN.

Return Type

IMSmacroStatement reference

Parameters

None

Object Class: DSIMSDataset

Cache View Name

DSIMSDataset

Storage View Name
DSIMSDataset

Inherits From
DSSegment

Description
Instances of this object type are used to add attributes for the DATASET macro statement to the relationship between a DBD instance and the instance of DDNAME that represents the DD1 attribute on the DATASET statement.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsids

DLL Name
atlmod

Attributes

Name	Description
------	-------------

collector	
------------------	--

This attribute points to the relationship between DBD and DDNAME to which these attributes apply.

Data Type
relationship to ImplementedBy

Valid Values
1 relationship instance

datasetLabel	
---------------------	--

This attribute holds a label used for reverse referencing of datasets in a HDAM or HIDAM database.

Data Type
char*

Valid Values
null or valid Assembler label

dd2	This attribute points to the DDNAME used for DATASET DD2= or DATASET OVFLW= and connected through the DATASET_HasOverflow_DDName relationship.
------------	--

Data Type
relationship to DDNAME

Valid Values
0 or 1 relationship instance

freeBlockFrequency

This attribute describes the frequency of free blocks in the initial dataset layout.

Data Type

short

Valid Values

- 0
- 2–100
- null

Default

null

freeSpacePercentage

This attribute describes the percentage of free space in the initial dataset layout.

Data Type

short

Valid Values

- 0–99
- null

Default

null

inputRecordLength

This attribute describes input or primary record length, or maximum record length for GSAM.

Data Type

short

Valid Values

- 1–32767
- null

Default

null

outputRecordLength

This attribute describes output or overflow record length, or minimum record length for GSAM.

Data Type

short

Valid Values

- 1–32767
- null

Default

null

overflowBlockingFactor

This attribute describes the output or overflow blocking factor.

Data Type

short

Valid Values

- 1–32767
- null

Default

null

primaryBlockingFactor

This attribute describes the input or primary blocking factor.

Data Type

short

Valid Values

- 1–32767
- null

Default

null

recordFormat

This attribute describes the record format for GSAM.

Data Type

char*

Valid Values

- "F"
- "V"
- "VB"
- "U"
- null

Default

null

Update Constraints**(document only)**

This attribute is only significant when
DBD.dliAccess="GSAM".

scanCylinders

This attribute describes the number of cylinders to be scanned to find space for new data.

Data type

short

Visibility

protected

Valid values

0-255

Default value

0

searchAlgorithm

This attribute specifies where should IMS look for space in which to put new data.

Data Type
char

Valid Values

- "0"
- "1"
- "2"
- null

Default
null

segments

This attribute points to the SegmentComplex instances in this DBD that are defined as being in this dataset.

Data Type
relationship to SegmentComplex

Valid Values
0-255 SegmentComplex relationships

size1 This attribute describes the Size of input or primary dataset.

Data Type
short

Valid Values
0 or 1-32767

Update Constraints

(document only)

0 is a significant value only when DBD.dliAccess="GSAM".

size2 This attribute describes the size of the output or overflow dataset.

Data type
short

Visibility
protected

Valid values
1-32767, null

Default value
null

Cache View Methods

Name	Description
-------------	--------------------

validateDATASET

This instance method validates the DATASET macro statement with reference to DBDGEN. The validation messages are returned in a formatted string.

Return Type
boolean

Parameters

Name	<Direction>	Type
-------------	--------------------------	-------------

messages
 <Output> string

accessMethod
 <Input > string

ddname
 <Input > string

osAccess
 <Input > boolean

writeDATASET

This instance method prepares the DATASET macro statement for DBDGEN.

Return Type

IMSmacroStatement reference

Parameters

None

Object Class: DSPSB

Cache View Name
 DSPSB

Storage View Name
 DSPSB

Inherits From
 DSTechnologyObject

Description

An instance of this object class represents the root entity of a PSB user object. Within IMS, a PSB (Program Specification Block) is a series of PCB macro instructions that describe an application program's I/O operations and its view and use of segments and fields in IMS databases. The types of PCBs are TP PCB (PCBteleprocessing) which describes interactions with logical terminals, GSAM PCB (PCBbased) which is based on a GSAM DBD used as an input or output dataset, and DB PCB (PCBdatabase) which can relate to segments and fields in its base DBD.

Public Class
 Yes

Extends
 None

Class Extent
 Versioned

Cache View File Name
 EWSWIMS

Storage View File Name
 ewsip0

DLL Name
 atlmod

Attributes

Name	Description
------	-------------

compatibility	
----------------------	--

The value in this attribute provides for compatibility between BMP or MSG and Batch-DL/I parameter lists. When TRUE, the PSB is always treated as if there were an I/O PCB, no matter how it is used. When FALSE, the PSB has an I/O PCB added only when run in a BMP or MSG region.

Data Type	
------------------	--

boolean

Valid Values	
---------------------	--

TRUE, FALSE

Default	
----------------	--

FALSE

hpPSBName	
------------------	--

This attribute specifies the PSB name for VisualGen when the PSB is exported to VisualGen.

Data Type	
------------------	--

char*

Valid Values	
---------------------	--

null or 1 to 8 characters longP

First character is an alphabetic or national (A-Z,\$,#,@) and the rest of characters are alphanumeric or national (A-Z,0-9,\$,#,@). The name cannot contain blanks or have an EZE prefix.

ioaSize	
----------------	--

This attribute holds the size of the largest I/O area to be used by the application program. The size specification is used to determine the amount of main storage reserved in the PSB pool to hold the control region's copy of the user's I/O area data during scheduling of this application program. If this value is not specified, the ACB utility program calculates a maximum I/O area size to be used as a default. The size calculated is the total length of all sensitive segments in the longest possible path call. The value specified is in bytes.

Data Type	
------------------	--

long

Valid Values	
---------------------	--

0-256000

Default	
----------------	--

0

ioErrorOption	
----------------------	--

The value in this attribute represents the condition code returned to the operating system when IMS/VS terminates normally and one or more input or output errors occurred on any data base during the application program execution.

Data Type	
------------------	--

short

Valid Values	
---------------------	--

0-4095

Default

0

language

This attribute holds the language label used on the PSBGEN statement.

Data Type

char*

Valid Values

- "ASSEM"
- "PASCAL"
- "C"
- "COBOL"
- "PL/I"
- null

Default

null

listPCBs

This attribute points to PCBs used by this PSB and connected through the PSB_Includes_PCB relationship.

Data Type

relationship to PCB

Valid Values

0-255 relationship instances

Update Constraints**(document only)**

The name of each PCB object instance must be unique within the PSB user object.

PCBteleprocessing instances (TP PCBs) must be listed first, followed by PCBdatabase instances (DB PCBs), followed by other PCBbased instances (GSAM PCBs).

lockMaximum

The value in this attribute indicates the maximum number of locks an application program can get at one time. The value is specified in units of 1000. For example, a lockMaximum value of 5 indicates a maximum of 5000 locks at one time. A value of 0 turns off the limit.

Data Type

short

Valid Values

0-255

Default

0

maximumQxCalls

The value in this attribute represents the maximum number of data base calls with Qx command codes which may be issued between synchronization points. If this number is exceeded, the application program will abend.

Data Type
unsigned short

Valid Values
0-32767

Default
0

onlinelImageCopy

This attribute specifies whether the user of this PSB is authorized to execute the Online Data Base Image Copy utility or the Surveyor utility feature run as a BMP against a data based named in this PSB. When TRUE, use of the Online Image Copy and the Surveyor utility feature is allowed; When FALSE, use of the Online Image copy and the Surveyor utility feature is prohibited.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
TRUE

Update Constraints

(document only)

TRUE is invalid if any PCBdatabase instance included in t PSB user object has procopt=L or procopt=LS.

onLinelImageCopy is not applicable to a CICS/OS/VS, GSAM, HSAM, MSDB, or DEDB database.

ssaSize

The value in this attribute represents the maximum total length of all SSAs to be used by the application program. The size specification is used to determine the amount of main storage reserved in the PSB pool to hold the control region's copy of the user's SSA string during scheduling of this application program. If not specified, the ACB utility program calculates the maximum SSA size to be used as a default. The size calculated is the maximum number of levels in any PCB within this PSB times 280. The value specified is in bytes.

Data Type
long

Valid Values
0-256000

Default
0

writeToOperator

This attributes holds a subparameter of the IOEROPN parameter. It is tied to the "write-to-operator-with-reply" function in the Utility Control facility. When TRUE, a WTOR for the DFS0451A I/O error message is issued, and DL/I waits for the operator to respond before continuing.

Data Type
boolean

Valid Values

TRUE, FALSE

Default

FALSE

Update Constraints

(document only)

If TRUE, ioErrorOption must be set.

Cache View Methods

Name	Description
------	-------------

getName	
----------------	--

This method returns the 8-character IMS-valid name for the PSB description. This name becomes the load module name for the PSB in the library IMSVS.PSBLIB. It must be the same as the program load module name in the program library called IMSVS.PGMLIB if the program is to run in a message processing region.

Return Type

string

Parameters

None

isUsed	
---------------	--

This instance method returns a TRUE when this PSB user object is used by an Executable instance.

Return Type

boolean

Parameters

None

validatePSB	
--------------------	--

This instance method validates the PSBGEN, PCB, SENSEG and SENFLD statements with reference to PSBGEN. Validation messages are returned as a string.

Return Type

boolean

Parameters

Name	<Direction>	Type
------	-------------	------

messages		
----------	--	--

<Output> string

writePSB	
-----------------	--

This instance method prepares the PCB, SENSEG, SENFLD, and PSBGEN statements.

Return Type

IMSmacroStatement reference

Parameters

None

writePSBasSource

This instance method writes the PSB information as a formatted string. The annotateFlag parameter determines whether existing annotation text is included as comments.

Return Type

string

Parameters

Name <Direction>**Type**

annotateFlag

<Input > boolean

Object Class: DSPCB**Cache View Name**

DSPCB

Storage View Name

DSPCB

Inherits From

DSTechnologyObject

Description

This is a virtual base class for the root of the PCB user object. Three types of PCBs can be defined: TP (in subclass PCBteleprocessing), GSAM (in subclass PCBbased), and DB (in subclass PCBdatabase). A PCB is a series of macro instructions contained in a PSB. Each PCB represents a message destination or data base to be used by the application program. PCB is used in names of relationships involving any of the PCB subclasses.

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWIMS

Storage View File Name

ewsip1

DLL Name

atlmod

Attributes**Name Description****list**

This attribute specifies whether a named PCB is included in the PCB list passed to the application program at entry. TRUE includes the PCB in the PCB list, FALSE excludes it from the PCB list.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

TRUE

Update Constraints**(document only)**

FALSE is only valid if the PCB has a PCBName.

typeOfPCB

This points to an instance of PCBtype that holds the type-specific PCB attributes.

Data Type

relationship to PCBtype

Valid Values

0 or 1 PCBtype

usingPSBs

This attribute points to the PSB user objects that use this PCB user object through the PSB_Includes_PCB relationship.

Data Type

relationship to PSB

Valid Values

0 or more relationship instances

Cache View Methods

Name	Description
------	-------------

getName

This method returns the 8-character IMS-valid name for the PCB description.

Return Type

string

Parameters

None

getPCBtypeString

This method returns the PCB type in normal-IMS terms

Return Type

string

Parameters

None

getProcoptFlag

This method returns TRUE if the PCB procopt (defined in PCBbased and PCBdb type instances) is L or LS and returns FALSE if procopt is not defined or is something other than L/LS.

Return Type

boolean

Parameters

None

isUsed

This instance method returns a TRUE when this PCB user object is used by a PSB.

Return Type

boolean

Parameters

None

validatePCB

This instance method validates the PCB, SENSEG and SENFLD statements with reference to PSBGEN. Validation messages are returned as a formatted string.

Return Type

boolean

Parameters

Name <Direction>Type

messages

<Output> string

writePCBasSource

This instance method writes the PCB information as a formatted string. The annotateFlag parameter determines whether existing annotation text is included as comments.

Return Type

string

Parameters

Name <Direction>Type

annotateFlag

<Input > boolean

Object Class: DSPCBtype**Cache View Name**

DSPCBtype

Storage View Name

DSPCBtype

Inherits From

DSTechnologyObject

Description

This virtual base class covers subtypes that hold the type-specific attributes of PCBs.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name
EWSWIMS

Storage View File Name
ewsipt

DLL Name
atlmod

Attributes

Name	Description
typedPCB	This points to the instance of PCB for which this PCBtype holds the type-specific PCB attributes.
Data Type	relationship to PCB
Valid Values	1 PCB

Cache View Methods

Name	Description						
getName	This method returns the 8-character IMS-valid name for the PCB refined by this type.						
Return Type	string						
Parameters	None						
getPCBtype	This method returns the PCB type in normal-IMS terms						
Return Type	string						
Parameters	None						
validatePCBtype	This instance method validates the PCB, SENSEG and SENFLD statements with reference to PSBGEN. Validation messages are returned as a formatted string.						
Return Type	boolean						
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th><Direction></th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>messages</td> <td><Output></td> <td>string</td> </tr> </tbody> </table>	Name	<Direction>	Type	messages	<Output>	string
Name	<Direction>	Type					
messages	<Output>	string					
writePCB	This instance method prepares the PCB, SENSEG and SENFLD statements.						

Return Type
IMSMacroStatement reference

Parameters
None

Object Class: DSPCBbased

Cache View Name
DSPCBbased

Storage View Name
DSPCBbased

Inherits From
DSPCBtype

Description
An instance of this object class represents an IMS PCB (Program Communication Block) that refers to a data base to be used by an application program. It states the DBD Name of the data base the application program will access and defines the types of operations (get, insert, replace, etc.) the application program can perform on that data base.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsip2

DLL Name
atlmod

Attributes

Name	Description
------	-------------

baseDBD	
----------------	--

	This attribute points to the DBD that represents the data base this PCB uses through the PCB_BasedOn_DBD relationship. This is either the physical or logical DBD to be used as the primary source of data base segments or data set description.
--	---

Data Type	relationship to DBD
------------------	---------------------

Valid Values	0 or 1 relationship instance
---------------------	------------------------------

Update Constraints	
---------------------------	--

(document only)	DBD instance must not have dliAccess=INDEX.
-----------------	---

procoptPCB

This attribute holds a string that represents the processing options on either the sensitive segments or the data set declared in this PCB and which can be used in an associated application program.

Data Type

char*

Default

null

Cache View Methods

Name	Description						
getPCBtype	This method returns the PCB type in normal-IMS terms						
Return Type	string						
Parameters	None						
validatePCBtype	This instance method validates the GSAM PCB statement with reference to PSBGEN. Validation messages are returned as a formatted string.						
Return Type	boolean						
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th><Direction></th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>messages</td> <td><Output></td> <td>string</td> </tr> </tbody> </table>	Name	<Direction>	Type	messages	<Output>	string
Name	<Direction>	Type					
messages	<Output>	string					
writePCB	This instance method prepares the GSAM PCB statement.						
Return Type	IMSmacroStatement reference						
Parameters	None						

Object Class: DSPCBdb

Cache View Name

DSPCBdb

Storage View Name

DSPCBdatabase

Inherits From

DSPCBbased

Description

An instance of this object class represents a data base PCB. A data base PCB describes an application programs access abilities to a physical or logical, non-GSAM, non-Index, hierarchical data base. This type of PCB defines a logical structure which is the hierarchical set of data segments to which the associated application program is sensitive.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsip3

DLL Name
atlmod

Attributes

Name	Description
keyLength	The value specified in bytes of the longest concatenated key for a hierarchic path of sensitive segments used by the application program in the logical data structure. This field is required and should be calculated for the user.
Data Type	unsigned short
Valid Values	1-32767
positioning	This attribute specifies whether single or multiple positioning is desired for the logical data structure. Single or multiple positioning provides a functional variation in the call. Multiple positioning is not supported by HSAM.
Data Type	char
Valid Values	<ul style="list-style-type: none"> • 'S' • 'M' • null
Default	'S' except for DBDs with access=DEDB with two dependent segments, which has 'M' as the default.
Update Constraints	<p>(document only)</p> <p>'M' is not supported for DBDs with access=HSAM.</p>
procseq	The target of the PCB_SequencedBy_DBD relationship must be a secondary index DBD of the base DBD. This DBD defines a secondary processing sequence for the base data base. This attribute is optional, and is valid only if a secondary index exists for the base data base.

- 'S'
- 'M'
- null

Data type

relationship to DBDIndex

Visibility

protected

Valid values

0 or 1 relationship instance

Constraints

If a procseq is specified, the SensitiveSegment hierarchy must reflect the secondary processing sequence in the indexed data base.

If procseq is specified, then procopt (inherited from PCB) may not be L or LS.

senseg

These SensitiveSegments define the logical structure, or the hierarchically related set of data segments to which the associated application program is sensitive. The target of these SensitiveSegment relationships refer to Segment instances defined in the base data base. Their order is affected by the hierarchy of the segments in the base DBD and the secondary processing sequence defined by the procseq indexDBD. Their order should be calculated for the user. All segments in the hierarchic path to any required segment must be specified.

Data Type

relationship to Segment

Valid Values

0 to 100 relationship instances

sequentialBuffering

The value in this attribute specifies if this PCB will be buffered using sequential buffering (SB). COND means the SB should be activated conditionally; NO means that SB should not be used for this DB PCB. This attribute is optional.

Data Type

char*

Valid Values

- "COND"
- "NO"
- null

Default

"NO"

Cache View Methods

Name Description**getPCBtype**

This method returns the PCB type in normal-IMS terms

Return Type

string

Parameters

None

countSensegs

This instance method counts the SENSEGs, SENFLDs and INDICES in the PCB. If the number allowed in a PCB or SENSEG is exceeded, a message is added and the return value is set to FALSE. It also returns the numbers of SENSEGs, SENFLDs and INDICES so that the PSB totals can be checked.

Return Type

boolean

Parameters

Name	<Direction>	Type
------	-------------	------

messages		
----------	--	--

	<Output>	string
--	----------	--------

*sensegCount		
--------------	--	--

	<Input >	uINT4
--	----------	-------

*senfldCount		
--------------	--	--

	<Input >	uINT4
--	----------	-------

*indexCount		
-------------	--	--

	<Input >	uINT4
--	----------	-------

validatePCBtype

This instance method validates the PCB, SENSEG and SENFLD statements with reference to PSBGEN. Validation messages are returned as a formatted string.

Return Type

boolean

Parameters

Name	<Direction>	Type
------	-------------	------

messages		
----------	--	--

	<Output>	string
--	----------	--------

writePCB

This instance method prepares the PCB, SENSEG and SENFLD statements.

Return Type

IMSmacroStatement reference

Parameters

None

Object Class: DSPCBtp**Cache View Name**

DSPCBtp

Storage View Name

DSPCBteleprocessing

Inherits From

DSPCBtype

Description

An instance of this object class represents an Alternate PCB user object (TP PCB). This type of PCB describes a destination other than the source

of the current input message. It allows the application program to send output messages to a destination other than the source of an input message.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWIMS

Storage View File Name

ewsip4

DLL Name

atlmod

Attributes

Name	Description
alternateResponse	This attribute specifies whether this PCB can be used instead of the I/O PCB for responding to terminal in response mode, conversational mode, or exclusive mode.
Data Type	boolean
Valid Values	TRUE, FALSE
Default	FALSE
destination	This attribute points to an object that contains either the logical terminal name or a transaction-code name through the PCB_RefersTo_OutputDest relationship.
Data Type	relationship to OutputMessageDestination
Valid Values	0 or 1 relationship instance
Update Constraints	(document only) This is a required attribute, unless the modify attribute is TRUE.
express	This attribute specifies whether messages from this alternate PCB are to be sent (TRUE) or are to be backed out (FALSE) if the application program should terminate abnormally.
Data type	boolean

Visibility

protected

Valid values

TRUE or FALSE

Default value

FALSE

modify

This attribute specifies whether the alternate PCB is modifiable. This feature allows for the dynamic modification of the destination name associated with this PCB.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

Update Constraints**(document only)**

When this attribute is TRUE, destination must not point to a destination.

sameTerminal

This attribute specifies whether IMS should verify that the logical terminal named in the response alternate PCB is assigned to the same physical terminal as the logical terminal that originated the input message.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

Update Constraints**(document only)**

sameTerminal=TRUE must be specified for alternate response PCBs used by conversational programs and programs operating with terminals in response mode.

sameTerminal=FALSE should be specified if alternate response PCBs are used to send messages to output-only devices that are in exclusive mode.

Cache View Methods

Name	Description
------	-------------

getPCBtype	
-------------------	--

	This method returns the PCB type in normal-IMS terms
--	--

Return Type	
--------------------	--

	string
--	--------

Parameters

None

validatePCBtype

This instance method validates the TP PCB statement with reference to PSBGEN. Validation messages are returned as a formatted string.

Return Type

boolean

Parameters

Name <Direction>Type

messages

<Output> string

writePCB

This instance method prepares the TP PCB statements.

Return Type

IMSmacroStatement reference

Parameters

None

Object Class: DSOutputDest**Cache View Name**

DSOutputDest

Storage View Name

DSOutputMessageDestination

Inherits From

DSPCBtype

Description

An instance of this object type represents either an IMS logical terminal or an IMS transaction as the destination of an alternate TP PCB. MsgDest is used as an abbreviation in relationship names.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWIMS

Storage View File Name

ewsio1

DLL Name

atlmod

Attributes

Name	Description
------	-------------

destinationType

The attribute specifies whether the ltermName attribute signifies a logical terminal or a transaction code. This attribute maps to the LTERM or NAME keyword on the PCB macro statement.

Data Type

char*

Valid Values

- "LTERM"
- "NAME"
- null

Default

null

isReferredToBy

This attribute points to the PCBteleprocessing that refers to this output destination through the PCB_RefersTo_OutputDest relationship.

Data Type

relationship to PCBteleprocessing

Valid Values

0 or more relationship instances

ltermName

This attribute specifies the name of the actual destination of the message and is either a logical terminal name or a transaction-code name. When the name is a transaction-code name, output messages to this PCB are queued for input to the program used to process the transaction-code named by the NAME attribute. The name must be specified in the user's IMS/VS system definition as a logical terminal name or transaction code. This attribute maps to the LTERM/NAME keyword on the PCB macro statement.

Data Type

char*

Cache View Methods

Name	Description
writeKeyword	This instance method prepares the destination keyword for the TP PCB statements.
Return Type	string
Parameters	None

Object Class: DSSenSegment

Cache View Name

DSSenSegment

Storage View Name

DSSensitiveSegment

Inherits From

DSPCBtype

Description

Instances of this objecttype are used to associate a PCBdatabase instance with the Segment instances that represent the segments to which the PCB is sensitive. SENSEG is used as an abbreviation in the names of relationships that connect to this class.

Source

PCBdatabase

Semantics

0..m

via Relationship

PCBdb_HasSensitive_SENSEG

Target Segment**Semantics**

0..m

via Relationship

SENSEG_MapsTo_SEGM

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWIMS

Storage View File Name

ewsis

DLL Name

atlmod

Attributes

Name	Description
childSenseg	This attribute points through the SENSEG_IsParentSensegOf_SENSEG relationship to the SensitiveSegment instances that use this SensitiveSegment as the hierarchical parent.
Data Type	relationship to SensitiveSegment
Valid Values	0 or more relationship instances
indices	This attribute points through the SENSEG_IndexedBy_DBD relationship to some or all of the IndexDBDs used as secondary indexes by the segment. The fields referenced by the HasSearch relationship are used to qualify SSAs for an indexed segment type. This attribute can be specified for indexed segment types only. These indices enable SSAs of calls for the indexed segment type to be qualified on the search field of the index segment type contained in each secondary index specified.

Data Type

relationship to DBDIndex

Valid Values

0 to 32 relationship instances

Update Constraints**(document only)**

There should be no indices if the PCB's PROCOPT = L, LS, I, or D.

parentSenseg

This attribute points through the SENSEG_HasParentSenseg_SENSEG relationship to the SensitiveSegment instance that is the hierarchical parent for this SensitiveSegment. If this attribute is not specified then this SENSEG defines a root segment type as being sensitive.

Data Type

relationship to SensitiveSegment

Valid Values

0 or 1 relationship instance

procoptSENSEG

This attribute holds the processing options allowable for use of this sensitive segment by an associated application program. It has the same meaning as the same attribute in the PCB, plus other options may be specified here which are not allowed on the PCB. This PROCOPT overrides the PCB PROCOPT.

Data Type

char*

senflds

This attribute points through the SensitiveField relationship to IMSField instances in the Segment instance. Each instance maps to a SENFLD macro statement under the SENSEG macro statement.

Data Type

relationship to IMSField

Valid Values

0 or more relationship instances

Update Constraints**(document only)**

A maximum of 255 SENFLDs can be defined in a single SENSEG.

A maximum of 10,000 SENFLD statements can be defined in a single PSB.

The same field may be referenced in more than one SENFLD statement within a SENSEG. If the duplicate field names participate in a concatenated segment and the same field name appears in both portions of the concatenation, the first reference will be to the logical child and all subsequent references will be to the logical parent.

There may be no SENFLDs in a SENSEG with PROCOPT=K.

There may be no SENFLDs within a SENSEG with PROCOPT=I or L, if the SENSEG refers to a logical child segment.

If there are SENFLDs used within a SENSEG with PROCOPT=I or L, a SENFLD must be included for the segment sequence field, if it exists.

There are no SENFLDs when the baseDBD has access MSDB or DEDB.

sourcePCB

This attribute points through the SENSEG_SensedBy_PCBdb relationship to the PCBdatabase instance that uses this SensitiveSegment.

Data Type

relationship to PCBdatabase

Valid Values

1 relationship instance

subsetPointer1

This attribute specifies sensitivity to the first subset pointer which may be R (read sensitive), U (update sensitive), or N (not sensitive).

Data Type

char

Valid Values

- 'R'
- 'N'
- 'U'

Default

'R'

Update Constraints

'U' cannot be used if "GO" has been specified in PCB.procoptPCB.

subsetPointer2

This attribute specifies sensitivity to the second subset pointer which may be R (read sensitive), U (update sensitive), or N (not sensitive).

Data type

char

Visibility

protected

Valid values

- 'R'
- 'N'
- 'U'

Default value

'R'

Constraints

'U' cannot be used if "GO" has been specified in PCB.procoptPCB.

subsetPointer3

This attribute specifies sensitivity to the third subset pointer which may be R (read sensitive), U (update sensitive), or N (not sensitive).

Data type

char

Visibility

protected

Valid values

- 'R'
- 'N'
- 'U'

Default value

'R'

Constraints

'U' cannot be used if "GO" has been specified in PCB.procoptPCB.

subsetPointer4

This attribute specifies sensitivity to the fourth subset pointer which may be R (read sensitive), U (update sensitive), or N (not sensitive).

Data type

char

Visibility

protected

Valid values

- 'R'
- 'N'
- 'U'

Default value

'R'

Constraints

'U' cannot be used if "GO" has been specified in PCB.procoptPCB.

subsetPointer5

This attribute specifies sensitivity to the fifth subset pointer which may be R (read sensitive), U (update sensitive), or N (not sensitive).

Data type

char

Visibility

protected

Valid values

- 'R'
- 'N'
- 'U'

Default value

'R'

Constraints

'U' cannot be used if "GO" has been specified in PCB.procoptPCB.

subsetPointer6

This attribute specifies sensitivity to the sixth subset pointer which may be R (read sensitive), U (update sensitive), or N (not sensitive).

Data type

char

Visibility

protected

Valid values

- 'R'
- 'N'
- 'U'

Default value

'R'

Constraints

'U' cannot be used if "GO" has been specified in PCB.procoptPCB.

subsetPointer7

This attribute specifies sensitivity to the seventh subset pointer which may be R (read sensitive), U (update sensitive), or N (not sensitive).

Data type

char

Visibility

protected

Valid values

- 'R'
- 'N'
- 'U'

Default value

'R'

Constraints

'U' cannot be used if "GO" has been specified in PCB.procoptPCB.

subsetPointer8

This attribute specifies sensitivity to the eighth subset pointer which may be R (read sensitive), U (update sensitive), or N (not sensitive).

Data type

char

Visibility

protected

Valid values

- 'R'
- 'N'
- 'U'

Default value

'R'

Constraints

'U' cannot be used if "GO" has been specified in PCB.procoptPCB.

targetSegment

This attribute points through the SENSEG_MapsTo_SEGM relationship to the Segment instance that defines this SensitiveSegment.

Data type
relationship to Segment

Visibility
protected

Valid values
1 relationship instance

Update constraints
(document only)
'U' cannot be used if "GO" has been specified in
PCB.procoptPCB.

Cache View Methods

Name **Description**

getName

This method returns the 8-character IMS-valid name for the SEGMENT that maps this SENSEG.

Return Type
string

Parameters
None

validateSENSEG

This instance method validates the SENSEG and SENFLD statements with reference to PSBGEN. Validation messages are returned as a formatted string.

Return Type
boolean

Parameters

Name **<Direction>Type**

messages
<Output> string

isRoot
<Input > boolean

writeSENSEG

This instance method prepares the SENSEG and SENFLD statements.

Return Type
IMSmacroStatement reference

Parameters
None

Object Class: DSSenField

Cache View Name
DSSenField

Storage View Name
DSSensitiveField

Inherits From
DSPCBtype

Description

Instances of this object class are used to associate a SensitiveSegment instance to the IMSField instances that represent the fields in the segment to which the PCB must be sensitive.

Field level sensitivity provides an increased level of data independence by isolating application programs from changes in the arrangement of fields within a segment and addition or deletion of data within a segment. Additionally, it enhances data security by limiting an application program to a subset of fields within a segment and controlling replace operations at the field level.

Source
SensitiveSegment

Semantics
0..m

via Relationship
DSSENSEG_Uses_SENFLD

Target IMSField

Semantics
0..m

via Relationship
DSIMSfld_IsUsedIn_SENFLD

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWIMS

Storage View File Name
ewsisf

DLL Name
atlmod

Attributes

Name	Description
------	-------------

replace	
---------	--

	The value of this attribute specifies whether this field may be altered on a replace call.
--	--

Data Type	boolean
------------------	---------

Valid Values	TRUE, FALSE
---------------------	-------------

Default
TRUE

startPosition

The value in this attribute specifies the starting position of this field relative to the beginning of the segment with the user's I/O area. The start position for the first byte of a segment is one.

Data Type
unsigned short

Valid Values
0, 1-32767

Default
0

Update Constraints

(document only)

This is a required attribute to generate a valid PCB.

sourceSENSESEG

This attribute points to the sensitive segment using this sensitive field.

Data type
relationship

Visibility
protected

Valid values
1 relationship instance

targetFIELD

This attribute points to the field to which the sensitive segment is sensitive.

Data Type
relationship

Valid Values
1 relationship instance

Cache View Methods

Name Description

validateSENFLD

This instance method validates the SENFLD statement with reference to PSBGEN. Validation messages are returned as a formatted string.

Return Type
boolean

Parameters

Name <Direction>Type
messages
<Output> string

writeSENFLD

This instance method prepares the SENFLD statement.

Return Type
IMSmacroStatement reference

Parameters
None

Relationship Class: DSBasedOn

Cache View Name
DSBasedOn

Storage View Name
DSBasedOn

Description
An instance of this relationship type is used to connect a PCBbased instance to the DBD instance that represents the DBD user object on which the PCB object is based.

Source Information:

Class Name
DSPCBbased

Attribute
baseDBD

Target Information:

Class Name
DSDBD

Attribute
basedPCBs

Forward Mapping Semantics:

Verb BasedOn

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsBasisOf

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsibo

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSCompressedBy

Cache View Name

DSCompressedBy

Storage View Name

DSCompressedBy

Description

An instance of this relationship type is used to connect a Segment instance to the CompRoutine instance that compresses segment data.

Source Information:

Class Name

DSSegment

Attribute

compRoutine

Target Information:

Class Name

DSCompRoutine

Attribute

segment

Forward Mapping Semantics:

Verb CompressedBy

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Compresses

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWIMS

Storage View File Name

ewsicb

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSContainsSegment**Cache View Name**

DSContainsSegment

Storage View Name

DSContainsSegment

Description

Instances of this relationship type are used to connect a DBD instance to the Segment instances that are used in the DBD user object.

Source Information:**Class Name**

DSDBD

Attribute

segments

Target Information:**Class Name**

DSSegment

Attribute

containingDBD

Forward Mapping Semantics:**Verb** Contains**Cardinality**

0..m

Ordered

Yes

Controlling

No

Inverse Mapping Semantics:**Verb** ContainedIn**Cardinality**

1..1

Ordered

No

Controlling
Yes

Cache View File Name
EWSWIMS

Storage View File Name
ewsics

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSDefinedByAccess

Cache View Name
DSDefinedByAccess

Storage View Name
DSDefinedByAccess

Description

An instance of this relationship type is used to connect an instance of DBD to an instance of DBDaccess or one of its subclasses in order to hold the access-specific attributes of a DBD. Using this class instead of subclassing DBD directly allows a tool to change the access method of a DBD (or decide the access method later) without having to change the class of the DBD.

Source Information:

Class Name
DSDBD

Attribute
dliAccess

Target Information:

Class Name
DSDBDaccess

Attribute
containingDBD

Forward Mapping Semantics:

Verb DefinedBy

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Defines

Cardinality
1..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWIMS

Storage View File Name
ewsidb

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasAlternate

Cache View Name
DSHasAlternate

Storage View Name
DSHasAlternate

Description

Instances of this relationship type are used to connect a Segment instance to data structures that provide alternate mappings for the segment data in the DBD user object.

Source Information:

Class Name
DSSegment

Attribute
alternateStructures

Target Information:

Class Name
DSDataStructure

Attribute
imsAlternateSegments

Forward Mapping Semantics:

Verb HasAlternate

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsAlternateFor

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsiha

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasDuplicateData

Cache View Name
DSHasDuplicateData

Storage View Name
DSHasDuplicateData

Description

Instances of this relationship type are used to connect a SecondaryIndex instance to the IMSField instances that are used by the secondary index as duplicate data fields (DDATA keyword on XDFLD statement).

Source Information:

Class Name
DSXDFLD

Attribute
ddata

Target Information:

Class Name

DSIMSField

Attribute

isDuplicateData

Forward Mapping Semantics:**Verb** HasDuplicateData**Cardinality**

0..5

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsDuplicateDataFor**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsidd

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasField

Cache View Name

DSHasField

Storage View Name

DSHasField

Description

An instance of this relationship type is used to connect an instance of Segment to an instance of IMSField.

Source Information:

Class Name
DSSegment

Attribute
fields

Target Information:

Class Name
DSIMSField

Attribute
sourceSegment

Forward Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsUsedBy

Cardinality
1..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWIMS

Storage View File Name
ewsihf

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasIndex

Cache View Name
DSHasIndex

Storage View Name
DSHasIndex

Description

Instances of this relationship type are used to connect a SegmentComplex instance to the SecondaryIndex instances that act as secondary indexes for the DBD user object (maps to combination of LCHILD and XDFLD statements).

Source Information:**Class Name**

DSSegmentComplex

Attribute

indices

Target Information:**Class Name**

DSXDFLD

Attribute

sourceSEGMcmpx

Forward Mapping Semantics:

Verb HasIndex

Cardinality

0..32

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Indexes

Cardinality

1..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWIMS

Storage View File Name

ewsihi

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasIndexSource

Cache View Name

DSHasIndexSource

Storage View Name

DSHasIndexSource

Description

An instance of this relationship type is used to connect a SecondaryIndex instance to the SegmentComplex instance that represents the index source segment for the secondary index (SEGMENT keyword on XDFLD statement).

Source Information:**Class Name**

DSXDFLD

Attribute

indexSource

Target Information:**Class Name**

DSSegmentComplex

Attribute

sourcedIndex

Forward Mapping Semantics:

Verb HasIndexSource

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb IsIndexSourceFor

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsiis

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLChild

Cache View Name

DSLChild

Storage View Name

DShasLogicalChild

Description

Instances of this relationship type are used to connect a SegmentComplex instance to the SegmentComplex instances for which it is the logical parent (maps to LCHILD statement and to the PARENT keyword on the SEGM statement). LCHILD is used as an abbreviation in names of relationships that connect to this relationship.

Source Information:

Class Name

DSSegmentComplex

Attribute

logChildren

Target Information:

Class Name

DSSegmentComplex

Attribute

logParent

Forward Mapping Semantics:

Verb HasLogicalChild

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb HasLogicalParent

Cardinality

0..1

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsilc

DLL Name

atlmod

Attributes

Name Description**counter**

This attribute holds a flag for whether COUNTER keyword is to be used in the POINTER= parameter on the child segment.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

lcPointer

This attribute holds a value used in the POINTER keyword on the LCHILD macro to specify amount of pointer fields to be reserved in the logical parent segment.

Data Type

char*

Valid Values

- "SNGL"
- "DBLE"
- "NONE"
- null

Default

null

lparent

This attribute holds a flag for whether LPARNT keyword is to be used in the POINTER= parameter on the child segment.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

ltwin

This attribute holds a value to be used in the POINTER= parameter on the child segment in order to specify logical twin pointers.

Data Type

char*

Valid Values

- "LTWIN"
- "LTWINBWD"
- null

Default

null

pairedSegment

This attribute points to the SegmentComplex named in the PAIR= keyword on the LCHILD macro. The relationship is LCHILD_HasPair_SEGM.

Data Type

relationship to SegmentComplex

Valid Values

0 or 1 SegmentComplex relationship

rules This attribute holds a value used in the RULES keyword on the LCHILD macro to control the logical twin sequence.

Data Type

char*

Valid Values

- "FIRST"
- "LAST"
- "HERE"
- null

Default

null

virtualParent

This attribute holds a value used in the PARENT parameter on the logical parent segment to specify whether the concatenated key of the logical parent segment is stored with each logical child segment.

Data Type

char*

Valid Values

- "VIRTUAL"
- "PHYSICAL"
- null

Default

null

Cache View Methods

Name	Description
------	-------------

validateStatement

This instance method validates the LCHILD macro statements with reference to DBDGEN. The validation messages are returned in a formatted string.

Return Type

boolean

Parameters

writeLCHILD writeLPARENT	<p>Name <Direction>Type</p> <p>messages <Output> string</p> <p>This instance method prepares the LCHILD macro statements for DBDGEN.</p> <p>Return Type IMSmacroStatement reference</p> <p>Parameters None</p> <p>This instance method prepares the logical PARENT keyword parameters, returning a string that does NOT include the PARENT= keyword.</p> <p>Return Type string</p> <p>Parameters None</p>
---	---

Relationship Class: DSHasOverflow

Cache View Name
DSHasOverflow

Storage View Name
DSHasOverflow

Description
An instance of this relationship type is used to connect a IMSDataset instance to a DDNAME used as an output or overflow dataset (DD2 or OVFLW attribute on DATASET statement).

Source Information:

Class Name Attribute	DSIMSDataset dd2
---	-------------------------

Target Information:

Class Name Attribute	DSDDNAME overflowedDBDs
---	--------------------------------

Forward Mapping Semantics:

Verb Cardinality Ordered Controlling	HasOverflow 0..1 No No
---	---

Inverse Mapping Semantics:**Verb** IsOverflowFor**Cardinality**
0..m**Ordered**
No**Controlling**
Yes**Cache View File Name**
EWSWIMS**Storage View File Name**
ewsiho**DLL Name**
atlmod**Attributes**

None

Cache View Methods

None

Relationship Class: DSHasPair**Cache View Name**
DSHasPair**Storage View Name**
DSHasPair**Description**

An instance of this relationship type is used to connect a HasLogicalChild instance to the SegmentComplex that represents the paired segment (PAIR keyword on LCHILD statement).

Source Information:**Class Name**
DSLChild**Attribute**
pairedSegment**Target Information:****Class Name**
DSSegmentComplex**Attribute**
pairedLogChild**Forward Mapping Semantics:****Verb** HasPair

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** PairedWith**Cardinality**

0..1

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsihp

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasParentSenseg**Cache View Name**

DSHasParentSenseg

Storage View Name

DSHasParentSenseg

Description

An instance of this relationship type is used to connect a SensitiveSegment instance to the SensitiveSegment instance that represents the hierarchical parent in the PCB user object.

Source Information:**Class Name**

DSSenSegment

Attribute

parentSenseg

Target Information:**Class Name**

DSSenSegment

Attribute

childSenseg

Forward Mapping Semantics:**Verb** HasParentSenseg**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsParentSensegOf**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsips

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSParent

Cache View Name

DSParent

Storage View Name

DSHasPhysicalParent

Description

An instance of this relationship type is used to connect an Segment instance to the hierarchical parent Segment instance in the same DBD user object (maps to PARENT keyword on SEGM macro statement).

Source Information:**Class Name**

DSSegment

Attribute
physicalParent

Target Information:

Class Name
DSSegment

Attribute
physicalChildren

Forward Mapping Semantics:

Verb HasPhysicalParent

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasPhysicalChild

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsipp

DLL Name
atlmod

Attributes

Name	Description
------	-------------

pcPointer	
------------------	--

	This attribute describes the type of physical child pointer to be stored in the prefix area of the segment in the DBD.
--	--

Data Type	char*
------------------	-------

Valid Values	
---------------------	--

- | | |
|--|--|
| | <ul style="list-style-type: none">• "SNGL"• "DBLE"• null |
|--|--|

Default	null
----------------	------

Cache View Methods

Name	Description
------	-------------

writePARENT	
--------------------	--

	This instance method prepares the physical PARENT keyword parameters, returning a string that does NOT include the PARENT= keyword.
--	---

Return Type	
	string

Parameters	
	None

Relationship Class: DSIndex

Cache View Name	
	DSIndex

Storage View Name	
	DSHasPrimary

Description	
--------------------	--

	An instance of this relationship type is used to connect an instance of DBDhidam to an instance of DBDindex that serves as the primary index for the HIDAM database.
--	--

Source Information:	
----------------------------	--

Class Name	
	DSDBDhidam

Attribute	
	primaryIndex

Target Information:	
----------------------------	--

Class Name	
	DSDBDindex

Attribute	
	primaryIndexTarget

Forward Mapping Semantics:	
-----------------------------------	--

Verb	HasPrimary
-------------	------------

Cardinality	
	0..1

Ordered	
	No

Controlling	
	No

Inverse Mapping Semantics:	
-----------------------------------	--

Verb	IsPrimaryFor
-------------	--------------

Cardinality	
	0..1

Ordered	
	No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsi1i

DLL Name
atlmod

Attributes

None

Cache View Methods

Name	Description
------	-------------

writeINDEX	
-------------------	--

This instance method prepares the LCHILD macro statement for the primary index of a HIDAM DBD.

Return Type

IMSmacroStatement reference

Parameters

None

writeLCHILD	
--------------------	--

This instance method prepares the LCHILD macro statement for a primary INDEX DBD.

Return Type

IMSmacroStatement reference

Parameters

None

Relationship Class: DSHasRootSequence

Cache View Name
DSHasRootSequence

Storage View Name
DSHasRootSequence

Description

Instances of this relationship type are used to connect a DBD instance to the IMSField instance that is the sequence field of the root segment. The information is used by the primary index DBD for the INDEX=keyword on the LCHILD statement.

Source Information:

Class Name
DSDBDhidam

Attribute
rootSequenceField

Target Information:

Class Name

DSIMSField

Attribute

isRootSequence

Forward Mapping Semantics:**Verb** HasRootSequence**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsRootSequenceFor**Cardinality**

0..1

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsirs

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasSearch**Cache View Name**

DSHasSearch

Storage View Name

DSHasSearch

Description

Instances of this relationship type are used to connect a SecondaryIndex instance to the IMSField instances that are used by the secondary index as search fields (SEARCH keyword on XDFLD statement).

Source Information:

Class Name
DSXDFLD

Attribute
search

Target Information:

Class Name
DSIMSField

Attribute
isSearch

Forward Mapping Semantics:

Verb HasSearch

Cardinality
0..5

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsSearchFor

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsihs

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasSensitive

Cache View Name
DSHasSensitive

Storage View Name
DSHasSensitive

Description

Instances of this relationship type are used to connect a PCBdatabase instance to the SensitiveSegment instances that reference the segments to which the PCB is sensitive.

Source Information:**Class Name**

DSPCBdb

Attribute

senseg

Target Information:**Class Name**

DSSenSegment

Attribute

sourcePCB

Forward Mapping Semantics:

Verb HasSensitive

Cardinality

0..m

Ordered

Yes

Controlling

No

Inverse Mapping Semantics:

Verb SensedBy

Cardinality

1..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWIMS

Storage View File Name

ewsisn

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSSource

Cache View Name

DSSource

Storage View Name

DShasSource

Description

Instances of this relationship type are used to connect a SegmentLogical instance to the one or two instances of Segment that represent the physical segments used as source segments. This relationship maps to the SOURCE keyword on the SEGM statement in a DBD with access=LOGICAL.

Source Information:**Class Name**

DSSegmentLogical

Attribute

sourceSegments

Target Information:**Class Name**

DSSegment

Attribute

logicalSegments

Forward Mapping Semantics:

Verb HasSource

Cardinality

0..2

Ordered

Yes

Controlling

No

Inverse Mapping Semantics:

Verb IsSourceFor

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsisr

DLL Name

atlmod

Attributes

Name	Description
------	-------------

dataAlso	
-----------------	--

This attribute indicates how segment data will be handled when the logical DBD is processed.

A value of TRUE specifies use of "DATA" as the type in the SOURCE keyword in the generated DBD, which directs the segment key to be placed in the key feedback area and the segment data to be placed in the user's I/O area.

A value of FALSE specified use of "KEY" as the type, which directs only the key to be placed in the key feedback area.

Data Type	
------------------	--

boolean

Valid Values	
---------------------	--

TRUE, FALSE

Default	
----------------	--

TRUE

Cache View Methods

Name	Description
------	-------------

writeSOURCE	
--------------------	--

This instance method prepares the SOURCE keyword parameters, returning a string that does NOT include the SOURCE= keyword.

Return Type	
--------------------	--

string

Parameters	
-------------------	--

None

Relationship Class: DSHasSubsequence

Cache View Name	
------------------------	--

DSHasSubsequence

Storage View Name	
--------------------------	--

DSHasSubsequence

Description	
--------------------	--

Instances of this relationship type are used to connect a SecondaryIndex instance to the IMSField instances that are used by the secondary index as subsequence fields (SUBSEQ keyword on XDFLD statement).

Source Information:	
----------------------------	--

Class Name	
-------------------	--

DSXDFLD

Attribute	
------------------	--

subseq

Target Information:	
----------------------------	--

Class Name	
-------------------	--

DSIMSField

Attribute
isSubsequence

Forward Mapping Semantics:

Verb HasSubsequence

Cardinality
0..5

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsSubsequenceFor

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsisq

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSDSColl

Cache View Name
DSDSColl

Storage View Name
DSImplementedBy

Description

This relationship type is the base class for IMSDataset and IMSArea. Instances of those subclasses are used to connect a DBD instance to the instances of DDNAME that represent the DD statements used by the DBD (DD1 attribute on DATASET or AREA statement).

Source Information:

Class Name
DSDBD

Attribute
datasets

Target Information:

Class Name
DSDDNAME

Attribute
implementedDBDs

Forward Mapping Semantics:

Verb ImplementedBy

Cardinality
0..m

Ordered
Yes

Controlling
No

Inverse Mapping Semantics:

Verb Implements

Cardinality
0..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWIMS

Storage View File Name
ewsiib

DLL Name
atlmod

Attributes

Name	Description
------	-------------

areaAttributes

This attribute points to the attributes for an AREA statement that might apply to this relationship between DBD and DDNAME.

Data Type
relationship to IMSArea

Valid Values
0 or 1 relationship instance

datasetAttributes

This attribute points to the attributes for a DATASET statement that might apply to this relationship between DBD and DDNAME.

Data Type
relationship to IMSDataset

Valid Values

0 or 1 relationship instance

dsDevice

This attribute holds a string that represents the type of storage unit on which named dataset exists.

Data Type

char*

Valid Values

- "2305"
- "2319"
- "3330"
- "3350"
- "3375"
- "3380"
- "3390"
- "2400"
- "3400"
- "TAPE"
- null

Default

null

dsModel

This attribute holds a string that represents the model number for a 2305 or 3330 device.

Data Type

char*

Valid Values

- "1"
- "2"
- "11"
- null

Default

null

Cache View Methods

Name	Description
------	-------------

getSize	
----------------	--

	This instance method returns the size of the AREA or DATASET represented by this ImplementedBy relationship.
--	--

Return Type	
--------------------	--

	ulNT2
--	-------

Parameters	
-------------------	--

	None
--	------

invalidDDName

This instance method checks a DDNAME against the reserved names. It returns TRUE if the DDNAME is a reserved word.

Return Type

boolean

Parameters

Name <Direction>Type

name <Input > string

maximumBlock

This instance method returns the maximum blocksize for datasets or areas on the specified device type.

Return Type

uINT4

Parameters

Name <Direction>Type

accessMethod

<Input > string

maximumTrack

This instance method returns the maximum tracks for datasets or areas on the specified device type.

Return Type

uINT4

Parameters

None

validateDATASET

This instance method validates the DATASET macro statement with reference to DBDGEN. The validation messages are returned in a formatted string.

Return Type

boolean

Parameters

Name <Direction>Type

messages

<Output> string

accessMethod

<Input > string

osAccess

<Input > boolean

Relationship Class: DSHasArea**Cache View Name**

DSHasArea

Storage View Name

DSHasArea

Description

Instances of this relationship type are used to add the attributes used for an AREA macro statement to the relationship between a DBD instance and an instance of DDNAME.

Source Information:**Class Name**

DSDSColl

Attribute

areaAttributes

Target Information:**Class Name**

DSArea

Attribute

collector

Forward Mapping Semantics:

Verb HasAttributes

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb HasAttributesFor

Cardinality

1..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWIMS

Storage View File Name

ewsiar2

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasDataset

Cache View Name

DSHasDataset

Storage View Name

DSHasDataset

Description

Instances of this relationship type are used to add the attributes used for an DATASET macro statement to the relationship between a DBD instance and an instance of DDNAME.

Source Information:**Class Name**

DSDSColl

Attribute

datasetAttributes

Target Information:**Class Name**

DSIMSDataset

Attribute

collector

Forward Mapping Semantics:

Verb HasAttributes

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb HasAttributesFor

Cardinality

1..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWIMS

Storage View File Name

ewsids5

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasIMSfldName

Cache View Name

DSHasIMSfldName

Storage View Name

DSHasIMSfldName

Description

An instance of this relationship represents the association between an IMS field and the IMS-valid name for the data item.

Source Information:

Class Name

DSIMSField

Attribute

fldName

Target Information:

Class Name

DSSDDAlias

Attribute

fieldNames

Forward Mapping Semantics:

Verb HasFldNameIn

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb IsFldNameFor

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsifn

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSHasIMSsegmName**Cache View Name**

DSHasIMSsegmName

Storage View Name

DSHasIMSsegmName

Description

An instance of this relationship represents the association between an IMS segment and the IMS-valid name for the data structure.

Source Information:**Class Name**

DSSegment

Attribute

segmName

Target Information:**Class Name**

DSDataStructureAlias

Attribute

segmNames

Forward Mapping Semantics:**Verb** HasIMSNameIn**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsIMSNameFor**Cardinality**

0..m

Ordered

No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsism

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSIncludesPCB

Cache View Name
DSIncludesPCB

Storage View Name
DSIncludesPCB

Description
Instances of this relationship type are used to connect a PSB instance to the PCB instances that represent the PCBs used in the PSB user object.

Source Information:

Class Name
DPSB

Attribute
listPCBs

Target Information:

Class Name
DPCB

Attribute
usingPSBs

Forward Mapping Semantics:

Verb Includes

Cardinality
0..m

Ordered
Yes

Controlling
No

Inverse Mapping Semantics:

Verb IncludedIn

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsip

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSIndexedBy

Cache View Name
DSIndexedBy

Storage View Name
DSIndexedBy

Description
Instances of this relationship type are used to connect a SensitiveSegment instance to the DBDindex instances used as INDICES by the SENSEG.

Source Information:

Class Name
DSSenSegment

Attribute
indices

Target Information:

Class Name
DSDBDindex

Attribute
indexSenseg

Forward Mapping Semantics:

Verb IndexedBy

Cardinality
0..32

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Indexes

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsixb

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSIsPartOf

Cache View Name
DSIsPartOf

Storage View Name
DSIsPartOf

Description

Instances of this relationship type are used to connect SegmentComplex instances to the IMSDataset instance that represent the dataset in which segment data will be stored. (this information is specified in the IMS source by the sequence of DATASET and SEGM statements).

Source Information:

Class Name
DSSegmentComplex

Attribute
dataset

Target Information:

Class Name
DSIMSDataset

Attribute
segments

Forward Mapping Semantics:

Verb IsPartOf

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Includes

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsipo

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSIsUseOf

Cache View Name
DSIsUseOf

Storage View Name
DSIsUseOf

Description

Instances of this relationship type are used to connect each SecondaryIndex instance to the DBDIndex instance that acts as a secondary indexes for the DBD user object (this information maps to the combination of LCHILD and XDFLD statements).

Source Information:

Class Name
DSXDFLD

Attribute
targetDBDIndex

Target Information:

Class Name
DSDBDIndex

Attribute
secondaryIndexTarget

Forward Mapping Semantics:

Verb IsUseOf

Cardinality
1..1

Ordered
No

Controlling
Yes

Inverse Mapping Semantics:

Verb IsUsedBy

Cardinality
0..1

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsiiu

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSMakesUseOf

Cache View Name
DSMakesUseOf

Storage View Name
DSMakesUseOf

Description
An instance of this relationship type is used to connect an IMSField instance to a Dataltem instance that is local to the IMS segment. This relationship and SharesUseOf are mutually exclusive.

Source Information:

Class Name

DSIMSField

Attribute

targetDataItem

Target Information:**Class Name**

DSDataItem

Attribute

fieldsForIMS

Forward Mapping Semantics:**Verb** MakesUseOf**Cardinality**

1..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsUsedAs**Cardinality**

0..m

Ordered

No

Controlling

Yes

Cache View File Name

EWSWIMS

Storage View File Name

ewsimu

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSMappedBy

Cache View Name

DSMappedBy

Storage View Name

DSMappedBy

Description

An instance of this relationship type is used to connect a Segment instance to the shareable data structure that provides a mapping for fields in the segment.

Source Information:**Class Name**

DSSegment

Attribute

mappingStructure

Target Information:**Class Name**

DSDataStructure

Attribute

imsMappedSegments

Forward Mapping Semantics:

Verb MappedBy

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Maps

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsimb

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSMapsTo

Cache View Name

DSMapsTo

Storage View Name

DSMapsTo

Description

Instances of this relationship type are used to connect a SensitiveSegment instance to the Segment instance that represents a segment to which the PCB is sensitive.

Source Information:**Class Name**

DSSenSegment

Attribute

targetSegment

Target Information:**Class Name**

DSSegment

Attribute

sensitiveSegments

Forward Mapping Semantics:

Verb MapsTo

Cardinality

1..1

Ordered

No

Controlling

Yes

Inverse Mapping Semantics:

Verb Maps

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsimt

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSRefersTo

Cache View Name

DSRefersTo

Storage View Name

DSRefersTo

Description

An instance of this relationship type is used to connect a PCBteleprocessing instance to the OutputMessageDestination instance that represents the output destination for the teleprocessing PCB.

Source Information:

Class Name

DSPCBtp

Attribute

destination

Target Information:

Class Name

DSOutputDest

Attribute

isReferredToBy

Forward Mapping Semantics:

Verb RefersTo

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb ReferredToBy

Cardinality

0..m

Ordered

No

Controlling

Yes

Cache View File Name

EWSWIMS

Storage View File Name

ewsirt

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSSequencedBy**Cache View Name**

DSSequencedBy

Storage View Name

DSSequencedBy

Description

An instance of this relationship type is used to connect a PCBdatabase instance to the DBDindex instance that represents the PROCSEQ for the PCB user object.

Source Information:**Class Name**

DSPCdb

Attribute

procseq

Target Information:**Class Name**

DSDBIndex

Attribute

sequencedPCB

Forward Mapping Semantics:**Verb** SequencedBy**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** Sequences**Cardinality**

0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsisb

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSSharesWith

Cache View Name
DSSharesWith

Storage View Name
DSSharesWith

Description

An instance of this relationship type is used to connect an instance of DBDindex to the instance of DBDindex that is implemented by an instance of IMSDataset. This relationship models the shared secondary index DBD collection.

Source Information:

Class Name
DSDBDindex

Attribute
sharedDBD

Target Information:

Class Name
DSDBDindex

Attribute
sharingDBDs

Forward Mapping Semantics:

Verb SharesWith

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb SharedBy

Cardinality
0..16

Ordered
No

Controlling
No

Cache View File Name
EWSWIMS

Storage View File Name
ewsisw

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSSpecifiedByType

Cache View Name
DSSpecifiedByType

Storage View Name
DSSpecifiedByType

Description

An instance of this relationship type is used to connect an instance of PCB to an instance of PCBtype or one of its subclasses in order to hold the type-specific attributes of a PCB. Using this class instead of subclassing PCB directly allows a tool to change the type of a PCB (or decide the type after instantiation) without having to change the class of the PCB.

Source Information:

Class Name
DSPCB

Attribute
typeOfPCB

Target Information:

Class Name
DSPCBtype

Attribute
typedPCB

Forward Mapping Semantics:

Verb SpecifiedBy

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Specifies

Cardinality
1..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWIMS

Storage View File Name
ewsifb

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSUsedAsSENFLD

Cache View Name
DSUsedAsSENFLD

Storage View Name
DSUsedAsSENFLD

Description
Instances of this relationship type are used to connect a SensitiveField instance to the Field instance that represents a field to which the sensitiveSegment is sensitive.

Source Information:

Class Name
DSSenField

Attribute
targetFIELD

Target Information:**Class Name**

DSIMSField

Attribute

senflds

Forward Mapping Semantics:**Verb** Uses**Cardinality**

1..1

Ordered

No

Controlling

Yes

Inverse Mapping Semantics:**Verb** IsUsedAs**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWIMS

Storage View File Name

ewsisf1

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSUsesSENFLD

Cache View Name

DSUsesSENFLD

Storage View Name

DSUsesSENFLD

Description

Instances of this relationship type are used to connect a SensitiveField instance to the SensitiveSegment that is sensitive to a field.

Source Information:

Class Name
DSSenSegment

Attribute
senflds

Target Information:

Class Name
DSSenField

Attribute
sourceSENSEG

Forward Mapping Semantics:

Verb Uses

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb UsedBy

Cardinality
1..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWIMS

Storage View File Name
ewsisf2

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSUsesDBDexit

Cache View Name
DSUsesDBDexit

Storage View Name
DSUsesDBDexit

Description

An instance of this relationship type is used to connect a DBD to the DataPropagationExit that represents the data capture routine (EXIT keyword on DBD statement).

Source Information:**Class Name**

DSDBD

Attribute

exits

Target Information:**Class Name**

DSExit

Attribute

capturedDBD

Forward Mapping Semantics:

Verb DataCapturedBy

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb CapturesDataFor

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWIMS

Storage View File Name

ewsidx

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSUsesSEGMexit

Cache View Name

DSUsesSEGMexit

Storage View Name

DSUsesSEGMexit

Description

An instance of this relationship type is used to connect a DBD to the DataPropagationExit that represents the data capture routine (EXIT keyword on SEGM statement).

Source Information:**Class Name**

DSSegment

Attribute

exits

Target Information:**Class Name**

DSExit

Attribute

capturedSegment

Forward Mapping Semantics:

Verb DataCapturedBy

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb CapturesDataFor

Cardinality

0..m

Ordered

No

Controlling

Yes

Cache View File Name

EWSWIMS

Storage View File Name

ewsisx

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSSharesUseOf

Cache View Name

DSSharesUseOf

Storage View Name

DSSharesUseOf

Description

An instance of this relationship type is used to connect an IMSField instance to a DataItem instance that is contained in the DataStructure instance connected to the Segment. This relationship and MakesUseOf are mutually exclusive.

Source Information:

Class Name

DSIMSField

Attribute

sharedDataItem

Target Information:

Class Name

DSDataItem

Attribute

fieldsSharedByIMS

Forward Mapping Semantics:

Verb SharesUseOf

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb IsSharedAs

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name
EWSWIMS

Storage View File Name
ewsisu

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Chapter 8. High Level Languages Model

Object Class: DSArray

Cache View Name

DSArray

Storage View Name

DSArray

Inherits From

DSSharedDataDefinition

Description

An object of this class represents a *shareable named* array.

Short name: Array

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWHLL

Storage View File Name

ewshc01

DLL Name

atlmod

Attributes

Name	Description
------	-------------

pArrayDef	
------------------	--

This attribute specifies an instance of the relationship
Array_DefinedBy_ArrayDef.

Data Type

relationship to ArrayDef

Valid Values

0 or 1 relationship instance

Cache View Methods

None

Object Class: DSUnion

Cache View Name

DSUnion

Storage View Name

DSUnion

Inherits From
DSSharedDataDefinition

Description
An object of this class represents a ***shareable named*** HLL union.

Short name: Union

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWHLL

Storage View File Name
ewshc05

DLL Name
atlmod

Attributes

Name	Description
pUnionDef	This attribute specifies an instance of the relationship Union_DefinedBy_UnionDef.
Data Type	relationship to UnionDef
Valid Values	0 or 1 relationship instance

Cache View Methods

None

Object Class: DSEnumeratedList

Cache View Name
DSEnumeratedList

Storage View Name
DSEnumeratedList

Inherits From
DSSharedDataDefinition

Description
An object of this class represents a ***shareable named*** enumeraetd list.

Short name: EnumeratedList

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWHLL

Storage View File Name
ewshc03

DLL Name
atlmod

Attributes

Name	Description
pEnumDef	This attribute specifies an instance of the relationship EnumLst_DefinedBy_EnumDef.
Data Type	relationship to EnumerationDef
Valid Values	0 or 1 relationship instance

Cache View Methods

None

Object Class: DSPointer

Cache View Name
DSPointer

Storage View Name
DSPointer

Inherits From
DSSharedDataDefinition

Description
An object of this class represents a **shareable named** pointer.
Short name: Pointer

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWHLL

Storage View File Name
ewshc07

DLL Name
atImod

Attributes

Name	Description
------	-------------

pPointerDef	
--------------------	--

	This attribute specifies an instance of the relationship Pointer_DefinedBy_PtrDef.
--	--

Data Type	
------------------	--

	relationship to PointerDef
--	----------------------------

Valid Values	
---------------------	--

	0 or 1 relationship instance
--	------------------------------

Cache View Methods

None

Object Class: DSArrayDef

Cache View Name

DSArrayDef

Storage View Name

DSArrayDef

Inherits From

DSDataDefinition

Description

An instance of this class represents the definition of one dimension of an array. Multi-dimensional array is represented using the relationship ArrayDef_HasElement_DataDef and the related classes.

Constraints

1. An object of this class can be the source of only one of the following relationships:
 - **ArrayDef_HasElement_TDRef**
 - **ArrayDef_HasElement_DataDef**
2. An object of this class can be the target of only one of the following relationships:
 - **Array_DefinedBy_ArrayDef**
 - **DataItn_DefinedBy_DataDef**

Short name: ArrayDef

Public Class

Yes

Extends

None

Class Extent
None

Cache View File Name
EWSWHLL

Storage View File Name
ewshc02

DLL Name
atlmod

Attributes

Name	Description
------	-------------

indexes

This attribute holds a string of maximum 12 COBOL data names that are specified after the COBOL clauses OCCURS ... INDEXED BY

Data Type

char*

keys This attribute collects instances of the relationship ArrayDef_HasKey_Dataitm.

Data Type

relationship to DataItem

Valid Values

0 or 1 relationship instance

lowerBound

This attribute specifies an integer representing the lower bound of an array dimension.

Data Type

long

Valid Values

non negative integer

Default

1

Update Constraints

(document only)

This attribute is mutually exclusive with the following two attributes: **lowerBoundExpression** and **lowerBoundVariable**.

lowerBoundExpression

This attribute holds the lower bound expression of an array dimension.

This attribute is applicable to PL/I only.

Data Type

char*

Valid Values

any valid PL/I expression

Update Constraints

(document only)

This attribute and the attribute ***lowerBound*** are mutually exclusive.

lowerBoundVariable

This attribute specifies an instance of the relationship
ArrayDef_LBRefersTo_DataItem.

This attribute is applicable to PL/I only.

Data Type

relationship to DataItem

Valid Values

0 or 1 relationship instance

Update Constraints

(document only)

This attribute and the attribute ***lowerBound*** are mutually exclusive.

minimumExtent

For COBOL only, This attribute specifies an integer representing the
integer-1 in the ...OCCURS integer-1 TO integer-2 TIMES... clause.

The integer-2, which is also called maximum extent, in the above clause is
derived using the following formula:

maximum extent = upperBound - lowerBound + 1

Data Type

unsigned long

Valid Values

non negative integer

Default

0

Update Constraints

(document only)

minimumExtent <= maximum extent

pDataDefinition

This attribute specifies an instance of the relationship
ArrayDef_HasElement_DataDef.

Data Type

relationship to DataDefinition

Valid Values

0 or 1 relationship instance

pShareableElement

This attribute specifies an instance of the relationship
ArrayDef_HasElement_SharedDataDef.

Data Type

relationship to SharedDataDefinition

Valid Values

0 or 1 relationship instance

pSharedArray

This attribute specifies an instance of the relationship
ArrayDef_Defines_Array.

Data Type

relationship to Array

Valid Values

0 or 1 relationship instance

pTypeDefReference

This attribute specifies an instance of the relationship
ArrayDef_HasElement_TDRef.

Data Type

relationship to TypeDefReference

Valid Values

0 or 1 relationship instance

upperBound

This attribute specifies an integer representing the upper bound of an array
dimension.

Data Type

long

Valid Values

non negative integer

Default

lowerBound + 1

Update Constraints**(document only)**

This attribute is mutually exclusive with the following two
attributes: **lowerBoundExpression** and
lowerBoundVariable.

upperBoundExpression

This attribute specifies the upper bound expression of an array dimension.

This attribute is applicable to PL/I only.

Data Type

char*

Valid Values

any valid PL/I expression

Update Constraints**(document only)**

This attribute and the attribute **upperBound** are mutually
exclusive.

upperBoundVariable

This attribute specifies an instance of the relationship
ArrayDef_UBRefersTo_DataItem.

Data Type

relationship to DataItem

Update Constraints

(document only)

This attribute and the attribute ***upperBound*** are mutually exclusive.

Cache View Methods

Name	Description
------	-------------

asCobolSource	
----------------------	--

This method returns a string in the format of a COBOL statement that corresponds to the array definitions represented by this object.

This method generates the details of the COBOL OCCURS clause:

- OCCURS ... TO ... TIMES
- DEPENDING ON ...
- INDEXED BY
- KEY IS

The ***elementTypeAsCobolSource()*** method generates the details of the array's element type definitions. The sender (caller) is responsible for deleting the returned object.

Return Type	
--------------------	--

string

Parameters	
-------------------	--

Name	<Direction>	Type
-------------	--------------------------	-------------

&aParmList		
-----------------------	--	--

<Input > DSHLLParmList

DSArrayDefnIt	
----------------------	--

This method initializes the default values for all attributes of this object.

This method is invoked from the constructor.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

None

elementTypeAsCobolSource	
---------------------------------	--

This method generates, into the source file specified in the parameter list, the details of the array's element type definitions.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

Name	<Direction>	Type
-------------	--------------------------	-------------

&aParmList		
-----------------------	--	--

<Input > DSHLLParmList

parentLevel		
--------------------	--	--

<Input > ulNT2

previousLevelPos		
-------------------------	--	--

<Input > ulNT2

elementTypeAsEnum

This method returns a dsEnumDataDef representing the array's element type. The sender (caller) is responsible for deleting the returned object.

Return Type

dsEnumDataDef

Parameters

None

elementTypeAsString

This method returns a string representing the array's element type. The sender (caller) is responsible for deleting the returned object.

Return Type

string

Parameters

Name <Direction>Type

&aParmList

<Input > DSHLLParmList

UBVariableAsString

This method returns a string representing the array's upper bound variable that is specified after the PL/I REFER option. The sender (caller) is responsible for deleting the returned object.

Return Type

string

Parameters

Name <Direction>Type

language

<Input > dsEnumLanguage

&aParmList

<Input > DSHLLParmList

Object Class: DSUnionDef

Cache View Name

DSUnionDef

Storage View Name

DSUnionDef

Inherits From

DSDataDefinition

Description

An instance of this object class represents C union data type, or PL/I UNION attribute.

Constraints

1. An object of this class can be the target of only one of the following relationships:
 - *Union_DefinedBy_UnionDef*
 - *DataItem_DefinedBy_DataDef*

Short name: UnionDef

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWHLL

Storage View File Name

ewshc06

DLL Name

atlmod

Attributes

Name Description

dataMembers

This attribute collects instances of the relationship UnionDef_HasMember_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or more relationship instances

pSharedUnion

This attribute specifies an instance of the relationship UnionDef_Defines_Union.

Data Type

relationship to Union

Valid Values

0 or 1 relationship instance

Cache View Methods

None

Object Class: DSEnumerationDef

Cache View Name

DSEnumerationDef

Storage View Name

DSEnumerationDef

Inherits From

DSDataDefinition

Description

An instance of this class represents C *enum* data type or PL/I ordinal data type.

Constraints

1. An object of this class can be the target of only one of the following relationships:

- *EnumLst_DefinedBy_EnumDef*
- *DataItn_DefinedBy_DataDef*

Short name: EnumerationDef

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWHLL

Storage View File Name

ewshc04

DLL Name

atlmod

Attributes

Name Description

idList This attribute specifies the list of identifiers of the enumeration.

Data Type

char*

precision

This attribute specifies the size in bits of each identifier of the enumeration.

Data Type

unsigned short

Default

16

pSharedEnumList

This attribute specifies an instance of the relationship EnumDef_Defines_EnumLst.

Data Type

relationship to EnumeratedList

Valid Values

0 or 1 relationship instance

Cache View Methods

Name Description**DSEnumerationDefInit**

This method initializes the default values for all attributes of this object.

This method is invoked from the constructor.

Return Type

void

Parameters

None

Object Class: DSPointerDef

Cache View Name

DSPointerDef

Storage View Name

DSPointerDef

Inherits From

DSDataDefinition

Description

An instance of this object class represents a high level language pointer type.

Constraints

1. An object of this class can be the source of only one of the following relationships:
 - ***PtrDef_PointsTo_TDRef***
 - ***PtrDef_PointsTo_DataDef***
2. An object of this class can be the target of only one of the following relationships:
 - ***Pointer_DefinedBy_PtrDef***
 - ***DataItn_DefinedBy_DataDef***

Short name: PointerDef

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWHLL

Storage View File Name

ewshc08

DLL Name

atlmod

Attributes

Name Description**isCobolProcPointer**

This attribute indicates whether the COBOL USAGE POINTER or the COBOL USAGE PROCEDURE-POINTER is specified.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

pDataDefinition

This attribute specifies an instance of the relationship PtrDef_PointsTo_DataDef.

Data Type

relationship to DataDefinition

Valid Values

0 or 1 relationship instance

pSharedPointer

This attribute specifies an instance of the relationship PtrDef_Defines_Pointer.

Data Type

relationship to Pointer

Valid Values

0 or 1 relationship instance

pTypeDefReference

This attribute specifies an instance of the relationship PtrDef_PointsTo_TDRef.

Data Type

relationship to TypeDefReference

Valid Values

0 or 1 relationship instance

Cache View Methods

Name Description**pointerTypeAsEnum**

This method returns a fdcEnumDataDef representing the pointer's data type. The sender (caller) is responsible for deleting the returned object.

Return Type

dsEnumDataDef

Parameters

None

pointerTypeAsString

This method returns a string representing the pointer's data type. The sender (caller) is responsible for deleting the returned object.

Return Type

string

Parameters

None

Object Class: DSIncludedSourceDef
Cache View Name

DSIncludedSourceDef

Storage View Name

DSIncludedSourceDef

Inherits From

DSTechnologyObject

Description

An instance of this object class represents a COBOL COPY, PL/I %Include, or C #include file contained in a source program. Each instance may be included multiple times and may contain nested include files.

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWHLL

Storage View File Name

ewsaisd

DLL Name

atlmod

Attributes**Name Description****includeComponents**

This attribute points to an ordered collection of pointers to the include components that are contained in this includedSource.

Data Type

relationship to IncludeComponent

Valid Values

empty collection or pointer collections

Update Constraints

(document only)

none

includeStatements

This attribute points to the IncludedStatement instances which reference the given IncludedSource program instance.

Data Type

relationship to IncludeStatement

Valid Values

empty collection or pointer collection

Update Constraints

(document only)

none

isrLanguage

This attribute specifies the compiler language used to compile the given source instance.

Data Type

char*

Valid Values

string character set

Default

null

Update Constraints

(document only)

none

isrProgrammer

This attribute specifies the owner for the given source instance.

Data Type

char*

Valid Values

string character

Default

null

Update Constraints

(document only)

none

Cache View Methods

Name Description

asCobolSource

Return Type

void

Parameters

Name <Direction>Type

&aParmList
<Input > DSHLLParmList

getApplication

This method finds all the applications that use this included source program

Return Type
void

Parameters
None

getExecutable

This method finds all the executable programs using this included source program

Return Type
void

Parameters
None

getJob

This method finds all the jobs that use this included source program

Return Type
void

Parameters
None

getNestedIncludedSource

This method finds the nested include programs contained in this included source program

Return Type
void

Parameters
None

getReconcilableObjects

This method finds all the reconcilable objects (DSDataItems that do not have pStructureDefs) in a DSIncludedSourceDef and puts them in a collection. Note: This method is recursive.

Return Type
void

Parameters

Name <Direction>Type
****ppRecObjectColl**
<Input > DSCollection
***pParmList**
<Input > DSParmList

reconcile

Return Type
void

Parameters

Name <Direction>Type

***pParmList**
 <Input > DSParmList

getProgramObject
 This method finds all the object programs using this include source progr

Return Type
 void

Parameters
 None

Object Class: DSIncludeStatement

Cache View Name
 DSIncludeStatement

Storage View Name
 DSIncludeStatement

Inherits From
 DSDataComponent

Description
 An instance of this class represents a COBOL COPY, PL/I %INCLUDE, or C #include statement.

Short name: IncldStm

Public Class
 Yes

Extends
 None

Class Extent
 None

Cache View File Name
 EWSWHLL

Storage View File Name
 ewshc15

DLL Name
 atlmod

Attributes

Name	Description
language	This attribute specifies the language to be used in validating/generating the <i>text</i> attribute.
Data Type	DSEnum::Language
Valid Values	<ul style="list-style-type: none"> LANGUAGE_NEUTRAL LANGUAGE_PLI LANGUAGE_C

	Default LANGUAGE_NEUTRAL
referencedInclSource	This attribute specifies an instance of the relationship Id2is..
	Data Type relationship to IncludedSourceDef
	Valid Values 0 or 1 relationship instance
text	This attribute specifies the text string referred to in the COBOL COPY, PL/I %INCLUDE, or C #include statement.
	Data Type char*
	Valid Values A text string conforming to the syntax of the language specified in the <i>language</i> attribute.
	Default null

Cache View Methods

Name	Description				
asCobolStatement	This method returns the COBOL source code represented by this object. The sender (caller) is responsible for deleting the returned object.				
	Return Type string				
	Parameters				
	<table> <tr> <th>Name</th><th><Direction>Type</th></tr> <tr> <td>&aParmList</td><td><Input > DSHLLParmList</td></tr> </table>	Name	<Direction>Type	&aParmList	<Input > DSHLLParmList
Name	<Direction>Type				
&aParmList	<Input > DSHLLParmList				
asPliStatement	This method returns the PL/I source code represented by this object. The sender (caller) is responsible for deleting the returned object.				
	Return Type string				
	Parameters				
	<table> <tr> <th>Name</th><th><Direction>Type</th></tr> <tr> <td>&aParmList</td><td><Input > DSHLLParmList</td></tr> </table>	Name	<Direction>Type	&aParmList	<Input > DSHLLParmList
Name	<Direction>Type				
&aParmList	<Input > DSHLLParmList				

Object Class: DSCobolDataDefinition

Cache View Name	DSCobolDataDefinition
Storage View Name	DSCobolDataDefinition

Inherits From

DSDataComponent

Description

An instance of this class represents the COBOL-specific data definitions for a data item.

Short name: CblDef

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWHLL

Storage View File Name

ewshc12

DLL Name

atlmod

Attributes

Name Description**bBlankWhenZero**

This attribute specifies a flag indicating whether or not the BLANK WHEN ZERO clause is specified.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

bGlobal

This attribute specifies a flag indicating whether or not the GLOBAL clause is specified.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

bJustified

This attribute specifies a flag indicating whether or not the JUSTIFIED clause is specified.

Data Type

boolean

Valid Values

TRUE, FALSE

Default
FALSE

refinedDataItem
This attribute specifies an instance of the relationship
DataItem_RefinedBy_CblDef.

Data Type
relationship to DataItem

Valid Values
1 relationship instance

Cache View Methods

Name	Description
------	-------------

DSCobolDataDefinitionInit	This method initializes the default values for all attributes of this object. This method is invoked from the constructor.
----------------------------------	---

Return Type	void
--------------------	------

Parameters	None
-------------------	------

Object Class: DSCobolSpecialDataName

Cache View Name
DSCobolSpecialDataName

Storage View Name
DSCobolSpecialDataName

Inherits From
DSDataComponent

Description
An instance of this object class represents either a COBOL level-66 or level-88 data names.

Short name: CblSpNm

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWHLL

Storage View File Name
ewshc13

DLL Name
atlmod

Attributes

Name	Description
------	-------------

conditionValues	
------------------------	--

This attribute specifies level-88 VALUE clause.

Data Type

char*

Update Constraints

(document only)

This attribute is applicable only if this object is the target of an instance of the relationship DataItem_HasLevel88_CbISpNm.

conditionVariable	
--------------------------	--

This attribute specifies an instance of the relationship CbISpNm_IsLevel88Of_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or 1 relationship instance

endRenamedItem	
-----------------------	--

This attribute specifies an instance of the relationship CbISpNm_EndRenames_DataItem.

Data Type

relationship to DataItem

Valid Values

0 or 1 relationship instance

Update Constraints

(document only)

This attribute is applicable only if this object is the target of an instance of the relationship HLLRecrd_HasLevel66_CbISpNm.

name	This attribute specifies a COBOL data name.
-------------	---

Data Type

char*

Valid Values

a string or *null*

Default

null

Update Constraints

(document only)

As follows

- Maximum 30 characters
- Valid characters: A through Z, a through z, 0 through 9, and hyphen (-)
- At least one alphabetic character

- The hyphen may not appear as the first or the last character

startRenamedItem

This attribute specifies an instance of the relationship CblSpNm_StartRenames_Dataltm.

Data Type

relationship Dataltem

Valid Values

0 or 1 relationship instance

Update Constraints

(document only)

This attribute is applicable only if this object is the target of an instance of the relationship HLLRecrd_HasLevel66_CblSpNm.

level66Dataltem

This attribute specifies an instance of the relationship REQCLEANUP - Created by ActiveSystems 2-19-97 Entity not defined..

Data Type

relationship to Dataltem

Valid Values

0 or 1 relationship instance

Cache View Methods

Name Description

renamedItemName

This method returns a string representing the COBOL data name that is specified after the RENAMES clause. The sender (caller) is responsible for deleting the returned object.

Return Type

string

Parameters

None

renamedThruItemName

This method returns a string representing the COBOL data name that is specified after the RENAMES...THRU clause. The sender (caller) is responsible for deleting the returned object.

Return Type

string

Parameters

None

Object Class: DSPLIDataDefinition

Cache View Name

DSPLIDataDefinition

Storage View Name

DSPLIDataDefinition

Inherits From

DSDataComponent

Description

An instance of this object class represents PL/I-specific data definitions for a data item.

Short name: PLIDef

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWHLL

Storage View File Name

ewshc14

DLL Name

atlmod

Attributes

Name	Description
------	-------------

hasBuiltInAttribute

This attribute specifies whether or not the BUILTIN attribute is declared for a data item.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

HasOptionalAttribute

This attribute specifies whether or not the OPTIONAL attribute is declared for a data item.

Data Type

boolean

Valid Values

TRUE, FALSE

Default

FALSE

hasParameterAttribute

This attribute specifies whether or not the PARAMETER attribute is declared for a data item.

Data Type

boolean

Valid Values

TRUE, FALSE

Default
FALSE

hasReservedAttribute

This attribute specifies whether or not the RESERVED attribute is declared for a data item.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

hasSystemAttribute

This attribute specifies whether or not the SYSTEM attribute is declared for a data item.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

isAssignable

This attribute specifies whether ASSIGNABLE or UNASSIGNABLE attribute is declared for a data item.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

isConnected

This attribute specifies whether or not the CONNECTED attribute is declared for a data item.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

isNative

This attribute specifies whether NATIVE or UNNATIVE attribute is declared for a data item.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

isNormal
This attribute specifies whether NORMAL or UNNORMAL attribute is declared for a data item.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

refinedDataItem
This attribute specifies an instance of the relationship PLIDef_Refines_DataItem.

Data Type
relationship to DataItem

Valid Values
1 instance

storageClass
This attribute specifies the storage class for a data item.

Data Type
DSEnum::PLIStorage

Valid Values

- PS_AUTO
- PS_BASED
- PS_CONTROLLED
- PS_STATIC
- PS_UNSPEC

Default
PS_UNSPEC

Cache View Methods

Name	Description
------	-------------

DSPLIDataDefinitionInit	This method initializes the default values for all attributes of this object. This method is invoked from the constructor.
--------------------------------	---

Return Type	void
--------------------	------

Parameters	None
-------------------	------

Object Class: DSPLIArea

Cache View Name	DSPLIArea
------------------------	-----------

Storage View Name
DSPLIArea

Inherits From
DSDataDefinition

Description
An instance of this object class represents PL/I AREA data attribute.

Short name: Area

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWHLL

Storage View File Name
ewshc09

DLL Name
atlmod

Attributes

Name	Description
sizeExpression	This attribute specifies the expression for the area size.
Data Type	char*
sizeVariable	This attribute specifies an instance of the relationship PLIArea_SizeRefersTo_DataItem.
Data Type	relationship to DataItem
Valid Values	0 or 1 relationship instance

Cache View Methods

Name	Description
areaTypeAsString	This method returns a string representing the PL/I area type The sender (caller) is responsible for deleting the returned object.
Return Type	string
Parameters	None

Object Class: DSOffset

Cache View Name

DSOffset

Storage View Name

DSOffset

Inherits From

DSDataDefinition

Description

An instance of this object class represents PL/I OFFSET data attribute.

Short name: Offset

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWHLL

Storage View File Name

ewshc10

DLL Name

atImod

Attributes

Name	Description
------	-------------

expression	
-------------------	--

This attribute holds the expression after the PL/I OFFSET attribute.

Data Type

char*

pPLIAreaVariable

This attribute specifies an instance of the relationship

Offset_InArea_DataItem

Data Type

relationship to DataItem

Valid Values

0 or 1 relationship instance

Update Constraints

(document only)

The target data item must be of type PLIArea.

Cache View Methods

Name	Description
------	-------------

areaDataName	
---------------------	--

This method returns the data name representing the PL/I AREA variable

that is specified after the PL/I OFFSET attribute. The sender (caller) is responsible for deleting the returned object.

Return Type
string

Parameters
None

offsetTypeAsString
This method returns the type of the OFFSET as a string. The sender (caller) is responsible for deleting the returned object.

Return Type
string

Parameters
None

Object Class: DSFunctionDeclaration

Cache View Name
DSFunctionDeclaration

Storage View Name
DSFunctionDeclaration

Inherits From
DSIncludeComponent

Description
An instance of this object class represents represents a function or procedure interface. For PL/I, it represents an ENTRY data declaration. For C, it represents a function prototype.

Short name: FnctnDcl

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWHLL

Storage View File Name
ewshc16

DLL Name
atlmod

Attributes

Name	Description
------	-------------

cFnPrototype	This attribute specifies an instance of the relationship FnctnDcl_RefinedBy_CfnPrtp.
---------------------	--

Data Type
relationship to CFunctionPrototype

Valid Values
0 or 1 relationship instance

parameterType
This attribute collects instances of the relationship
FnctnDcl_HasParameter_DataDef.

Data Type
relationship to DataDefinition

Valid Values
0 or more relationship instances

pliEntryData
This attribute specifies an instance of the relationship
FnctnDcl_RefinedBy_PLIEntry.

Data Type
relationship to PLIEntryDataDCL

Valid Values
0 or 1 relationship instance

returnedType
This attribute specifies an instance of the relationship
FnctnDcl_Returns_DataDef.

Data Type
relationship DataDefinition

Valid Values
0 or 1 relationship instance

Cache View Methods

None

Object Class: DSPLIEntryDataDCL

Cache View Name
DSPLIEntryDataDCL

Storage View Name
DSPLIEntryDataDCL

Inherits From
DSIncludeComponent

Description
An instance of this object class represents PL/I ENTRY data declaration.

Short name: PLIEntry

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWHLL

Storage View File Name
ewshc17

DLL Name
atlmod

Attributes

Name	Description
------	-------------

hasVariableAttribute	
-----------------------------	--

This attribute specifies a flag indicating whether the VARIABLE attribute is explicitly declared for an entry variable.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

initialValue	
---------------------	--

This attribute specifies the string declared after the INITIAL attribute of an ENTRY variable. If this attribute is not null, it overrides the **name** attribute of the FunctionDeclaration class.

Data Type
char*

Default
null

Update Constraints

(document only)

As follows

- Maximum 31 characters
- Valid characters: extended alphabetic characters (A through Z, a through z, #, @, and \$), 0 through 9, and underscore (_)
- The first character must be one of the 29 extended alphabetic characters.

isExternal	
-------------------	--

This attribute specifies a flag indicating whether the EXTERNAL attribute is explicitly declared for an entry variable.

Data Type
boolean

Valid Values
TRUE, FALSE

Default
FALSE

name This attribute specifies the PL/I data name that is declared as an ENTRY variable. If this attribute is not null, it overrides the **name** attribute of the FunctionDeclaration class.

Data Type

char*

Valid Values

a PL/I data name or null

Default

null

Update Constraints

(document only)

As follows

- Maximum 31 characters
- Valid characters: extended alphabetic characters (A through Z, a through z, #, @, and \$), 0 through 9, and underscore (_)
- The first character must be one of the 29 extended alphabetic characters.

optionsList

This attribute specifies the list of options specified after the OPTIONS attribute in the entry variable declaration. variable.

Data Type

char*

Default

null

pFunctionDcl

This attribute specifies an instance of the relationship PLIEntry_Refines_FncnDcl.

Data Type

relationship to DataItem

Valid Values

instance

storageClass

This attribute specifies the storage class for an entry variable.

Data Type

DSEnum::PLIStorage

Valid Values

- PS_AUTO
- PS_BASED
- PS_CONTROLLED
- PS_STATIC
- PS_UNSPEC

Default

PS_UNSPEC

Cache View Methods

Name	Description
------	-------------

DSPLIEntryDataDCLInit	
------------------------------	--

This method initializes the default values for all attributes of this object.

This method is invoked from the constructor.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

None

Object Class: DSTypeDefReference

Cache View Name	
------------------------	--

DSTypeDefReference

Storage View Name	
--------------------------	--

DSTypeDefReference

Inherits From	
----------------------	--

DSIncludeComponent

Description	
--------------------	--

An instance of this class represents the include component that is a C/C++ **typedef** or PL/I **alias** declaration. This class is needed to maintain the relative position of a type definition within an include file.

Public Class	
---------------------	--

Yes

Extends	
----------------	--

None

Class Extent	
---------------------	--

None

Cache View File Name	
-----------------------------	--

EWSWHLL

Storage View File Name	
-------------------------------	--

ewshc19

DLL Name	
-----------------	--

atlmod

Attributes

Name	Description
------	-------------

arrayDefs	
------------------	--

This attribute collects instances of the relationship TDRef_IsElementOf_ArrayDef.

Data Type	
------------------	--

relationship to ArrayDef

Valid Values	
---------------------	--

0 or more relationship instances

cName

This attribute specifies an instance of the relationship
TDRef_HasCNameIn_SDDAlias.

Data Type

relationship to SDDAlias

Valid Values

0 or 1 relationship instance

pliName

This attribute specifies an instance of the relationship
TDRef_HasPLINameIn_SDDAlias.

Data Type

relationship to SDDAlias

Valid Values

0 or 1 relationship instance

pointerDefs

This attribute collects instances of the relationship TDRef_IsTypeOf_PtrDef.

Data Type

relationship to PointerDef

Valid Values

0 or more relationship instances

pSharedDataDef

This attribute specifies an instance of the relationship
TDRef_RefersTo_SDD.

Data Type

relationship to SharedDataDefinition

Valid Values

1 relationship instance

Cache View Methods

None

Object Class: DSCFunctionPrototype

Cache View Name

DSCFunctionPrototype

Storage View Name

DSCFunctionPrototype

Inherits From

DSIncludeComponent

Description

An instance of this class represents a C function prototype.

Short name: CFnPrtp

Public Class

Yes

Extends

None

Class Extent
None

Cache View File Name
EWSWHLL

Storage View File Name
ewshc18

DLL Name
atlmod

Attributes

Name	Description
pFunctionDcl	This attribute specifies an instance of the relationship CFnPrtp_Refines_FnctnDcl.
Data Type	relationship to FunctionDeclaration
Valid Values	1 relationship instance

Cache View Methods

None

Relationship Class: DSLinkArray2ArrayDef

Cache View Name
DSLLinkArray2ArrayDef

Storage View Name
DSLLinkArray2ArrayDef

Description
An instance of this relationship represents the association between a shareable array and its array type definition.

Source Information:

Class Name	DSArray
Attribute	pArrayDef

Target Information:

Class Name	DSArrayDef
Attribute	pSharedArray

Forward Mapping Semantics:

Verb	DefinedBy
Cardinality	1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Defines

Cardinality
0..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWHLL

Storage View File Name
ewshl09

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkUnion2UnionDef

Cache View Name
DSLinkUnion2UnionDef

Storage View Name
DSLinkUnion2UnionDef

Description
An instance of this relationship represents the association between a shareable union and its union type definition.

Source Information:

Class Name
DSUnion

Attribute
pUnionDef

Target Information:

Class Name
DSUnionDef

Attribute
pSharedUnion

Forward Mapping Semantics:

Verb DefinedBy

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Defines

Cardinality
0..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWHLL

Storage View File Name
ewshl14

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSEnumLst2EnumDef

Cache View Name
DSEnumLst2EnumDef

Storage View Name
DSEnumLst2EnumDef

Description
An instance of this relationship represents the association between a shareable enumerated list and its enumeration type definition.

Source Information:

Class Name
DSEnumeratedList

Attribute
pEnumDef

Target Information:

Class Name
DSEnumerationDef

Attribute
pSharedEnumList

Forward Mapping Semantics:

Verb DefinedBy

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Defines

Cardinality
0..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWHLL

Storage View File Name
ewshl16

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkPointer2PtrDef

Cache View Name
DSLinkPointer2PtrDef

Storage View Name
DSLinkPointer2PtrDef

Description
An instance of this relationship represents the association between a shareable pointer and its pointer type definition.

Source Information:

Class Name
DSPointer

Attribute
pPointerDef

Target Information:

Class Name
DSPointerDef

Attribute
pSharedPointer

Forward Mapping Semantics:

Verb DefinedBy

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Defines

Cardinality
0..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWHLL

Storage View File Name
ewshl17

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkIncSrcDef2InclCmp

Cache View Name
DSLinkIncSrcDef2InclCmp

Storage View Name
DSLinkIncSrcDef2InclCmp

Description

An instance of this relationship represents the association between an include file and the include component within the include file.

The include component can be a data item. If the data item is a substructure, **all** of its data components must be the target of this relationship.

- An include statement, which represents a nested COBOL COPY or PL/I %INCLUDE statement.
- A data definition such as typedef in C.
- A C function prototype.

Note that **every** include component must participate as the target in this relationship. This is required in order to:

- Preserve the original order of the include components within an include file. This order is established by the Populate tool and is used by the Generate tool.
- Support COBOL COPY statement and PL/I %INCLUDE statement within a data structure.

Source Information:

Class Name

DSIncludedSourceDef

Attribute

includeComponents

Target Information:

Class Name

DSIncludeComponent

Attribute

pSharedPointer

Forward Mapping Semantics:

Verb Contains

Cardinality

0..m

Ordered

Yes

Controlling

No

Inverse Mapping Semantics:

Verb IsMemberOf

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWHLL

Storage View File Name

ewshl28

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkCobolRenamesThru**Cache View Name**

DSLinkCobolRenamesThru

Storage View Name

DSLinkCobolRenamesThru

Description

An instance of this relationship represents the association between a COBOL level-66 data name and the data item specified in the COBOL RENAME THRU... clause.

Source Information:**Class Name**

DSCobolSpecialDataName

Attribute

endRenamedItem

Target Information:**Class Name**

DSDataItem

Attribute

endRenamingItems

Forward Mapping Semantics:**Verb** EndRenames**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsEndRenamedBy**Cardinality**

0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWHLL

Storage View File Name
ewshl23

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkArrayDef2DataDef

Cache View Name
DSLinkArrayDef2DataDef

Storage View Name
DSLinkArrayDef2DataDef

Description
An instance of this relationship represents the association between an array and its element type.

Source Information:

Class Name
DSArrayDef

Attribute
pDataDefinition

Target Information:

Class Name
DSDataDefinition

Attribute
pArrayDef

Forward Mapping Semantics:

Verb HasElement

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsElementOf

Cardinality
0..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWHLL

Storage View File Name
ewshl10

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkArrayDef2SDD

Cache View Name
DSLinkArrayDef2SDD

Storage View Name
DSLinkArrayDef2SDD

Description
An instance of this relationship represents the association between an array and its element type.

Source Information:

Class Name
DSArrayDef

Attribute
pShareableElement

Target Information:

Class Name
DSSharedDataDefinition

Attribute
pArrayDef

Forward Mapping Semantics:

Verb HasElement

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsElementOf

Cardinality
0..1

Ordered
No

Controlling
No

Cache View File Name
EWSWHLL

Storage View File Name
ewshl15

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkArrDef2TDRRef

Cache View Name
DSLinkArrDef2TDRRef

Storage View Name
DSLinkArrDef2TDRRef

Description
An instance of this relationship represents the association between an array and its element type, which is a shareable named data definition. mutually exclusive.

Source Information:

Class Name
DSArrayDef

Attribute
pTypeDefReference

Target Information:

Class Name
DSTypeDefReference

Attribute
arrayDefs

Forward Mapping Semantics:

Verb HasElement

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsElementOf

Cardinality
0..m

Ordered
No

Controlling
Yes

Cache View File Name
EWSWHLL

Storage View File Name
ewshl35

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkArrayDefKey

Cache View Name
DSLinkArrayDefKey

Storage View Name
DSLinkArrayDefKey

Description
An instance of this relationship represents the connection between an array and the key data item specified in the COBOL OCCURS...KEY IS... clause.

Source Information:

Class Name
DSArrayDef

Attribute
keys

Target Information:

Class Name
DSDataItem

Attribute
pHasKeyRelSource

Forward Mapping Semantics:

Verb HasKey

Cardinality
0..12

Ordered
Yes

Controlling
No

Inverse Mapping Semantics:

Verb IsKeyOf

Cardinality
0..1

Ordered
No

Controlling
No

Cache View File Name
EWSWHLL

Storage View File Name
ewshl11

DLL Name
atlmod

Attributes

Name	Description
------	-------------

isAscending	
--------------------	--

	This attribute specifies a flag indicating whether the key is ascending or descending. This attribute represents the ASCENDING or DESCENDING keyword in the above mentioned COBOL clause.
--	---

Data Type	boolean
------------------	---------

Valid Values	TRUE, FALSE
---------------------	-------------

Default	TRUE
----------------	------

Cache View Methods

None

Relationship Class: DSLinkCobolLevel88

Cache View Name

DSLinkCobolLevel88

Storage View Name

DSLinkCobolLevel88

Description

An instance of this relationship represents the association between a COBOL conditional variable and a condition name.

Source Information:**Class Name**

DSDatalItem

Attribute

level88Items

Target Information:**Class Name**

DSCobolSpecialDataName

Attribute

conditionVariable

Forward Mapping Semantics:

Verb HasLevel88

Cardinality

0..m

Ordered

Yes

Controlling

No

Inverse Mapping Semantics:

Verb IsLevel88Of

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWHLL

Storage View File Name

ewshl21

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkUnionDefMember

Cache View Name

DSLinkUnionDefMember

Storage View Name

DSLinkUnionDefMember

Description

An instance of this relationship represents the association between a ***union*** definition and the data item that is directly contained within the union. If the data item is a substructure; its data components are not the target of this relationship.

Source Information:**Class Name**

DSUnionDef

Attribute

dataMembers

Target Information:**Class Name**

DSDataItem

Attribute

pUnion

Forward Mapping Semantics:

Verb HasMember

Cardinality

0..m

Ordered

Yes

Controlling

No

Inverse Mapping Semantics:

Verb IsMemberOf

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWHLL

Storage View File Name

ewshl15

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkFunctionParameterType

Cache View Name
DSLinkFunctionParameterType

Storage View Name
DSLinkFunctionParameterType

Description
An instance of this relationship represents the association between a function (or procedure) declaration and its parameter type.

Source Information:

Class Name
DSFunctionDeclaration

Attribute
parameterType

Target Information:

Class Name
DSDataDefinition

Attribute
pParmTypeFunction

Forward Mapping Semantics:

Verb HasParameter

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsParmTypeOf

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWHLL

Storage View File Name
ewshl30

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkOffsetArea

Cache View Name
DSLLinkOffsetArea

Storage View Name
DSLLinkOffsetArea

Description

An instance of this relationship represents the association between a PL/I OFFSET definition and the containing AREA variable. be of type PLIArea.

Source Information:

Class Name
DSOffset

Attribute
pPLIAreaVariable

Target Information:

Class Name
DSDataItem

Attribute
areaOffsets

Forward Mapping Semantics:

Verb InArea

Cardinality
0..1

Ordered
No

Controlling
Yes

Inverse Mapping Semantics:

Verb AreaContains

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWHLL

Storage View File Name
ewshl26

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkArrayDefLBVariable

Cache View Name
DSLinkArrayDefLBVariable

Storage View Name
DSLinkArrayDefLBVariable

Description
An instance of this relationship represents the association between a variable length array definition and the data item that defines the actual size of the lower bound of the array.

Source Information:

Class Name
DSArrayDef

Attribute
lowerBoundVariable

Target Information:

Class Name
DSDataItem

Attribute
pLBRefersToRelSource

Forward Mapping Semantics:

Verb LBRefersTo

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsLBOf

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWHLL

Storage View File Name
ewshl12

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkPtrDef2DataDef

Cache View Name
DSLinkPtrDef2DataDef

Storage View Name
DSLinkPtrDef2DataDef

Description

An instance of this relationship represents the association between a pointer definition and the type pointed to by this pointer. are mutually exclusive.

Source Information:

Class Name
DSPointerDef

Attribute
pDataDefinition

Target Information:

Class Name
DSDataDefinition

Attribute
pPointerDef

Forward Mapping Semantics:

Verb PointsTo

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** isTypeOf**Cardinality**

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWHLL

Storage View File Name

ewshl18

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkPtrDef2TDRRef

Cache View Name

DSLinkPtrDef2TDRRef

Storage View Name

DSLinkPtrDef2TDRRef

Description

An instance of this relationship represents the association between a pointer definition and the data type pointed to by this pointer. mutually exclusive.

Source Information:**Class Name**

DSPointerDef

Attribute

pTypeDefReference

Target Information:**Class Name**

DSTypeDefReference

Attribute
pointerDefs

Forward Mapping Semantics:

Verb PointsTo

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsTypeOf

Cardinality
0..m

Ordered
No

Controlling
Yes

Cache View File Name
EWSWHLL

Storage View File Name
ewshl36

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkInclDStm2InclDSrc

Cache View Name
DSLinkInclDStm2InclDSrc

Storage View Name
DSLinkInclDStm2InclDSrc

Description
An instance of this relationship represents the association between an include statement and the referenced include file.

Source Information:

Class Name
DSIncludeStatement

Attribute
referencedInclSource

Target Information:

Class Name
DSIncludedSourceDef

Attribute
includeStatements

Forward Mapping Semantics:

Verb References

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ReferencedBy

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWHLL

Storage View File Name
ewshl29

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataItm2CbIDef

Cache View Name
DSLinkDataItm2CbIDef

Storage View Name
DSLinkDataItm2CbIDef

Description

An instance of this relationship represents the association between a data item and its refined definition in COBOL.

Source Information:**Class Name**

DSDatalItem

Attribute

cobolDefinition

Target Information:**Class Name**

DSCobolDataDefinition

Attribute

refinedDataItem

Forward Mapping Semantics:

Verb RefinedBy

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Refines

Cardinality

1..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWHLL

Storage View File Name

ewshl24

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkDataItn2PLIDtDfn

Cache View Name

DSLinkDataItn2PLIDtDfn

Storage View Name

DSLinkDataItn2PLIDtDfn

Description

An instance of this relationship represents the association between a data item and its refined definition in PL/I.

Source Information:**Class Name**

DSDataItem

Attribute

pliDefinition

Target Information:**Class Name**

DSPLIDataDefinition

Attribute

refinedDataItem

Forward Mapping Semantics:

Verb RefinedBy

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Refines

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWHLL

Storage View File Name

ewshl25

DLL Name

atItnmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkFnctnDcl2PLIEntry

Cache View Name

DSLinkFnctnDcl2PLIEntry

Storage View Name

DSLinkFnctnDcl2PLIEntry

Description

An instance of this relationship represents the association between an function definition its refinement in PL/I

Source Information:**Class Name**

DSFunctionDeclaration

Attribute

pliEntryData

Target Information:**Class Name**

DSPLIEntryDataDCL

Attribute

pFunctionDcl

Forward Mapping Semantics:

Verb RefinedBy

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Refines

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWHLL

Storage View File Name

ewshl32

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkFunctionReturnType

Cache View Name

DSLinkFunctionReturnType

Storage View Name

DSLinkFunctionReturnType

Description

An instance of this relationship represents the association between a function (or procedure) definition and its returned data type.

Source Information:

Class Name

DSFunctionDeclaration

Attribute

returnedType

Target Information:

Class Name

DSDataDefinition

Attribute

pReturnTypeFunction

Forward Mapping Semantics:

Verb Returns

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb IsReturnTypeOf

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWHLL

Storage View File Name

ewshl31

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkFnctnDcl2CFnPrtp**Cache View Name**

DSLinkFnctnDcl2CFnPrtp

Storage View Name

DSLinkFnctnDcl2CFnPrtp

Description

An instance of this relationship represents the association between an function definition its refinement in C.

Source Information:**Class Name**

DSFunctionDeclaration

Attribute

cFnPrototype

Target Information:**Class Name**

DSCFunctionPrototype

Attribute

pFunctionDcl

Forward Mapping Semantics:**Verb** RefinedBy**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** Refines**Cardinality**

1..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWHLL

Storage View File Name

ewshl33

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkAreaSizeVariable**Cache View Name**

DSLinkAreaSizeVariable

Storage View Name

DSLinkAreaSizeVariable

Description

An instance of this relationship represents the association between a PL/I AREA size and the data item that defines the actual size of the area.

Source Information:**Class Name**

DSPLIArea

Attribute

sizeVariable

Target Information:**Class Name**

DSDataItem

Attribute

pArea

Forward Mapping Semantics:**Verb** SizeRefersTo**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** DefinesSizeOf

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWHLL

Storage View File Name
ewshl19

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkCobolRenames

Cache View Name
DSLinkCobolRenames

Storage View Name
DSLinkCobolRenames

Description
An instance of this relationship represents the association between a COBOL level-66 data name and the data item that is specified after the RENAMES clause.

Source Information:

Class Name
DSCobolSpecialDataName

Attribute
startRenamedItem

Target Information:

Class Name
DSDataItem

Attribute
startRenamingItems

Forward Mapping Semantics:

Verb StartRenames

Cardinality
0..1

Ordered
No

Controlling

Yes

Inverse Mapping Semantics:**Verb** IsStartRenamedBy**Cardinality**
0..m**Ordered**
No**Controlling**
No**Cache View File Name**

EWSWHLL

Storage View File Name

ewshl22

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkTDRRef2SDDAlias2**Cache View Name**

DSLinkTDRRef2SDDAlias2

Storage View Name

DSLinkTDRRef2SDDAlias2

Description

An instance of this relationship represents the association between a TypeDefReference and its C/C++ name.

Source Information:**Class Name**

DSTypeDefReference

Attribute

cName

Target Information:**Class Name**

DSSDDAlias

Attribute

TDRNames

Forward Mapping Semantics:**Verb** HasCNameIn

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsCNameFor**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWHLL

Storage View File Name

ewshl38

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkTDRRef2SDDAlias1**Cache View Name**

DSLinkTDRRef2SDDAlias1

Storage View Name

DSLinkTDRRef2SDDAlias1

Description

An instance of this relationship represents the association between a TypeDefReference and its PL/I name.

Source Information:**Class Name**

DSTypeDefReference

Attribute

pliName

Target Information:**Class Name**

DSSDDAlias

Attribute

TDRPLINames

Forward Mapping Semantics:**Verb** HasPLINameln**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** IsPLINameFor**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWHLL

Storage View File Name

ewshl37

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkTDRRef2SDD

Cache View Name

DSLinkTDRRef2SDD

Storage View Name

DSLinkTDRRef2SDD

Description

An instance of this relationship represents the association between a user defined type in an include file and the shareable data definition.

Source Information:**Class Name**

DSTypeDefReference

Attribute
pSharedDataDef

Target Information:

Class Name
DSSharedDataDefinition

Attribute
typeDefReferences

Forward Mapping Semantics:

Verb RefersTo

Cardinality
0..1

Ordered
No

Controlling
Yes

Inverse Mapping Semantics:

Verb UsedBy

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWHLL

Storage View File Name
ewshl39

DLL Name
atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkArrayDefUBVariable

Cache View Name
DSLinkArrayDefUBVariable

Storage View Name
DSLinkArrayDefUBVariable

Description

An instance of this relationship represents the association between a variable length array definition and the data item that defines the actual size of the upper bound of the array.

Source Information:**Class Name**

DArrayDef

Attribute

upperBoundVariable

Target Information:**Class Name**

DSDatItem

Attribute

pUBRefersToRelSource

Forward Mapping Semantics:

Verb UBRefersTo

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb IsUBOf

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWLL

Storage View File Name

ewshl13

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Relationship Class: DSLinkCobolLevel66DataItm

Cache View Name

DSLinkCobolLevel66DataItm

Storage View Name

DSLinkCobolLevel66DataItm

Description

An instance of this relationship represents the association between a COBOL data item (level 01) and a level 66 data name.

Source Information:**Class Name**

DSDatalItem

Attribute

level66DataItems

Target Information:**Class Name**

DSCobolSpecialDataName

Attribute

level66DataItem

Forward Mapping Semantics:

Verb HasLevel66Name

Cardinality

0..m

Ordered

Yes

Controlling

No

Inverse Mapping Semantics:

Verb IsLevel66Of

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWHLL

Storage View File Name

ewshl44

DLL Name

atlmod

Attributes

None

Cache View Methods

None

Chapter 9. Relational Database: Common Submodel

Object Class: DSRNamedRelation

Cache View Name

DSRNamedRelation

Storage View Name

DSRNamedRelation

Inherits From

DSTechnologyObject

Description

An instance of this object class represents an SQL Table, View, or an Alternate name (Alias or Synonym). This class is a subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=NamdRItn

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD1

Storage View File Name

ewsr25

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

createdTS	
------------------	--

This attribute represents the time when the CREATE statement was executed.

Data Type	
------------------	--

char*

Default	
----------------	--

null

pContainedInRC	
-----------------------	--

This attribute specifies an instance of the relationship NamdRItn_ContainedInRC_RCollctn.

Data Type	
------------------	--

relationship

pHasAlternateName

This attribute specifies an instance of the relationship
NamdRltn_HasAlternateName_AltrntNm.

Data Type

relationship

primaryCreator

This attribute represents the primary authorization ID of the user who
created the table, view, alias, or synonym.

Data Type

char*

Default

null

registeredFlag

This attribute indicates whether the SQL name is under the control of a
registration facility for the relational database system. A registration facility
controls who can issue DDL for the named construct. Y - indicates SQL
name is controlled by the registration facility. N - Indicates SQL name is not
controlled by the registration facility.

Data Type

char

Valid Values

'Y', 'N'

Default

null

Cache View Methods

Name	Description
DSRNamedRelationInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRAlternateName

Cache View Name

DSRAlternateName

Storage View Name

DSRAlternateName

Inherits From

DSRNamedRelation

Description

An instance of this object class represents an alternate name for Table or
View. In DB2 for MVS, an alternate name is an Alias or a Synonym. In
SQL/DS, an alternate name is a Synonym.

The abbreviated name of this class (used in relationship classes)=AltrntNm

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD1

Storage View File Name
ewsr00

DLL Name
ewsrdb

Attributes

Name Description

comment

This is a character string that a user may associate with an alternate name to describe its purpose, etc. For DB2 for MVS, this attribute is meaningful for an alias and ignored for a synonym.

Data Type

char*

Valid Values

a character string

Default

null

label

This is a character string that a user may associate with an alternate name for the presentation purposes. For DB2 for MVS, this attribute is meaningful for an alias and ignored for a synonym.

Data Type

char*

Valid Values

a character string

Default

null

pInPhysicalDesign

This attribute specifies an instance of the relationship AltrntNm_In_PhysDsgn.

Data Type

relationship

pInToDoList

This attribute specifies an instance of the relationship AltrntNm_In_ToDoList.

Data Type

relationship

pSynonymOrAliasFor

This attribute specifies an instance of the relationship AltrntNm_SynonymOrAliasFor_NamdRltn.

Data Type

relationship

synonymAliasFlag

This attribute indicates whether the alternate name is Synonym or an Alias.

- A - Alias
- S - Synonym

Data Type

char

Valid Values

'A', 'S'

Default*null*

Cache View Methods

Name Description**DSRAlternateNameInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRCheckConstraint

Cache View Name

DSRCheckConstraint

Storage View Name

DSRCheckConstraint

Inherits From

DSRNamedRelation

Description

An instance of this object class represents the check constraint for the ANSI SQL table and ANSI SQL column.

The abbreviated name of this class (used in relationship classes)=ChckCnst

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD1

Storage View File Name

ewsr03

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

pCConstraintCDef	
-------------------------	--

This attribute specifies an instance of the relationship
ChckCnst_CCConstraintCDef_ClmnDfnt.

Data Type	
------------------	--

relationship

pCConstraintTDef	
-------------------------	--

This attribute specifies an instance of the relationship
ChckCnst_CCConstraintTDef_TblDfntn.

Data Type	
------------------	--

relationship

searchCondition	
------------------------	--

This attribute defines a search condition of a check constraint

Data Type	
------------------	--

char*

Valid Values	
---------------------	--

a character string

Default	
----------------	--

null

constraintName	
-----------------------	--

This attribute defines the locally unique name of the check constraint.

Data Type	
------------------	--

char*

Default	
----------------	--

null

comment	
----------------	--

This attribute holds the ON COMMENT text for a constraint

Data Type	
------------------	--

char*

Default	
----------------	--

null

Cache View Methods

Name	Description
------	-------------

DSRCheckConstraintInit	
-------------------------------	--

This instance method initializes the attributes of this class.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

None

Object Class: DSRColumnDefinition

Cache View Name

DSRColumnDefinition

Storage View Name

DSRColumnDefinition

Inherits From

DSTechnologyObject

Description

An instance of this object class represents a column for relational Tables. A column uses some information from a Data Element.

The abbreviated name of this class (used in relationship classes)=ClmnDfnt

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWRD1

Storage View File Name

ewsr05

DLL Name

ewsrdb

Attributes

Name Description**columnName**

This attribute represents the name of a column. This attribute must not be a SQL reserved word and must be a valid SQL identifier. The length of this attribute must be 18 characters or less for DB2, DB2/2, and DB2/6000.

Data Type

char*

Default*null***comment**

This attribute defines AN SQL comment on the column. This is a text string. For DB2 for MVS, validation should limit the length to 254.

Data Type

char*

Valid Values

a character string

Default*null***defaultOption**

This attribute indicates that the definition of the column DEFAULT includes DEFAULT keyword. Acceptable values are:

- Y - initialize with system DEFAULT
- I - initialize with value from initialValue attribute
- N - null

- C - initialize with CURRENT SQLID
- U - initialize with USER id
- D - initialize with CURRENT DATE
- T - initialize with CURRENT TIME
- S - initialize with CURRENT TIMESTAMP

Defaults requiring specification of a constant value will use the initialValue attribute. Defaults requiring a cast function name will use the castFunctionName attribute in conjunction with this attribute.

Data Type

char

Valid Values

- 'Y'
- 'I'
- 'N'
- 'C'
- 'U'
- 'D'
- 'T'
- 'S'

Default

null

initialValue

This attribute describes the initial value for a column. The initialValue string is used if the defaultOption attribute is set to 'I'.

Data Type

char*

Default

null

label This attribute defines an SQL label for the column text. This is a string that may be associated with a column to serve as an alternative name for presentation purposes. For DB2 for MVS, validation should limit the length to 30.

Data Type

char*

Valid Values

a character string

Default

null

nullsFlag

This attribute indicates the attribute that follows the DEFAULT option.

- Y - NULL
- N - not NULL

Data Type

char

Valid Values

'Y', 'N'

Default

null

pCDefForPRel

This attribute specifies an instance of the relationship ClmnDfnt_pCDefForPRel_PrsstRel.

Data Type

relationship

pColumnNameAlias

This attribute specifies an instance of the relationship ClmnDfnt_ColumnNameAlias_DEAlias.

Data Type

relationship

pColumnsForKD

This attribute specifies an instance of the relationship ClmnDfnt_ColumnsForKD_KeyDfntn.

Data Type

relationship

pColumnsForPRD

This attribute specifies an instance of the relationship ClmnDfnt_ColumnsForPRD_PrsstnRD.

Data Type

relationship

pColumnOfMRDTP

This attribute specifies an instance of the relationship DSRClmnDfnt_ColumnOfMRDTP_MRDTP.

Data Type

relationship

pHasCConstraint

This attribute specifies an instance of the relationship ClmnDfnt_HasCConstraint_CheckConstraint.

Data Type

relationship

pHasColumnType

This attribute specifies an instance of the relationship ClmnDfnt_HasColumnType_DataElmn.

Data Type

relationship

pHasUConstraint

This attribute specifies an instance of the relationship ClmnDfnt_HasUConstraint_UniqueConstraint.

Data Type

relationship

pLocalType

This attribute specifies an instance of the relationship ClmnDfnt_LocalType_SmplType.

Data Type
relationship

pReferencedByCD

This attribute specifies an instance of the relationship
ClmnDfnt_ReferencedByCD_ClmnDfnt.

Data Type
relationship

pReferencesCD

This attribute specifies an instance of the relationship
ClmnDfnt_ReferencesCD_ClmnDfnt.

Data Type
relationship

pRefinedByMCDE

This attribute specifies an instance of the relationship
ClmnDfnt_RefinedByMCDE_MRDClmnD.

Data Type
relationship

pRefinedByWCDE

This attribute specifies an instance of the relationship
ClmnDfnt_RefinedByWCDE_WRDClmnD.

Data Type
relationship

pBasedOnAttribute;

This attribute specifies an instance of the relationship
ClmnDfnt_BasedOn_cdmAttributeToColumnMapping This attribute locates
the attribute to column mapping object that is linked to that column
definition. The attribute to column mapping object contains information
about the attribute that is the source for the column definition

Data Type
relationship

longName

This attribute represents the conceptual name of a column definition. This
attribute need not be a valid SQL identifier. It is the name of a column in a
relational design. columnName holds the SQL name of column used in
generated DDL.

Data Type
char*

Valid Values
a character string

Default
null

castFunctionName

This attribute contains the user-specified name of the cast function to be
used to set the DEFAULT values for a column

Data Type
char*

Valid Values
a character string

Default
null

Cache View Methods

Name **Description**

DSRColumnDefinitionInit

This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

reconcileObject

This method reconciles the collection of DSRColumnDefinitions.

Return Type
void

Parameters

Name **<Direction>Type**

****ppRecObjectColl**
<Input > DSCollection

***pParmList**
<Input > DSRDBParmList

Object Class: DSRColumnType

Cache View Name
DSRColumnType

Storage View Name
DSRColumnType

Inherits From
DSTechnologyObject

Description

An instance of this class represents the RDB data attributes that are used in the definition of a column. This class has no relationships with any other RDB class. It is used to define the return value of the getColumnTypes method of the ColumnDefinition class.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWRD1

Storage View File Name

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

bitFlag	
----------------	--

This attribute specifies an enumerated value indicating the sybtype of the column.

Data Type

enumColumnSubType

precision	
------------------	--

This attribute specifies an integer representing the total number of bits or digits of a binary or decimal numeric data.

This number is used to derive the size of a numeric data type.

Data Type

unsigned short

Valid Values

non-negative integer

Default

0

scale	This attribute specifies the position of the binary point or decimal point.
--------------	---

Data Type

unsigned short

Default

0

Update Constraints

(document only)

scale < precision

strLength	
------------------	--

This attribute specifies an integer representing the number of bytes specified in the definition of a character or graphic string data type. For graphic string, the actual size occupied in storage is twice the strLength.

Data Type

unsigned long

Valid Values

non negative integer

Default

0

typeName	
-----------------	--

This attribute specifies an enumerated value indicating the type of the column.

Data Type

enumColumnType

Cache View Methods

Name	Description
DSRColumnTypeInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRDatabase

Cache View Name	DSRDatabase
Storage View Name	DSRDatabase
Inherits From	DSTechnologyObject
Description	<p>This is a base class for the Database objects in Database2 for MVS, Database2 for OS/2 and Database2 for AIX. A subclass of this class is to instantiated for the database object. This class is a subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.</p> <p>The abbreviated name of this class (used in relationship classes)=Database</p>
Public Class	Yes
Extends	None
Class Extent	None
Cache View File Name	EWSWRD1
Storage View File Name	ewsr07
DLL Name	ewsrdb

Attributes

Name	Description
comment	<p>This attribute represents a description for a Database. This attribute is ignored for Database2 for MVS.</p>
Data Type	char*

Valid Values

a character string

Default

null

pContainedInRDBS

This attribute specifies an instance of the relationship Database_ContainedInRDBS_RDBSystem.

Data Type

relationship

Cache View Methods

Name	Description
------	-------------

DSRDatabaseInit	
------------------------	--

This instance method initializes the attributes of this class.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

None

Object Class: DSRIndex

Cache View Name

DSRIndex

Storage View Name

DSRIndex

Inherits From

DSTechnologyObject

Description

An instance of this object class represents an SQL Index. This class is a subclass of TechnologyObject to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=Index

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD1

Storage View File Name

ewsr08

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

clusterRatio	
---------------------	--

This attribute identifies the percentage of rows that are in clustering order.

Data Type	
------------------	--

long

Default	
----------------	--

-1

creator	
----------------	--

This attribute identifies the primary authorization ID of the user who created the index.

Data Type	
------------------	--

char*

Valid Values	
---------------------	--

a character string

Default	
----------------	--

null

firstKeyCard	
---------------------	--

This attribute identifies the number of distinct values of the first key column.

Data Type	
------------------	--

long

Default	
----------------	--

-1

fullKeyCard	
--------------------	--

This attribute identifies the number of distinct values of the key.

Data Type	
------------------	--

long

Default	
----------------	--

-1

numLeafs	
-----------------	--

This attribute identifies the number of active leaf pages in the index.

Data Type	
------------------	--

long

Default	
----------------	--

-1

numLevels	
------------------	--

This attribute identifies the number of levels in the index tree.

Data Type	
------------------	--

long

Default	
----------------	--

-1

pContainedInRC	
-----------------------	--

This attribute specifies an instance of the relationship
Index_ContainedInRC_RCollctn.

Data Type
relationship

pUsesKey
This attribute specifies an instance of the relationship
Index_UsesKey_KeyDfntn.

Data Type
relationship

uniqueFlag
This attribute defines the uniqueness flag for Index.

- Y - Unique
- N - Not unique

Data Type
char

Valid Values
'Y', 'N'

Default
null

Cache View Methods

Name	Description
------	-------------

DSRIndexInit	This instance method initializes the attributes of this class.
---------------------	--

Return Type	void
--------------------	------

Parameters	None
-------------------	------

Object Class: DSRKeyDefinition

Cache View Name
DSRKeyDefinition

Storage View Name
DSRKeyDefinition

Inherits From
DSTechnologyObject

Description
An instance of this object class represents an ordered set of columns from a table. This set of columns may correspond to a SQL key, a SQL index, or both.

In DB2 and DB2/2, there are two kinds of keys for a table, namely, primary key and foreign key. In ANSI SQL, it is also possible to have a key that is neither a foreign or a primary key (this happens when a set of columns that is not a primary key is referenced by a foreign key of another table.) A primary key must be unique. A table can have only one primary key and that is assured by one-to-one relation IsPrimaryKey between KeyDefinition and TableDefinition. A table can have many foreign keys. A Referential Constraint on a table is indicated by the relationship RConstrains from one instance of KeyDefinition to another. (The source of RConstrains is the

KeyDefinition that represents the primary key. The target is the KeyDefinition that represents the foreign key.)

This class is a subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=KeyDfntn

Constraints:

- If a KeyDefinition object is the target of a HasKey relation, the columns that belong to the key must also belong to the TableDefinition that is the source of the HasKey relation.
- If a KeyDefinition object is the target of a RConstrains relation, each ColumnDefinition associated with this KeyDefinition must **reference** the **corresponding** ColumnDefinition associated with the KeyDefinition that is the source of the RConstrains relation.

Definitions:

- "Column A **referenes** column B" means that there is a chain of zero or more ReferencesCD relations such that column A is the head of the chain and column B is the tail of the chain.
- "Column A **corresponds** to column B" means that if column A is the target of the nth instance of the set of KeyColumns for a given KeyDefinition, column B must be the nth instance of the set of KeyColumns associated with another KeyDefinition.

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWRD1

Storage View File Name

ewsr09

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

plsAKey	This attribute specifies an instance of the relationship KeyDfntn_IsAKey_TblDfntn.
----------------	--

Data Type	relationship
------------------	--------------

plsPrimaryKey	This attribute specifies an instance of the relationship KeyDfntn_IsPrimaryKey_TblDfntn.
----------------------	--

Data Type	relationship
------------------	--------------

pKeyColumns

This attribute specifies an instance of the relationship KeyDfntn_KeyColumns_ClmnDfnt.

Data Type

relationship

pRConstrainedBy

This attribute specifies an instance of the relationship KeyDfntn_RConstrainedBy_KeyDfntn.

Data Type

relationship

pRConstrains

This attribute specifies an instance of the relationship KeyDfntn_RConstrains_KeyDfntn.

Data Type

relationship

pUsedByIndex

This attribute specifies an instance of the relationship KeyDfntn_UsedByIndex_Index.

Data Type

relationship

pBasedOnRelationship

This attribute specifies an instance of the relationship Index_BasedOn_cdmRelshipToFKeyMapping This attribute locates the relationship to key mapping object that is linked to that key definition. The relationship to key mapping object contains information about the relationship that is the source for the key definition.

Data Type

relationship

Cache View Methods

Name	Description
DSRKeyDefinitionInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRPersistentRelation

Cache View Name

DSRPersistentRelation

Storage View Name

DSRPersistentRelation

Inherits From

DSRNamedRelation

Description

An instance of a subtype of this object class represents an SQL relation (table and/or view). This class is kept for future extension of model, i.e., if model has to provide some common attributes or relationships for Table or View that can be defined on this class.

The abbreviated name of this class (used in relationship classes)=PrsstntR

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD1

Storage View File Name

ewsr30

DLL Name

ewsrdb

Attributes

Name Description**pPRelHasCDef**

This attribute specifies an instance of the relationship PrsstRel_pPRelHasCDef_ClmnDfnt.

Data Type

relationship

Cache View Methods

Name Description**DSRPersistentRelationInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRPersistentRelDef

Cache View Name

DSRPersistentRelDef

Storage View Name

DSRPersistentRelDef

Inherits From

DSTechnologyObject

Description

An instance of a subtype of this object class represents the definition of any number of persistent relations (tables and/or views). This class is a

subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=PrsstnRD

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD1

Storage View File Name

ewsr31

DLL Name

ewsrdb

Attributes

Name	Description
comment	This attribute represents an SQL comment on the persistent relation.
Data Type	char*
Valid Values	a character string
Default	<i>null</i>
label	This attribute represents an SQL label for a persistent relation, serving as an alternate name.
Data Type	char*
Valid Values	a character string
Default	<i>null</i>
pCollectedBy	This attribute specifies an instance of the relationship PrsstnRD_CollectedInRelDsgn.
Data Type	relationship
pForVD	This attribute specifies an instance of the relationship PrsstnRD_ForVD_ViwDfntn.
Data Type	relationship

pHasColumns

This attribute specifies an instance of the relationship PrsstnRD_HasColumns_ClmnDfnt.

Data Type

relationship

Cache View Methods

Name Description**DSRPersistentRelDefInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRPhysicalDesign

Cache View Name

DSRPhysicalDesign

Storage View Name

DSRPhysicalDesign

Inherits From

DSTechnologyObject

Description

An instance of this object class represents the extensions to the RelationalDesign required to collect object needed for a Physical Design.

The abbreviated name of this class (used in relationship classes)=PhysDsgn

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWRD1

Storage View File Name

ewsr7d

DLL Name

ewsrdb

Attributes

Name Description**pInRDesign**

This attribute specifies an instance of the relationship PhysDsgn_In_RelDsgn.

Data Type
relationship

pHasAltNames
This attribute specifies an instance of the relationship
DSRPhysDsgn_Has_AltmntNm.

Data Type
relationship

pHasToDoList
This attribute specifies an instance of the relationship
DSRPhysDsgn_Has_ToDoList.

Data Type
relationship

Cache View Methods

Name	Description
DSRPhysicalDesignInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRRCollection

Cache View Name
DSRRCollection

Storage View Name
DSRRCollection

Inherits From
DSTechnologyObject

Description
An instance of this object class represents an SQL collection, currently implemented in some relational database products as an authorization identifier. An SQL collection is used as a second part in the fully qualified SQL names. This class is a subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=RCollctn

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD1

Storage View File Name

ewsr04

DLL Name

ewsrdb

Attributes

Name Description**pContainedInRDBS**

This attribute specifies an instance of the relationship RCollctn_ContainedInRDBS_RDBSystem.

Data Type

relationship

pHasIndex

This attribute specifies an instance of the relationship RCollctn_HasIndex_Index.

Data Type

relationship

pHasNRelation

This attribute specifies an instance of the relationship RCollctn_HasNRelation_NamdRltn.

Data Type

relationship

pOwnsMRDDatabase

This attribute specifies an instance of the relationship RCollctn_OwnsMRDDatabase_MRDDatbs.

Data Type

relationship

pOwnsMRDStogroup

This attribute specifies an instance of the relationship RCollctn_OwnsMRDStogroup_MRDSrvg.

Data Type

relationship

pOwnsMRDTablespace

This attribute specifies an instance of the relationship RCollctn_OwnsMRDTablespace_MRDTblsp.

Data Type

relationship

Cache View Methods

Name Description**DSRRCollectionInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRRDBSystem

Cache View Name

DSRRDBSystem

Storage View Name

DSRRDBSystem

Inherits From

DSTechnologyObject

Description

An instance of this object class represents a Relational Database System with a name to facilitate distributed data representation. This class is a subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=RDBSystem

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWRD1

Storage View File Name

ewsr33

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

pHasBufferpool	
-----------------------	--

This attribute specifies an instance of the relationship RDBSystem_HasBufferpool_MRDBffrp

Data Type

relationship

pHasDatabase	
---------------------	--

This attribute specifies an instance of the relationship RDBSystem_HasDatabase_Database.

Data Type

relationship

pHasRCollection	
------------------------	--

This attribute specifies an instance of the relationship RDBSystem_HasRCollection_RCollctn. RCollctn is used for authorization and/or a middle qualifier in relation objects name.

Data Type

relationship

pHasStoragegroup

This attribute specifies an instance of the relationship RDBSystem_HasStoragegroup_MRDSrugg.

Data Type

relationship

pROSharedBy

This attribute specifies an instance of the relationship RDBSystem_ROSharedBy_MRDDatbs.

Data Type

relationship

rdbSystemType

This attribute defines a code for relational database product running as operational database system. The values are hexadecimal. Possible values are:

- 0 - Undefined
- 0x0101 - Release 3.2 of DB2/MVS
- 0x0102 - Release 4.1 of DB2/MVS
- 0x0103 - Release 5.1 of DB2/390
- 0x0201 - Release 2.1 of DB2/CS
- 0x0202 - Release 3.0 of DB2/UD
- 0x0301 - Release 7.3 of Oracle

Data Type

short

Default

0

rdbProductName

This attribute defines a full name of the database product installed on the operational database system, such as, IBM Database 2 2.3.

Data Type

char*

Valid Values

a character string or *null*

Default

null

rdbProductVersion

This attribute identifies the version/release of the database product installed on the operational database system, such as MVS_32, MVS_41, CS_30, Oracle_73.

Data Type

char*

Valid Values

a character string or *null*

Default

null

pOracleTS

This attribute specifies an instance of the relationship RDBSystem_has_OracleTbIspace

Data Type
relationship

sysplex

This attribute identifies this RDBSystem (unique catalog) as a data-sharing environment (sysplex)

Data Type
boolean

pORDTblspace

This attribute specifies an instance of the relationship RDBSystem_has_ORDTblspace.

Data Type
relationship

Cache View Methods

Name	Description
------	-------------

DSRRDBSystemInit	
-------------------------	--

	This instance method initializes the attributes of this class.
--	--

Return Type	void
--------------------	------

Parameters	None
-------------------	------

Object Class: DSRRelationalDesign

Cache View Name
DSRRelationalDesign

Storage View Name
DSRRelationalDesign

Inherits From
DSTechnologyObject

Description

An instance of this object class represents a group of persistent relations (tables and/or views) which have been designed together. A conceptual data model can be transformed into a relational design. The abbreviated name of this class = RelDsgn.

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD1

Storage View File Name
ewsr3w

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

pCollectsPrsstnRD	
--------------------------	--

This attribute specifies an instance of the relationship RelDsgn_Collects_PrsstnRD.

Data Type	
------------------	--

relationship

pHasPDesign	
--------------------	--

This attribute specifies an instance of the relationship RelDsgn_Has_PhysDsgn.

Data Type	
------------------	--

relationship

status	This attribute indicates whether the relational design has been refined for a specific data base management system (DBMS).
---------------	--

Data Type	
------------------	--

char

Valid Values	
---------------------	--

a character

Default	
----------------	--

null

pTargetOfTP	
--------------------	--

This attribute specifies an instance of the relationship RelDsgn_TargetOf_cdmTransformProcess This attribute locates the CDM transform process the relational design is target of.

Data Type	
------------------	--

relationship

Cache View Methods

Name	Description
------	-------------

DSRRelationalDesignInit	
--------------------------------	--

This instance method initializes the attributes of this class.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

None

Object Class: DSRTTable

Cache View Name	
------------------------	--

DSRTTable

Storage View Name	
--------------------------	--

DSRTTable

Inherits From	
----------------------	--

DSRPersistentRelation

Description	
--------------------	--

An instance of this object class represents an SQL Table.

The abbreviated name of this class (used in relationship classes)=Table

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD1

Storage View File Name

ewsr38

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

cardinality	This attribute represents the total number of rows in the table.
--------------------	--

Data Type	long
------------------	------

Default	-1
----------------	----

catalogDBID	This attribute represents an internal identifier of the database.
--------------------	---

Data Type	short
------------------	-------

Default	-1
----------------	----

catalogOBID	This attribute represents an internal identifier of the table.
--------------------	--

Data Type	short
------------------	-------

Default	-1
----------------	----

keyOBID	This attribute represents an internal identifier of the index that enforces uniqueness of the table's primary key.
----------------	--

Data Type	short
------------------	-------

Default	-1
----------------	----

nPages	This attribute represents the total number of pages on which rows of the table appear.
---------------	--

Data Type
long

Default
-1

pHasTDefinition
This attribute specifies an instance of the relationship
Table_HasTDefinition_TblDfntn.

Data Type
relationship

plsParentTable
This attribute specifies an instance of the relationship
Table_IsParentTable_TableMap.

Data Type
relationship

plsChildTable
This attribute specifies an instance of the relationship
Table_IsChildTable_TableMap.

Data Type
relationship

pBasedOnEntity
This attribute specifies an instance of the relationship
Table_BasedOn_cdmEntityToTableMapping This attribute locates the entity
to table mapping object that is linked to that table. The entity to table
mapping object contains information about the entity that is the source for
the table.

Data Type
relationship

Cache View Methods

Name	Description
------	-------------

DSRTableInit	This instance method initializes the attributes of this class.
---------------------	--

Return Type	void
--------------------	------

Parameters	None
-------------------	------

getReconcilableObjects	This method finds all the reconcilable objects (DSRColumnDefinitions) in a DSRTableDefinition and puts them into a collection.
-------------------------------	---

Return Type	void
--------------------	------

Parameters	
-------------------	--

Name	<Direction>	Type
-------------	--------------------------	-------------

**ppRecObjectColl		
	<Input >	DSCollection

***pParmList**
<Input > DSParmList

getTableInfo

This instance method collects the information about a table that is needed to generate a maintenance folder.

Return Type
void

Parameters

Name	<Direction>	Type
name	<Output>	IString
comments	<Output>	IString
createDTS	<Output>	IString
modDTS	<Output>	IString
owner	<Output>	IString
database	<Output>	IString

PopulateTable

This virtual instance method populates a DSRTTable from the catalog.

Return Type
int

Parameters
None

Object Class: DSRTTableDefinition

Cache View Name
DSRTTableDefinition

Storage View Name
DSRTTableDefinition

Inherits From
DSRPersistentRelDef

Description

An instance of this object class represents the definition for SQL table. The table definition can be shared by many SQL Tables.

The abbreviated name of this class (used in relationship classes)=TblDfntn

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD1

Storage View File Name
ewsr39

DLL Name
ewsrdb

Attributes

Name	Description
auditOption	This attribute defines the type of access to this table. N - NONE; no auditing to be performed when table is accessed. C - CHANGES; auditing to be performed upon the first INSERT, UPDATE, or DELETE operation within each unit of recovery. A - ALL; auditing to be performed upon the first access of any kind occurring within each unit of recovery.
Data Type	char
Valid Values	'N','C','A'
Default	<i>null</i>
pDefinesTable	This attribute specifies an instance of the relationship TblDfntn_DefinesTable_Table.
Data Type	relationship
pHasKey	This attribute specifies an instance of the relationship TblDfntn_HasKey_KeyDfntn.
Data Type	relationship
pHasPrimaryKey	This attribute specifies an instance of the relationship TblDfntn_HasPrimaryKey_KeyDfntn.
Data Type	relationship
pIncludesCConstraint	This attribute specifies an instance of the relationship TblDfntn_IncludesCConstraint_CheckConstraint.
Data Type	relationship
pIncludesUConstraint	This attribute specifies an instance of the relationship TblDfntn_IncludesUConstraint_UniqueConstraint.
Data Type	relationship

pBasedOnEntity

This attribute specifies an instance of the relationship
 TblDfntn_BasedOn_cdmEntityToTableMapping This attribute locates the
 entity to table mapping object that is linked to that table definition. The
 entity to table mapping object contains information about the entity that is
 the source for the table definition.

Data Type

relationship

Cache View Methods

Name Description**DSRTableDefinitionInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRTableMap

Cache View Name

DSRTableMap

Storage View Name

DSRTableMap

Inherits From

DSRPersistentRelDef

Description

An instance of this object class represents the association between a
 referential constraint and the tables which own the primary and foreign keys
 on which the constraint is defined.

This mapping is needed in order to disambiguate the primary key/foreign
 key relationship whenever a table definition is shared by more than one
 table.

The abbreviated name of this class (used in relationship classes)=UnqCnstr

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD1

Storage View File Name

ewsr5a

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

pMapsConstraint	
------------------------	--

This attribute specifies an instance of the relationship TableMap_MapsConstraint_RConstrs.

Data Type	
------------------	--

relationship

pParentTable	
---------------------	--

This attribute specifies an instance of the relationship TableMap_ParentTable_Table.

Data Type	
------------------	--

relationship

pChildTable	
--------------------	--

This attribute specifies an instance of the relationship TableMap_ChildTable_Table.

Data Type	
------------------	--

relationship

Cache View Methods

Name	Description
------	-------------

DSRTableMapInit	
------------------------	--

This instance method initializes the attributes of this class.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

None

Object Class: DSRUniqueConstraint

Cache View Name	
------------------------	--

DSRUniqueConstraint

Storage View Name	
--------------------------	--

DSRUniqueConstraint

Inherits From	
----------------------	--

DSRPersistentRelDef

Description	
--------------------	--

An instance of this object class represents the unique constraint on ANSI SQL and SQL/DS table and column.

The abbreviated name of this class (used in relationship classes)=UnqCnstr

Public Class	
---------------------	--

Yes

Extends	
----------------	--

None

Class Extent	
---------------------	--

None

Cache View File Name
EWSWRD1

Storage View File Name
ewsr40

DLL Name
ewsrdb

Attributes

Name	Description
pUConstraintCDef	This attribute specifies an instance of the relationship UnqCnstr_UConstraintCDef_ClmnDfnt.
Data Type	relationship
pUConstraintTDef	This attribute specifies an instance of the relationship UnqCnstr_UConstraintTDef_TblDfntn.
Data Type	relationship

Cache View Methods

Name	Description
DSRUniqueConstraintInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRView

Cache View Name
DSRView

Storage View Name
DSRView

Inherits From
DSRPersistentRelation

Description
An instance of this object class represents an SQL View.
The abbreviated name of this class (used in relationship classes)=View

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWRD1

Storage View File Name
ewsr41

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

pHasVDefinition	
------------------------	--

This attribute specifies an instance of the relationship View_HasVDefinition_ViWdfntr.

Data Type	
------------------	--

relationship

select	This attribute represents the actual SQL DDL of the select statement using actual Table or View names.
---------------	--

Data Type	
------------------	--

char*

Valid Values	
---------------------	--

a character string

Default	
----------------	--

null

Cache View Methods

Name	Description
------	-------------

DSRViewInit	
--------------------	--

This instance method initializes the attributes of this class.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

None

Object Class: DSRViewDefinition

Cache View Name
DSRViewDefinition

Storage View Name
DSRViewDefinition

Inherits From
DSRPersistentRelDef

Description

An instance of this object class represents the definition of any number of SQL Views.

The abbreviated name of this class (used in relationship classes)=ViWdfntr

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD1

Storage View File Name
ewsr42

DLL Name
ewsrdb

Attributes

Name	Description
checkOption	This attribute indicates whether the DBMS should check the results of updates to this view against the view's search condition. Valid values are: <ul style="list-style-type: none"> • Y - use local check • C - use CASCADED check • N - no check <p>Data Type char</p> <p>Valid Values <ul style="list-style-type: none"> • 'Y' • 'N' • 'C' • <i>null</i> <p>Default <i>null</i></p> </p>
pForView	This attribute specifies an instance of the relationship ViwDfntn_ForView_View. <p>Data Type relationship</p>
pResolvesInPRD	This attribute specifies an instance of the relationship ViwDfntn_ResolvesInPRD_PrsstnRD. <p>Data Type relationship</p>
select	This attribute defines an abstraction of actual SQL DDL of the select, using the technology independent names of underlying persistent relation definition. <p>Data Type char*</p> <p>Valid Values a character string</p>

checkOption

This attribute indicates whether the DBMS should check the results of updates to this view against the view's search condition. Valid values are:

- Y - use local check
- C - use CASCADED check
- N - no check

Data Type
char

Valid Values

- 'Y'
- 'N'
- 'C'
- *null*

Default
null

pForView

This attribute specifies an instance of the relationship ViwDfntn_ForView_View.

Data Type
relationship

pResolvesInPRD

This attribute specifies an instance of the relationship ViwDfntn_ResolvesInPRD_PrsstnRD.

Data Type
relationship

select This attribute defines an abstraction of actual SQL DDL of the select, using the technology independent names of underlying persistent relation definition.

Data Type
char*

Valid Values
a character string

Default
null

Cache View Methods

Name	Description
------	-------------

DSRViewDefinitionInit	
------------------------------	--

This instance method initializes the attributes of this class.

Return Type	void
--------------------	------

Parameters	None
-------------------	------

Relationship Class: DSRLinkPRelHasCDef

Cache View Name	DSRLinkPRelHasCDef
------------------------	--------------------

Storage View Name	DSRLinkPRelHasCDef
--------------------------	--------------------

Description	An instance of this relationship represents PersistentRelation and ColumnDefinition.
--------------------	--

Source Information:

Class Name	DSRPersistentRelation
-------------------	-----------------------

Attribute	pPreHasCDef
------------------	-------------

Target Information:

Class Name	DSRColumnDefinition
-------------------	---------------------

Attribute	pCDefForPRel
------------------	--------------

Forward Mapping Semantics:

Verb	PRelHasCDef
-------------	-------------

Cardinality	0..m
--------------------	------

Ordered	No
----------------	----

Controlling	No
--------------------	----

Inverse Mapping Semantics:

Verb	CDefForPRel
-------------	-------------

Cardinality	0..m
--------------------	------

Ordered	No
----------------	----

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr7a

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

colCard

The number of distinct values in the column.

Data Type
long

Default
-1

high2Key

This attribute represents second highest value of the column.

Data Type
char*

Valid Values
a character string

Default
null

low2Key

This attribute represents second lowest value of the column.

Data Type
char*

Valid Values
a character string

Default
null

Cache View Methods

Name	Description
------	-------------

DSRLinkCDefForPREInit

This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Relationship Class: DSRLinkPRelDefInRDesign

Cache View Name

DSRLinkPRelDefInRDesign

Storage View Name

DSRLinkPRelDefInRDesign

Description

An instance of this relationship represents DSRRelationalDesign and DSRPersistentRelDef

Source Information:**Class Name**

DSRPersistentRelDef

Attribute

pCollectedBy

Target Information:**Class Name**

DSRRelationalDesign

Attribute

pCollectsPrsstnRD

Forward Mapping Semantics:

Verb CollectedIn

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Collects

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD1

Storage View File Name

ewsr3x

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRColumnNameAlias

Cache View Name

DSRColumnNameAlias

Storage View Name

DSRColumnNameAlias

Description

An instance of this relationship represents DSRColumnDefinition and DSDataElementAlias.

Source Information:**Class Name**

DSRColumnDefinition

Attribute

pCollectedBy

Target Information:**Class Name**

DSDataElementAlias

Attribute

pForColumnDef

Forward Mapping Semantics:

Verb ColumnNameAlias

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb ForColumnDef

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD1

Storage View File Name

ewsr3i

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkHasCConstraint

Cache View Name

DSRLinkHasCConstraint

Storage View Name

DSRLinkHasCConstraint

Description

An instance of this relationship represents DSRColumnDefinition and DSRCheckConstraint.

Source Information:

Class Name

DSRColumnDefinition

Attribute

pCollectedBy

Target Information:

Class Name

DSRCheckConstraint

Attribute

pCConstraintCDef

Forward Mapping Semantics:

Verb HasCConstraint

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb CConstraintCDef

Cardinality

0..1

Ordered

No

Controlling

No

Cache View File Name

EWSWRD1

Storage View File Name

ewsr62

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkHasColumns**Cache View Name**

DSRLinkHasColumns

Storage View Name

DSRLinkHasColumns

Description

An instance of this relationship represents DSRPersistentRelDef and DSRColumnDefinition.

Source Information:**Class Name**

DSRPersistentRelDef

Attribute

pCollectedBy

Target Information:**Class Name**

DSRColumnDefinition

Attribute

pColumnsForPRD

Forward Mapping Semantics:**Verb** HasColumns**Cardinality**

0..m

Ordered

Yes

Controlling

No

Inverse Mapping Semantics:**Verb** ColumnsForPRD**Cardinality**

0..1

Ordered

No

Controlling
Yes

Cache View File Name
EWSWRD1

Storage View File Name
ewsr2a

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkHasColumnType

Cache View Name
DSRLinkHasColumnType

Storage View Name
DSRLinkHasColumnType

Description
An instance of this relationship represents DSRColumnDefinition and DataElement.

Source Information:

Class Name
DSRColumnDefinition

Attribute
pCollectedBy

Target Information:

Class Name
DSDataElement

Attribute
pForColumnDef

Forward Mapping Semantics:

Verb HasColumnType

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ForColumnDef

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr2b

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkDBaseInRDBSys

Cache View Name
DSRLinkDBaseInRDBSys

Storage View Name
DSRLinkDBaseInRDBSys

Description
An instance of this relationship represents DSRRDBSystem and DSRDatabase.

Source Information:

Class Name
DSRDatabase

Attribute
pCollectedBy

Target Information:

Class Name
DSRRDBSystem

Attribute
pHasDatabase

Forward Mapping Semantics:

Verb ContainedInRDBSys

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasDatabase

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr2g

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkIndexInRCollection

Cache View Name
DSRLinkIndexInRCollection

Storage View Name
DSRLinkIndexInRCollection

Description
An instance of this relationship represents RCollection and DSRIIndex.

Source Information:

Class Name
DSRIIndex

Attribute
pCollectedBy

Target Information:

Class Name
DSRRCollection

Attribute
pHasIndex

Forward Mapping Semantics:

Verb ContainedInRColl

Cardinality

1..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** HasIndex**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD1

Storage View File Name

ewsr67

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkHasKey

Cache View Name

DSRLinkHasKey

Storage View Name

DSRLinkHasKey

Description

An instance of this relationship represents DSRTABLEDefinition and DSRKeyDefinition. than the table containing its columns.

Source Information:**Class Name**

DSRTABLEDefinition

Attribute

pCollectedBy

Target Information:**Class Name**

DSRKeyDefinition

Attribute
plsAKey

Forward Mapping Semantics:

Verb HasKey

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsAKey

Cardinality
0..1

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr69

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

foreignKeyFlag

This attribute is a flag to indicate whether the target KeyDfntn represents one or more foreign keys of the source TbIDfntn.

Data Type
boolean

Default
false

primaryKeyFlag

This attribute is a flag to indicate whether the target KeyDfntn represents the primary key of the source TbIDfntn.

Data Type
boolean

Default
false

uniqueFlag

This attribute is a flag to indicate the uniqueness that ensures the key columns in a table are unique.

- Y - Unique

- N - Not Unique

Data Type
char

Valid Values
'Y', 'N'

Default
null

Update Constraints

(document only)

If the target KeyDfntn is the source of a IsPrimaryKey relation and the source TblDfntn is the target of the same IsPrimaryKey relation, uniqueFlag must be 'Y'.

constraintName

This attribute represents the unique constraint name, i.e, represents primary or unique key of a Table.

Data Type
char*

comment

This attribute holds the ON COMMENT text for a unique constraint name.

Data Type
char*

Cache View Methods

Name Description

DSRLinkHasKeyInit

This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Relationship Class: DSRLinkNRelationInRColl

Cache View Name

DSRLinkNRelationInRColl

Storage View Name

DSRLinkNRelationInRColl

Description

An instance of this relationship represents RCollection and DSRNamedRelation.

Source Information:

Class Name
DSRNamedRelation

Attribute
pCollectedBy

Target Information:

Class Name
DSRRCollection

Attribute
pHasNRelation

Forward Mapping Semantics:

Verb ContainedInRColl

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasNRelation

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr70

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkRCollectionInRDBSys

Cache View Name
DSRLinkRCollectionInRDBSys

Storage View Name
DSRLinkRCollectionInRDBSys

Description
An instance of this relationship represents DSRRDBSystem and RCollection.

Source Information:

Class Name
DSRRCollection

Attribute
pCollectedBy

Target Information:

Class Name
DSRRDBSystem

Attribute
pHasRCollection

Forward Mapping Semantics:

Verb ContainedInRDBSys

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasRCollection

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr64

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkMapsConstraints

Cache View Name
DSRLinkMapsConstraints

Storage View Name
DSRLinkMapsConstraints

Description

An instance of this relationship represents TableMap and RConstrainedBy.

This relation is used to associate a RConstrainedBy relationship with one or more TableMap objects. The TableMap objects are used to explicitly identify the pair(s) of tables that contain the primary and foreign keys. (There will be more than one such pair of tables whenever a TableDefinition object is shared by more than one Table object.)

Source Information:**Class Name**

DSRTableMap

Attribute

pMapsConstraint

Target Information:**Class Name**

DSRLinkRConstrainedBy

Attribute

pHasTableMap

Forward Mapping Semantics:

Verb MapsConstraint

Cardinality

1..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb HasTableMap

Cardinality

0..n

Ordered

No

Controlling

No

Cache View File Name

EWSWRD1

Storage View File Name

ewsr5b

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkHasTDefinition

Cache View Name

DSRLinkHasTDefinition

Storage View Name

DSRLinkHasTDefinition

Description

An instance of this relationship represents DSRTTable and DSRTTableDefinition.

Source Information:**Class Name**

DSRTTable

Attribute

pMapsConstraint

Target Information:**Class Name**

DSRTTableDefinition

Attribute

pDefinesTable

Forward Mapping Semantics:

Verb HasTDefinition

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb DefinesTable

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD1

Storage View File Name

ewsr76

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkHasUConstraint

Cache View Name

DSRLinkHasUConstraint

Storage View Name

DSRLinkHasUConstraint

Description

An instance of this relationship represents DSRColumnDefinition and DSRLinkHasUConstraint.

Source Information:

Class Name

DSRColumnDefinition

Attribute

pMapsConstraint

Target Information:

Class Name

DSRLinkHasUConstraint

Attribute

pUConstraintCDef

Forward Mapping Semantics:

Verb HasUConstraint

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb UConstraintCDef

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWRD1

Storage View File Name

ewsr78

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkHasVDefinition**Cache View Name**

DSRLinkHasVDefinition

Storage View Name

DSRLinkHasVDefinition

Description

An instance of this relationship represents DSRView and DSRViewDefinition that can be used by many views.

Source Information:**Class Name**

DSRView

Attribute

pMapsConstraint

Target Information:**Class Name**

DSRViewDefinition

Attribute

pForView

Forward Mapping Semantics:**Verb** HasVDefinition**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** ForView**Cardinality**

0..m

Ordered

No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr79

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkIncludesCConstraint

Cache View Name
DSRLinkIncludesCConstraint

Storage View Name
DSRLinkIncludesCConstraint

Description
An instance of this relationship represents DSRTableDefinition and DSRCheckConstraint.

Source Information:

Class Name
DSRTableDefinition

Attribute
pMapsConstraint

Target Information:

Class Name
DSRCheckConstraint

Attribute
pCConstraintTDef

Forward Mapping Semantics:

Verb IncludesCConstraint

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb CConstraintTDef

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWRD1

Storage View File Name

ewsr84

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkIncludesUConstraint**Cache View Name**

DSRLinkIncludesUConstraint

Storage View Name

DSRLinkIncludesUConstraint

Description

An instance of this relationship represents DSRTableDefinition and DSRTUniqueConstraint.

Source Information:**Class Name**

DSRTTableDefinition

Attribute

pMapsConstraint

Target Information:**Class Name**

DSRTUniqueConstraint

Attribute

pUConstraintTDef

Forward Mapping Semantics:**Verb** IncludesUConstraint**Cardinality**

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** UConstraintTDef**Cardinality**

0..1

Ordered

No

Controlling

No

Cache View File Name

EWSWRD1

Storage View File Name

ewsr87

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkIsPrimaryKey**Cache View Name**

DSRLinkIsPrimaryKey

Storage View Name

DSRLinkIsPrimaryKey

Description

An instance of this relationship represents DSRKeyDefinition and DSRTableDefinition.

Source Information:**Class Name**

DSRKeyDefinition

Attribute

pMapsConstraint

Target Information:**Class Name**

DSRTableDefinition

Attribute

pHasPrimaryKey

Forward Mapping Semantics:**Verb** IsPrimaryKey

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** HasPrimaryKey**Cardinality**

0..1

Ordered

No

Controlling

No

Cache View File Name

EWSWRD1

Storage View File Name

ewsr89

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkKeyColumns**Cache View Name**

DSRLinkKeyColumns

Storage View Name

DSRLinkKeyColumns

Description

An instance of this relationship represents DSRKeyDefinition and DSRColumnDefinition.

Source Information:**Class Name**

DSRKeyDefinition

Attribute

pMapsConstraint

Target Information:**Class Name**

DSRColumnDefinition

Attribute
pColumnsForKD

Forward Mapping Semantics:

Verb KeyColumns

Cardinality
0..m

Ordered
Yes

Controlling
No

Inverse Mapping Semantics:

Verb KeyColumns

Cardinality
0..m

Ordered
Yes

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr2h

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

ascDescFlag

This attribute specifies the ascending or descending characteristics of a column used by the index on a table.

- A - ASC
- D - DESC

Data Type
char

Valid Values

- "A"
- "D"
- *null*

Default
null

Cache View Methods

Name	Description
------	-------------

DSRLinkKeyColumnsInit

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Relationship Class: DSRLocalType**Cache View Name**

DSRLocalType

Storage View Name

DSRLocalType

Description

An instance of this relationship represents DSRColumnDefinition and SmpIType.

Source Information:**Class Name**

DSRColumnDefinition

Attribute

pMapsConstraint

Target Information:**Class Name**

DSbkm:(compact) ipf:(compact)TypeDef

Attribute

pForLocalColDef

Forward Mapping Semantics:

Verb LocalType

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb ForLocalColDef

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWRD1

Storage View File Name

ewsr3k

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkAltNameInPDesign**Cache View Name**

DSRLinkAltNameInPDesign

Storage View Name

DSRLinkAltNameInPDesign

Description

An instance of this relationship represents PhysicalDesign and AlternateName.

Source Information:**Class Name**

DSRAAlternateName

Attribute

pInPhysicalDesign

Target Information:**Class Name**

DSRPhysicalDesign

Attribute

pHasAltNames

Forward Mapping Semantics:**Verb** In**Cardinality**

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** Has**Cardinality**

0..m

Ordered

No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr7g

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkRConstrainedBy

Cache View Name
DSRLinkRConstrainedBy

Storage View Name
DSRLinkRConstrainedBy

Description

An instance of this relationship represents DSRKeyDefinition and DSRKeyDefinition.

This relation is many-to-many in order to support sharing of key definitions within a table. For example, if the same set of columns is specified in more than one FOREIGN KEY clause, these keys may be represented by a single instance of the KeyDefinition class.

The abbreviated name of this class (used in relationship classes) = RConstrs

Source Information:

Class Name
DSRKeyDefinition

Attribute
pInPhysicalDesign

Target Information:

Class Name
DSRKeyDefinition

Attribute
pRConstrains

Forward Mapping Semantics:

Verb RConstrainedBy

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb RConstrains

Cardinality
0..n

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr96

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

constraintName	
-----------------------	--

This attribute represents the referential constraint name, i.e, foreign key of a Table.

Data Type
char*

Valid Values
a character string or *null*

Default
null

Update Constraints
(document only)
The name of a referential constraint must be unique within a table definition.

deleteOption	
---------------------	--

This attribute represents the delete rule for a referential constraint.

Data Type
char

Valid Values

- C - CASCADE
- R - RESTRICT
- N - SET NULL
- A - NO ACTION

Default
null

pHasTableMap

This attribute specifies an instance of the relationship RConstrs_HasTableMap_TableMap.

Data Type

relationship

reLOBID1

This attribute represents the internal identifier of the constraint with respect to the database that contains the parent table.

Data Type

short

Default

-1

reLOBID2

This attribute represents the internal identifier of the constraint with respect to the database that contains the dependent table.

Data Type

short

Default

-1

timeStamp

This attribute represents the date and time the constraint was defined.

Data Type

char*

Default

null

updatesFlag

This attribute indicates whether the column can be updated.

Data Type

char

Valid Values

- Y - YES
- R - RESTRICT

Default

null

comment

This attribute holds the ON COMMENT text for a referential constraint.

Data Type

char*

Default

null

Cache View Methods

Name Description**DSRLinkRConstrainsInit**

This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Relationship Class: DSRLinkPDesignInRDesign

Cache View Name
DSRLinkPDesignInRDesign

Storage View Name
DSRLinkPDesignInRDesign

Description
An instance of this relationship represents RelationalDesign and PhysicalDesign.

Source Information:

Class Name
DSRPhysicalDesign

Attribute
pInRDesign

Target Information:

Class Name
DSRRelationalDesign

Attribute
pHasPDesign

Forward Mapping Semantics:

Verb In

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr7f

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkReferencesCD

Cache View Name
DSRLinkReferencesCD

Storage View Name
DSRLinkReferencesCD

Description
An instance of this relationship represents DSRColumnDefinition and DSRColumnDefinition.

Source Information:

Class Name
DSRColumnDefinition

Attribute
pInRDesign

Target Information:

Class Name
DSRColumnDefinition

Attribute
pReferencedByCD

Forward Mapping Semantics:

Verb ReferencesColDef

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ReferencedByColDef

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr4a

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkResolvesInPRD

Cache View Name
DSRLinkResolvesInPRD

Storage View Name
DSRLinkResolvesInPRD

Description
An instance of this relationship represents DSRViewDefinition and DSRPersistentRelDef that are referenced in the DSRViewDefinition.

Source Information:

Class Name
DSRViewDefinition

Attribute
pInRDesign

Target Information:

Class Name
DSRPersistentRelDef

Attribute
pForVD

Forward Mapping Semantics:

Verb ResolvesInPRD

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ForVD

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr0m

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkParentTable

Cache View Name
DSRLinkParentTable

Storage View Name
DSRLinkParentTable

Description
An instance of this relationship represents TableMap and Table.
This relation is used to associate a TableMap object with the Table object that contains the primary key that is part of a referential constraint.

Source Information:

Class Name
DSRTableMap

Attribute
pParentTable

Target Information:

Class Name
DSRTable

Attribute
plsParentTable

Forward Mapping Semantics:

Verb ParentTable

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsParentTable

Cardinality
0..n

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsr5c

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkSynonymOrAliasFor

Cache View Name
DSRLinkSynonymOrAliasFor

Storage View Name
DSRLinkSynonymOrAliasFor

Description
An instance of this relationship represents DSRAAlternateName and DSRNamedRelation on which a synonym or an alias is defined.

Update Constraints
(document only)
An alias or synonym can not have an alias.
A synonym can not have a synonym.

Source Information:
Class Name
DSRAAlternateName
Attribute
pSynonymOrAliasFor

Target Information:
Class Name
DSRNamedRelation

Attribute
pHasAlternateName

Forward Mapping Semantics:

Verb IsSynonymOrAliasFor

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasAlternateName

Cardinality
0..n

Ordered
No

Controlling
No

Cache View File Name
EWSWRD1

Storage View File Name
ewsrOp

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship: DSRLinksChildTable

Description

An instance of this relationship represents Table and TableMap.

This relation is used to associate a TableMap object with the Table object that contains the foreign key that is part of a referential constraint.

Source Information:

Class Name
DSRTable

Attribute
plsChildTable

Target Information:

Class Name
DSRTableMap

Attribute
pChildTable

Forward Mapping Semantics:

Verb IsChildTable

Cardinality
0..n

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ChildTable

Cardinality
1..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWRD1

Storage View File Name
ewsrOp

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkIsChildTable

Cache View Name
DSRLinkIsChildTable

Storage View Name
DSRLinkIsChildTable

Description

An instance of this relationship represents Table and TableMap.

This relation is used to associate a TableMap object with the Table object that contains the foreign key that is part of a referential constraint.

Source Information:

Class Name
DSRTable

Attribute
plsChildTable

Target Information:

Class Name
DSRTableMap

Attribute
pChildTable

Forward Mapping Semantics:

Verb IsChildTable

Cardinality
0..n

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ChildTable

Cardinality
1..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWRD1

Storage View File Name
ewsr5d

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkUsesKey

Cache View Name
DSRLinkUsesKey

Storage View Name
DSRLinkUsesKey

Description

An instance of this relationship represents DSRIndex and DSRKeyDefinition.

Source Information:**Class Name**

DSRIndex

Attribute

plsChildTable

Target Information:**Class Name**

DSRKeyDefinition

Attribute

pUsedByIndex

Forward Mapping Semantics:

Verb UsesKey

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb UsedByIndex

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD1

Storage View File Name

ewsrOu

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Chapter 10. Relational Database: MVS Submodel

Object Class: DSRMRDBufferpool

Cache View Name

DSRMRDBufferpool

Storage View Name

DSRMRDBufferpool

Inherits From

DSTechnologyObject

Description

An instance of this class represents the bufferpool used by Tablespace, Database, Tablespace Partition, and Indexspace Partition under Database2 for MVS. This class is a subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=MRDBffrp

Constraint - The product code for an instance of this class related to the RDBSystem class must be MRD.

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWRD2

Storage View File Name

ewsr10

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

bufferPoolNo	This attribute represents the Bufferpool number used.
---------------------	---

Data Type

char*

Valid Values

- "BP0"
- "BP1"
- "BP2" ... "BP49"
- "BP32K"
- "BP32K1"
- "BP32K2", ... "BP32K9"

Default
"BP0"

pContainedInRDBS

This attribute specifies an instance of the relationship
MRDBffrp_ContainedInRDBS_RDBSystem.

Data Type
relationship

pInPhysicalDesign

This attribute specifies an instance of the
relationshipDSRMRDBffrp_In_MRDPDsgn

Data Type
relationship

pInToDoList

This attribute specifies an instance of the
relationshipDSRMRDBffrp_In_ToDoList

Data Type
relationship

size This attribute specifies the size of the bufferpool in number of pages.

Data Type
long

type This attribute specifies the bufferpool type. A - 4K page size, B - 32K page size.

Data Type
char

usageIntent

This attribute specifies the usage intent of the bufferpool. "Tablespace" -
used for tablespaces "Index" - used for indexes
<database_name.tablespace_name> - specific tablespace

Data Type
char*

pReferredByMIS

This attribute specifies an instance of the relationship
MRDBffrp_ReferredByMIS_MRDIIndex.

Data Type
relationship

Update Constraints

(document only)

Only 4K buffer pools (i.e. **bufferPoolNo** value is one of
"BP0", "BP1", "BP2", ... "BP49") may be used by an Index.

pReferredByMTS

This attribute specifies an instance of the relationship
MRDBffrp_ReferredByMTS_MRDTblsp.

Data Type
relationship

pUsedByMDB

This attribute specifies an instance of the relationship
MRDBffrp_UsedByMDB_MRDDatbs.

Data Type

relationship

Cache View Methods**Name Description****DSRMRDBufferpoolInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRMRDColumnDefExt**Cache View Name**

DSRMRDColumnDefExt

Storage View Name

DSRMRDColumnDefExt

Inherits From

DSTechnologyObject

Description

An instance of this object class represents the extensions in the
columnDefinition required for Database2 for MVS column definition.

The abbreviated name of this class (used in relationship
classes)=MRDCImnD

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD2

Storage View File Name

ewsr11

DLL Name

ewsrdb

Attributes**Name Description****constants**

List of constants needed for fieldProc

Data Type
char*

fieldProc
Name of field proc for this column.

Data Type
char*

pRefinesCD
This attribute specifies an instance of the relationship
MRDCImnD_RefinesCD_CImnDfnt.

Data Type
relationship

arithmeticFunctionsFrequency
This attribute specifies how frequently SQL statements are executed in
which the column is used with the following arithmetic functions: COUNT,
AVG, SUM, MAX, MIN.

Data Type
short

Valid Values
0 to 10

Default
-1

joinFrequency
This attribute specifies how frequently SQL statements are executed in
which the column is used for joins.

Data Type
short

Valid Values
0 to 10

Default
-1

rangePredicatesFrequency
This attribute specifies how frequently SQL statements are executed in
which the column is used with the following predicates: <, >, <=, >=, LIKE,
BETWEEN.

Data Type
short

Valid Values
0 to 10

Default
-1

skewingFactor
This attribute specifies the number of the most frequently occurring values.

Data Type
short

Default
-1

skewingPercentage

This attribute specifies the percentage of all rows in which the most frequently occurring values appear.

Data Type

short

Default

-1

sortingFunctionsFrequency

This attribute specifies how frequently SQL statements are executed in which the column is used with the following functions: GROUP BY, ORDER BY, DISTINCT.

Data Type

short

Valid Values

0 to 10

Default

-1

updateFrequency

This attribute specifies how frequently the column is updated.

Data Type

short

Valid Values

0 to 10

Default

-1

whereClauseFrequency

This attribute specifies how frequently SQL statements are executed in which the column is used in a WHERE clause.

Data Type

short

Valid Values

0 to 10

Default

-1

averageLength

This attribute specifies the average length of a VARCHAR or LONG VARCHAR column.

Data Type

short

Default

-1

initialNoOfDistinctValues

This attribute specifies the initial number of distinct values of the column.

Data Type

long

Default
-1

Cache View Methods

Name	Description
DSRMRDColumnDefExtInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRMRDDatabase

Cache View Name	DSRMRDDatabase
Storage View Name	DSRMRDDatabase
Inherits From	DSRDatabase
Description	<p>An instance of this class represents a Database2 for MVS Database.</p> <p>The abbreviated name of this class (used in relationship classes)=MRDDatbs</p> <p>Constraint - The product code for an instance of this class related to the RDBSystem class must be MRD.</p>
Public Class	Yes
Extends	None
Class Extent	Versioned
Cache View File Name	EWSWRD2
Storage View File Name	ewsr12
DLL Name	ewsrdb

Attributes

Name	Description
catalogDBID	This attribute represents internal identifier of the database in system catalog tables.
Data Type	short

Default

-1

creator

This attribute represents the value for original creator of database.

Data Type

char*

Default

null

pContainsMRDIndex

This attribute specifies an instance of the relationship MRDDatbs_ContainsMRDIndex_MRDIndex.

Data Type

relationship

pContainsMRDView

This attribute specifies an instance of the relationship MRDDatbs_ContainsMRDView_MRDView.

Data Type

relationship

pHasTablespace

This attribute specifies an instance of the relationship MRDDatbs_HasTablespace_MRDTblsp.

Data Type

relationship

pInPhysicalDesign

This attribute specifies an instance of the relationshipDSRMRDDatbs_In_MRDPDsgn

Data Type

relationship

pInToDoList

This attribute specifies an instance of the relationshipDSRMRDDatbs_In_ToDoList

Data Type

relationship

text

This attribute specifies a detailed description of the database.

Data Type

string<254>

Default

null

usageIntent

This attribute specifies the applications the database will be used for.

Q - QMF

N - others

Data Type

char

Valid Values

'Q', 'N'

Default
null

pOwnedByRC

This attribute specifies an instance of the relationship MRDDatbs_OwnedByRC_RCollctn.

Data Type
relationship

pROShare

This attribute specifies an instance of the relationship MRDDatbs_ROShare_RDBSystem.

Data Type
relationship

pUsedByMTB

This attribute specifies an instance of the relationship MRDDatbs_UsedByMTB_MRTable.

Data Type
relationship

pUsesBufferpool

This attribute specifies an instance of the relationship MRDDatbs_UsesBufferpool_MRDBffrp.

Data Type
relationship

pUsesStogroup

This attribute specifies an instance of the relationship MRDDatbs_UsesStogroup_MRStrgg.

Data Type
relationship

roShareFlag

This attribute indicates how the MRDDatabase will be using shared read-only data.

- O - OWNER
- R - READ ONLY

Data Type
char

Valid Values
'O', 'R'

Default
null

timeStamp

This attribute represents the time the database became shared on the owning system through shared read only data. A default value represents database being not shared.

Data Type
char*

Default
"0001-01-01-00.00.00.000000"

ccsid This attribute indicates whether or not the CCSID clause is to be generated for this database and the value to be set. If this attribute is null, no CCSID will be generated. Other values are 'E' - EBCDIC or 'A' - ASCII

Data Type
char

Cache View Methods

Name	Description
------	-------------

DSRMRDDatabaseInit	
---------------------------	--

This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Object Class: DSRMRDICFCatalog

Cache View Name
DSRMRDICFCatalog

Storage View Name
DSRMRDICFCatalog

Inherits From
DSTechnologyObject

Description
An instance of this class represents the ICF catalog used by Storage Group, Index and Tablespace in Database2 for MVS. This class is a subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=MRDICFCt

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD2

Storage View File Name
ewsr13

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

catAlias

This attribute represents an alias for ICF catalog name to be used for constructing DB2 data sets.

This attribute must not be a SQL reserve word and must be a valid SQL identifier.

Data Type

char*

Valid Values

a character string

Default

null

catName

This attribute represents the name of the ICF Catalog.

Data Type

char*

Valid Values

a valid SQL name string or *null*

Default

null

pForMP

This attribute specifies an instance of the relationship MRDICFCt_ForMP_MRDPrttn.

Data Type

relationship

pForSpace

This attribute specifies an instance of the relationship MRDICFCt_ForSpace_MRDSpace.

Data Type

relationship

pReferredByMSG

This attribute specifies an instance of the relationship MRDICFCt_ReferredByMSG_MRDSrugg.

Data Type

relationship

Cache View Methods

Name	Description
------	-------------

DSRMRDICFCatalogInit	
-----------------------------	--

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRMRDIndex

Cache View Name
DSRMRDIndex

Storage View Name
DSRMRDIndex

Inherits From
DSRIndex

Description
An instance of this class represents an Index in DB2.

The abbreviated name of this class (used in relationship classes)=MRDIndex

Constraint - The product code for an instance of this class related to the RDBSystem class via RCollection class must be MRD.

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD2

Storage View File Name
ewsr2r

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

catalogDBID	This attribute identifies the internal identifier of database for index.
--------------------	--

Data Type	short
------------------	-------

Default	-1
----------------	----

catalogISOBID	This attribute identifies the internal identifier for index page set descriptor.
----------------------	--

Data Type	short
------------------	-------

Default	-1
----------------	----

catalogOBID	This attribute identifies the internal identifier for index fan set descriptor.
--------------------	---

Data Type	short
------------------	-------

Default	-1
clusterFlag	This attribute identifies whether CLUSTER was specified when the index was created.
Data Type	char
Valid Values	'Y', 'N'
Default	<i>null</i>
pageSize	This attribute identifies size in bytes of the subpages in the index.
Data Type	short
Valid Values	256, 512, 1024, 2048, or 4096
Default	-1
pContainedInMRDD	This attribute specifies an instance of the relationship MRDIndex_ContainedInMRDD_MRDDatbs.
Data Type	relationship
pForMRDTable	This attribute specifies an instance of the relationship MRDIndex_ForMRDTable_MRDTable.
Data Type	relationship
pInPhysicalDesign	This attribute specifies an instance of the relationshipDSRMRDIndex_In_MRDPDsgn
Data Type	relationship
pInToDoList	This attribute specifies an instance of the relationshipDSRMRDIndex_In_ToDoList
Data Type	relationship
text	This attribute specifies a detailed description of the index.
Data Type	string<254>
Default	<i>null</i>

pMRDStoredIn

This attribute specifies an instance of the relationship
MRDIndex_MRDStoredIn_MRDIndxs.

Data Type

relationship

space This attribute identifies the number of kilobytes of DASD storage allocated to the index.

Data Type

long

Default

-1

statsTime

The data and time when the last invocation of RUNSTATS updated the statistics.

- The timestamp has a maximum of 26 digits and separators formatted as: yyyy-mm-dd-hh.mm.ss.nnnnnn..
- Where: yyyy - year, mm - month, dd - day, hh - hour, mm - minute, ss - second, nnnnnn - microseco.

Data Type

char*

Valid Values

timestamp or *null*

Default

null

whereNotNull

This attribute indicates whether or not the WHERE NOT NULL clause is to be generated for this Index.

Data Type

boolean

Valid Values

YES, NO

indexType

This attribute identifies the index as Type 1 or Type 2.

Data Type

char

Valid Values

1, 2, or *null*

Default

null

pieceSize

This attribute defines the value for the PIECESIZE clause for the index.

Data Type

long

Valid Values

non-negative integer

dataRcdsInIndex

This attribute specifies the number of records in a non-unique index.

Data Type

long

Valid Values

non-negative integer

distValsInIndex

This attribute specifies the number of distinct values in a non-unique index.

Data Type

long

Valid Values

non-negative integer

Cache View Methods

Name	Description
------	-------------

DSRIndexInit	
---------------------	--

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRMRDPartition

Cache View Name

DSRMRDPartition

Storage View Name

DSRMRDPartition

Inherits From

DSRIndex

Description

An instance of this class represents a partition for tablespace and index in Database2 for MVS. This is a superclass for MRDIndexPartition and MRDTSPartition.

The abbreviated name of this class (used in relationship classes)=MRDPrttn

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD2

Storage View File Name

ewsr16

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

cardinality	
--------------------	--

This attribute represents the number of rows referred to by index or table space or partition.

Data Type
long

Default
-1

farIndRef	
------------------	--

This attribute represents the number of rows that have been relocated far from their original page.

Data Type
long

Default
-1

freePage	
-----------------	--

This attribute represents the number of pages that are loaded before a page is left as free space.

Data Type
short

Valid Values
0 to 255

Default
-1

nearIndRef	
-------------------	--

This attribute represents the number of rows that have been relocated near their original page.

Data Type
long

Default
-1

partNo	
---------------	--

This attribute represents the partition number. If the tablespace or index is not partitioned, the partition number will be 0.

Data Type
short

Valid Values
0 to 64

Default
-1

Update Constraints

(document only)

Partition number must be less than or equal to the number of partitions.

pctFree

This attribute represents the percentage of each subpage or nonleaf page that is left as free space.

Data type

short

Visibility

protected

Valid values

0 to 99

Default value

-1

Constraints

percActive

This attribute represents the percentage of space occupied by rows of data from active tables.

Data Type

short

Default

-1

percDrop

This attribute represents the percentage of space occupied by rows of dropped tables.

Data Type

short

Default

-1

pRefersStogroup

This attribute specifies an instance of the relationship MRDPrtn_RefersStogroup_MRDCStrgg.

Data Type

relationship

pUsesCatalog

This attribute specifies an instance of the relationship MRDPrtn_UsesCatalog_MRDCICFct.

Data Type

relationship

space

This attribute identifies the number of kilobytes of DASD storage allocated to the indexspace or tablespace partition.

Data Type

long

Default

-1

statsTime

The data and time when the last invocation of RUNSTATS updated the statistics.

- The timestamp has a maximum of 26 digits and separators formatted as: yyyy-mm-dd-hh.mm.ss.nnnnnn..
- Where: yyyy - year, mm - month, dd - day, hh - hour, mm - minute, ss - second, nnnnnn - microseco.

Data Type

char*

Valid Values

timestamp or *null*

Default

null

gbpCache

This attribute indicates whether or not the GBPCACHE block is to be generated for this partition and the value to be set. If this attribute is null, no GBPCACHE will be generated. Other values are 'C' - CHANGED or 'A' - ALL

Data Type

char

Valid Values

'C', 'A'

Cache View Methods

Name	Description
DSRPartitionInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRMRDIndexPartition

Cache View Name

DSRMRDIndexPartition

Storage View Name

DSRMRDIndexPartition

Inherits From

DSRMRDPartition

Description

An instance of this class represents the Index Partition under Database2 for MVS.

The abbreviated name of this class (used in relationship classes)=MRDIdxP

Public Class

Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWRD2

Storage View File Name
ewsr14

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

farOffPos

This attribute represents number of referred to rows far from optimal position because of an insert into a full page.

Data Type
long

Default
-1

leafDist

This attribute represents 100 times the average number of pages between successive leaf pages of the index.

Data Type
long

Default
-1

nearOffPos

This attribute represents number of referred to rows near, but not at optimal position because of an insert into a full page.

Data Type
long

Default
-1

pForMIS

This attribute specifies an instance of the relationship MRDIndxP_ForMIS_MRDIndxs.

Data Type
relationship

values

This attribute represents a value for each partition. These values are concatenated, and the concatenation of all the values is the highest value of the key in the corresponding partition of the Index.

Data Type
char*

Valid Values

a valid SQL character string or *null*

Default

null

Cache View Methods

Name Description**DSRIndexPartitionInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRMRDSpace

Cache View Name

DSRMRDSpace

Storage View Name

DSRMRDSpace

Inherits From

DSTechnologyObject

Description

An instance of this object class is a base class for MRDIndexspace and MRDTablespace. This class is a subclass of TechnologyObject to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=MRDSpace

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD2

Storage View File Name

ewsr18

DLL Name

ewsrdb

Attributes

Name Description**closeRule**

This attribute defines the close rule option for an index or table space.

- Y - YES; data set are eligible for closing.
- N - NO; data set are not eligible for closing.

Data Type

char

Valid Values

'Y', 'N'

Default

null

dsetPassword

This attribute stores the password for the data sets of the index or table space.

Data Type

char*

Default

null

dsetPasswordFlag

This attribute indicates whether the data sets of index or table space are password protected.

- Y - data set are password protected
- N - data set are not password protected.

Data Type

char

Valid Values

'Y', 'N'

Default

null

freePage

This attribute represents the number of pages that are loaded before a page is left as free space.

Data Type

short

Valid Values

0 to 255

Default

-1

numParts

This attribute indicates the number of partitions.

Data Type

short

Valid Values

0 to 64

Default

-1

Update Constraints

(document only)

The number of partitions specified in the index space must be equal to the number of partitions specified in the table space.

pctFree

This attribute represents the percentage of each subpage or nonleaf page that is left as free space.

Data type

short

Visibility

protected

Valid values

0 to 99

Default value

-1

Constraints

pRefersStogroup

This attribute specifies an instance of the relationship MRDSpace_RefersStogroup_MRDStrgg.

Data type

relationship

Visibility

protected

Valid values

Default value

Constraints

The page set management of a space must be consistent, and if a space is in storage group it must not be in ICF catalog. (mutually exclusive)

pUsesCatalog

This attribute specifies an instance of the relationship MRDSpace_UsesCatalog_MRDICFCt.

Data Type

relationship

gbpCache

This attribute indicates whether or not the GBPCACHE block is to be generated for this indexspace or tablespace and the value to be set. If this attribute is null, no GBPCACHE will be generated. Other values are 'C' - CHANGED or 'A' - ALL. The data and time when the last invocation of RUNSTATS updated the statistics.

Data Type

char

Valid Values

'C', 'A'

Cache View Methods

Name	Description
------	-------------

DSRSpaceInit

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRMRDIndexspace**Cache View Name**

DSRMRDIndexspace

Storage View Name

DSRMRDIndexspace

Inherits From

DSRMRDSpace

Description

An instance of this object class represents the extension to the index required for the definition of Index in Database2 for MVS.

The abbreviated name of this class (used in relationship classes)=MRDIdxs

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWRD2

Storage View File Name

ewsr15

DLL Name

ewsrdb

Attributes**Name Description****deferRule**

This attribute indicates whether the index is built during execution of the CREATE INDEX statement.

- N - The index is built (this is the default.)
- Y - The index is not built.

Data Type

char

Valid Values

null, 'Y', 'N'

Default

null

pClusteredISForMTB

This attribute specifies an instance of the relationship
MRDIndxs_ClusteredISForMTB_MRDTable.

Data Type

relationship

pContainsPartition

This attribute specifies an instance of the relationship
MRDIndxs_ContainsPartition_MRDIndxP.

Data Type

relationship

pMRDStoresIndex

This attribute specifies an instance of the relationship
MRDIndxs_MRDStoresIndex_MRDIndex.

Data Type

relationship

pPartitionsTS

This attribute specifies an instance of the relationship
MRDIndxs_PartitionsTS_MRDTSPrt.

Data Type

relationship

pRefersBufferpool

This attribute specifies an instance of the relationship
MRDIndxs_RefersBufferpool_MRDBffrp.

Data Type

relationship

subPages

This attribute defines the number of subpages for each physical page.

Data Type

short

Valid Values

1, 2, 4, 8, 16

Default

-1

Cache View Methods

Name Description**DSRIndexspaceInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRMRDPhysicalDesign

Cache View Name

DSRMRDPhysicalDesign

Storage View Name

DSRMRDPhysicalDesign

Inherits From

DSRPhysicalDesign

Description

An instance of this object class represent a PhysicalDesign in Database2 for MVS.

The abbreviated name of this class (used in relationship classes)=MRDPDsgn

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWRD2

Storage View File Name

ewsr7e

DLL Name

ewsrdb

Attributes

Name Description**pHasTables**

This attribute specifies an instance of the relationship MRDPDsgn_Has_MRDTTable.

Data Type

relationship

pHasTablespaces

This attribute specifies an instance of the relationship MRDPDsgn_Has_MRDTblsp.

Data Type

relationship

pHasStogroup

This attribute specifies an instance of the relationship MRDPDsgn_Has_MRDCstrgg.

Data Type

relationship

pHasViews

This attribute specifies an instance of the relationship MRDPDsgn_Has_MRDCview.

Data Type

relationship

pHasIndexes

This attribute specifies an instance of the relationship MRDPDsgn_Has_MRDIIndex.

Data Type

relationship

pHasDatabases

This attribute specifies an instance of the relationship MRDPDsgn_Has_MRDDatbs.

Data Type

relationship

pHasBufferpools

This attribute specifies an instance of the relationship MRDPDsgn_Has_MRDBffrp.

Data Type

relationship

Cache View Methods

Name	Description
------	-------------

DSRMRDPhysicalDesignInit	
---------------------------------	--

	This instance method initializes the attributes of this class.
--	--

Return Type	
--------------------	--

	void
--	------

Parameters	
-------------------	--

	None
--	------

Object Class: DSRMRDStoragegroup

Cache View Name

DSRMRDStoragegroup

Storage View Name

DSRMRDStoragegroup

Inherits From

DSTechnologyObject

Description

An instance of this class represents a Storage Group in Database2 for MVS. This class is a subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=MRDStrgg

Constraint - The product code for an instance of this class related to the RDBSystem class must be MRD.

Public Class

Yes

Extends

None

Class Extent
Versioned

Cache View File Name
EWSWRD2

Storage View File Name
ewsr19

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

password

VSAM control or master level password.

Data Type

char*

Default

null

passwordFlag

This attribute indicates the password flag for the Storage Group. Y - storage group is password protected. N - storage group is not password protected.

Data Type

char

Valid Values

'Y', 'N'

Default

null

pContainedInRDBS

This attribute specifies an instance of the relationship MRDStrgg_ContainedInRDBS_RDBSystem.

Data Type

relationship

pForMP

This attribute specifies an instance of the relationship MRDStrgg_ForMP_MRDPrttn.

Data Type

relationship

pForSpace

This attribute specifies an instance of the relationship MRDStrgg_ForSpace_MRDSpace..

Data Type

relationship

pInPhysicalDesign

This attribute specifies an instance of the relationshipDSRMRDStrgg_In_MRDPDsgn

Data Type

relationship

plnToDoList

This attribute specifies an instance of the relationshipDSRMRDStrgg_In_ToDoList

Data Type

relationship

deviceType

This attribute specifies the device type of the volumes that are associated with the storagegroup.

Data Type

string<30>

Default

null

requiredSpace

This attribute specifies the space that has been calculated by the physical database design function.

Data Type

short

Default

-1

designStatus

This attribute indicates the status of the storagegroup design.

- W - in Work
- C - Completed
- I - Inconsistent

Data Type

char

Valid Values

'W', 'C', or 'I'

Default

null

text This attribute specifies a detailed description of the storagegroup.

Data Type

string<254>

Default

null

usageIntent

This attribute specifies if the storagegroup should be reserved for tablespaces or indexes.If no usage intent is specified, the first assignment of either a tablespace or an index defines the usage intent.This attribute indicates the status of the storagegroup design.

- I - Index
- T - Table

Data Type

char

Valid Values

'I', 'T'

Default
null

pOwnedByRC
This attribute specifies an instance of the relationship
MRDStrgg_OwnedByRC_RCollctn.

Data Type
relationship

pRefersCatalog
This attribute specifies an instance of the relationship
MRDStrgg_RefersCatalog_MRDICFCt.

Data Type
relationship

primaryCreator
This attribute represents the primary authorization ID of the user who
created the storage group.

Data Type
char*

Valid Values
a character string

Default
null

pUsedByMDB
This attribute specifies an instance of the relationship
MRDStrgg_UsedByMDB_MRDDatbs.

Data Type
relationship

pUsesVolume
This attribute specifies an instance of the relationship
MRDStrgg_UsesVolume_MRDVolum.

Data Type
relationship

Update Constraints
(document only)
All volumes in a storage group must be of same device
type.

space This attribute represents the number of kilobytes of DASD storage allocated
to the storage group as determined by the last execution of the STOSPACE
utility.

Data Type
long

Default
-1

spcDate
This attribute represents the date when the space column was last updated,
in the form yyddd.

Data Type
char*

Valid Values
a character string

Default
null

Cache View Methods

Name	Description
DSRStoragegroupInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRMRDTable

Cache View Name
DSRMRDTable

Storage View Name
DSRMRDTable

Inherits From
DSRTable

Description
An instance of this object class represent a Table in Database2 for MVS.

The abbreviated name of this class (used in relationship classes)=MRDTable

Constraint - The product code for an instance of this class related to the RDBSystem class via RCollection class must be MRD.

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD2

Storage View File Name
ewsr20

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

alterDTS

This attribute represents the time when the latest ALTER TABLE statement was applied.

Data Type

string

Default

null

checkFlag

This attribute indicates whether the table space containing the table is in check pending mode and there are rows in the table that can violate referential constraint.

Data Type

char

Valid Values

'C' or blank

Default

null

checkRID

This attribute represents the RID of the first row of the table that can violate referential constraint.

Data Type

string<4>

Default

null

dataCaptureFlag

This attribute specifies whether the logging of SQL INSERT, UPDATE, and DELETE operations on the table is augmented by additional information.

- N - None
- C - Changes

Data Type

char

Valid Values

'N', 'C'

Default

'N'

editProc

The name of the edit proc associated with this table.

Data Type

string

pctPages

This attribute represents the percentage of active table space pages that contain rows of the table.

Data Type

short

Default

-1

pctRowComp

This attribute represents the percentage of rows, multiplied by 100, compressed within the total number of active rows in the table. This includes any row transformed by an editproc such that its length is less than the length of the original row. -1 if the row describes a view or alias, or statistics have not been gathered.

Data Type

short

Default

-1

pHasClustering

This attribute specifies an instance of the relationship MRDTable_HasClustering_MRDIndex.

Data Type

relationship

Update Constraints

(document only)

only one clustering index per table.

pInPhysicalDesign

This attribute specifies an instance of the relationshipDSRMRDTable_In_MRDPDsgn

Data Type

relationship

pInToDoList

This attribute specifies an instance of the relationshipDSRMRDTable_In_ToDoList

Data Type

relationship

pMRDIndexedBy

This attribute specifies an instance of the relationship MRDTable_MRDIndexedBy_MRDIndex.

Data Type

relationship

pMRDHasPartitions

This attribute specifies an instance of the relationship MRDTable_Has_MRDTablePartition.

Data Type

relationship

pUsesDatabase

This attribute specifies an instance of the relationship MRDTable_UsesDatabase_MRDDatbs.

Data Type

relationship

pUsesTablespace

This attribute specifies an instance of the relationship MRDTable_UsesTablespace_MRDTblsp.

Data Type
relationship

recLength
This attribute represents the maximum length of any record in the table.

Data Type
short

Default
-1

statsTime
The data and time when the last invocation of RUNSTATS updated the statistics.

- The timestamp has a maximum of 26 digits and separators formatted as: yyyy-mm-dd-hh.mm.ss.nnnnnn..
- Where: yyyy - year, mm - month, dd - day, hh - hour, mm - minute, ss - second, nnnnnn - microseco.

Data Type
string<26>

Valid Values
timestamp or *null*

Default
null

status This attribute indicates the status of the table:

- I - table's definition is incomplete because it lacks a primary index
- X - table has a primary index
- blank - table has no primary key, or is a catalog table, or the row describes a view or alias.

Data Type
char

Valid Values
'I', 'X', or blank

Default
null

validProc
The name of the valid proc associated with this table.

Data Type
string

restrictDrop
This attribute indicates whether or not the RESTRICT ON DROP clause is to be generated for this table

Data Type
boolean

ccsid This attribute indicates whether or not the CCSID clause is to be generated for this table and the value to be set. If this attribute is null, no CCSID will be generated. Other values are 'E' - EBCDIC or 'A' - ASCII

Data Type
char

Valid Values
'E' or 'A'

randomUpdate
This attribute specifies whether the rows in the table will be updated randomly or sequentially.

Data Type
boolean

Cache View Methods

Name	Description
------	-------------

DSRTableInit	This instance method initializes the attributes of this class.
---------------------	--

Return Type	void
--------------------	------

Parameters	None
-------------------	------

Object Class: DSRMRDTablespace

Cache View Name
DSRMRDTablespace

Storage View Name
DSRMRDTablespace

Inherits From
DSRMRDSpace

Description
An instance of this object class represents the Tablespaces under Database2 for MVS.

The abbreviated name of this class (used in relationship classes)=MRDTblsp

Constraint - The product code for an instance of this class related to the RDBSystem class via RCollection class must be MRD.

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD2

Storage View File Name
ewsr22

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

compressFlag	
---------------------	--

This attribute indicates whether data compression applies to the rows of the table space.

- Y - Yes
- N - No

Data Type

char

Valid Values

'Y', 'N', *null*

Default

null

implicit	
-----------------	--

This attribute indicates whether the table space was created implicitly.

- Y - Yes
- N - No

Data Type

char

Valid Values

'Y', 'N', or blank

Default

null

lockSize	
-----------------	--

This attribute indicates locksize for the table space.

- A - ANY
- P - PAGE
- S - TABLESPACE
- T - TABLE

Data Type

char

Valid Values

'A', 'P', 'S', 'T'

Default

null

nActive	
----------------	--

This attribute indicates the number of active pages in the table space.

Data Type

long

Default

-1

nTables	
----------------	--

This attribute indicates the number of tables defined in the table space.

Data Type

short

Default

-1

pageSize

This attribute indicates the size of pages in the table space in kilobytes.

Data Type

short

Default

-1

pHasPartitions

This attribute specifies an instance of the relationship MRDTblsp_HasPartitions_MRDTSPrt.

Data Type

relationship

pInMRDDatabase

This attribute specifies an instance of the relationship MRDTblsp_InMRDDatabase_MRDDatbs.

Data Type

relationship

pInPhysicalDesign

This attribute specifies an instance of the relationshipDSRMRDTblsp_In_MRDPDsgn

Data Type

relationship

pInToDoList

This attribute specifies an instance of the relationshipDSRMRDTblsp_In_ToDoList

Data Type

relationship

pOwnedByRC

This attribute specifies an instance of the relationship MRDTblsp_OwnedByRC_RCollctn.

Data Type

relationship

pPartitionedByIX

This attribute specifies an instance of the relationship MRDTblsp_PartitionedByIX_MRDIndex.

Data Type

relationship

pRefersBuffer

This attribute specifies an instance of the relationship MRDTblsp_RefersBuffer_MRDBffrp.

Data Type

relationship

primaryCreator

This attribute represents the primary authorization ID of the user who created the table space.

Data Type
string<8>

Default
null

pUsedByMTB

This attribute specifies an instance of the relationship MRDTblsp_UsedByMTB_MRDTable.

Data Type
relationship

Update Constraints
(document only)
only one table per partitioned table space.

segSize

This attribute indicates for a segmented tablespace the size of each segment. (must be 4 <= segsize <= 64)

Data Type
short

Valid Values
multiple of 4 (from 4 to 64)

Default
-1

space This attribute indicates number of kilobytes of DASD storage allocated to the table space.

Data Type
long

Default
-1

status This attribute indicates whether the table space was created implicitly.

- A - Availability
- C - Incomplete definition
- P - Table space is in Check Pending mode
- S - Table space is in Check Pending mode with the scope less than the entire table space
- T - Definition is incomplete because no table has been created

Data Type
char

Valid Values
'A', 'C', 'P', 'S', 'T' or blank

Default
null

keepInMainStorage

This attribute specifies if the tablespace should be kept in main storage.

Data Type
boolean

designStatus

This attribute indicates the status of the tablespace design.

- W — in Work
- C — Completed
- I — Inconsistent

Data Type

char

text This attribute specifies a detailed description of the tablespace.

Data Type

string

usageIntent

This attribute specifies the applications the database will be used for.

- Q — QMF
- B — Batch
- T — Test
- R — Random

Data Type

char

lockmax

This attribute indicates whether or not the LOCKMAX clause is to be generated for this tablespace and the value to be set. If this attribute is null, no LOCKMAX will be generated. Other values are 'S' - SYSTEM or 'V' - value specified in "maxValue" attribute.

Data Type

char

maxValue

An instance of this attribute contains the user-defined value for the LOCKMAX keyword. The value in this attribute is ignored if "lockmax" is null or has a value of 'S'. ****Constraint**** This attribute must have a value if lockmax is 'V'.

Data Type

long

maxRows

An instance of this attribute contains the user-defined value for the MAXROWS keyword. The value in this attribute can range from 1 to 255. ****Constraint**** Allowed values: 1-255

Data Type

short

lockPart

An instance of this attribute identifies whether or not the LOCKPART clause will be generated.

Data Type

boolean

largeTS

An instance of this attribute identifies whether or not the LARGE keyword will be generated on a CREATE TABLESPACE statement.

Data Type

boolean

ccsid This attribute indicates whether or not the CCSID clause is to be generated for this tablespace and the value to be set. If this attribute is null, no CCSID will be generated. Other values are 'E' - EBCDIC or 'A' - ASCII

Data Type

char

Cache View Methods

Name Description**DSRTablespaceInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: DSRMRDTSPartition

Cache View Name

DSRMRDTSPartition

Storage View Name

DSRMRDTSPartition

Inherits From

DSRMRDPartition

Description

This class represents the Tablespace Partition under Database2 for MVS.

The abbreviated name of this class (used in relationship classes)=MRDTSPrt

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD2

Storage View File Name

ewsr23

DLL Name

ewsrdb

Attributes

Name Description**checkFlag**

This attribute is used to identify check pending state of table space. If value

of this attribute is 'C', the table space partition is in check pending mode and there are rows that can violate referential constraints.

Data Type

char

Valid Values

'C' or blank

Default

null

checkRID

This attribute represents the RID of the first row of the table space partition that can violate referential constraint.

Data Type

string<4>

Default

null

compressFlag

This attribute indicates whether data compression applies to the rows of the partition.

- Y - Yes
- N - No

Data Type

char

Valid Values

'Y', 'N', or *null*

Default

null

pageSave

This attribute represents the percentage of pages, multiplied by 100, saved in the table space or partition as a result of using data compression.

Data Type

short

Default

-1

pUsedByMTS

This attribute specifies an instance of the relationship MRDTSPrt_UsedByMTS_MRDTblsp.

Data Type

relationship

Cache View Methods

Name	Description
------	-------------

DSRTSPartitionInit	
---------------------------	--

	This instance method initializes the attributes of this class.
--	--

Return Type	
--------------------	--

	void
--	------

Parameters
None

Object Class: DSRMRDView

Cache View Name
DSRMRDView

Storage View Name
DSRMRDView

Inherits From
DSRView

Description
An instance of this object class represent a View in a Database for DB2 for MVS.

The abbreviated name of this class (used in relationship classes)=MRDView

Constraint - The product code for an instance of this class related to the RDBSystem class via RCollection class must be MRD.

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD2

Storage View File Name
ewsr2m

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

pInMRDDatabase	This attribute specifies an instance of the relationship MRDView_InMRDDatabase_MRDDatbs.
-----------------------	--

Data Type	relationship
------------------	--------------

pInPhysicalDesign	This attribute specifies an instance of the relationshipDSRMRDView_In_MRDPDsgn
--------------------------	--

Data Type	relationship
------------------	--------------

pInToDoList	This attribute specifies an instance of the relationshipDSRMRDView_In_ToDoList
--------------------	--

Data Type
relationship

Cache View Methods

Name	Description
------	-------------

DSRViewInit	
--------------------	--

This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Object Class: DSRMRDVolume

Cache View Name
DSRMRDVolume

Storage View Name
DSRMRDVolume

Inherits From
DSTechnologyObject

Description

An instance of this class represents the volume used by a storage group in Database2 for MVS. This class is a subclass of TechnologyObject class to support view types in the repository, although many of the data members and member functions of the base class may not be applicable in this class.

The abbreviated name of this class (used in relationship classes)=MRDVolum

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD2

Storage View File Name
ewsr24

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

pUsedByMSG	
-------------------	--

This attribute specifies an instance of the relationship MRDVolum_UsedByMSG_MRDStrgg.

Data Type
relationship

volumeID

This attribute is the volume ID which identifies volume serial number of an OS/VS storage volume used for Storage Groups.

Data Type

string<6>

Valid Values

a valid SQL name or *null*

Default

null

Cache View Methods

Name	Description
------	-------------

DSRVolumelnit	
----------------------	--

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Relationship Class: DSRContainedInMRDD

Cache View Name

DSRContainedInMRDD

Storage View Name

DSRContainedInMRDD

Description

An instance of this relationship represents MRDIndex and MRDDatabase.

Source Information:**Class Name**

DSRMRDIndex

Attribute

pContainedInMRDD

Target Information:**Class Name**

DSRMRDDatabase

Attribute

pContainsMRDIndex

Forward Mapping Semantics:

Verb ContainedInMRDD

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb ContainsMRDIndex

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr3m

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkContainsPartition

Cache View Name
DSRLinkContainsPartition

Storage View Name
DSRLinkContainsPartition

Description
An instance of this relationship represents MRDIndexspace and MRDIndexPartition.

Source Information:

Class Name
DSRMRDIndexspace

Attribute
pContainsPartition

Target Information:

Class Name
DSRMRDIndexPartition

Attribute
pForMIS

Forward Mapping Semantics:

Verb ContainsPartition

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ForMIS

Cardinality
1..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWRD2

Storage View File Name
ewsr56

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkMRDBPoolInRDBSys

Cache View Name
DSRLinkMRDBPoolInRDBSys

Storage View Name
DSRLinkMRDBPoolInRDBSys

Description
An instance of this relationship represents DSRRDBSystem and MRDBufferpool.

Source Information:

Class Name
DSRMRDBufferpool

Attribute
pContainedInRDBS

Target Information:

Class Name
DSRRDBSystem

Attribute
pHasBufferpool

Forward Mapping Semantics:

Verb ContainedInRDBSys

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasBufferpool

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr61

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkClusteredISForMTB

Cache View Name
DSRLinkClusteredISForMTB

Storage View Name
DSRLinkClusteredISForMTB

Description
An instance of this relationship represents MRDIndexspace and MRDTable.
An instance of this relationship represents the fact that the MRDTable has a clustering Index. is getting partitioned by the MRDIndexspace

Attributes

None

Methods

None

Relationships

None

Relationship: DSRLinkHasPartitions

Description

An instance of this relationship represents MRDTablespace and MRDPartition.

Source Information:

Class Name

DSRMRDTablespace

Attribute

pHasPartitions

Target Information:

Class Name

DSRMRDTSPartition

Attribute

pUsedByMTS

Forward Mapping Semantics:

Verb HasPartitions

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb UsedByMTS

Cardinality

1..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWRD2

Storage View File Name

ewsr63

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkHasPartitions

Cache View Name

DSRLinkHasPartitions

Storage View Name

DSRLinkHasPartitions

Description

An instance of this relationship represents MRDTablespace and MRDPartition.

Source Information:

Class Name

DSRMRDTablespace

Attribute

pHasPartitions

Target Information:

Class Name

DSRMRDTSPartition

Attribute

pUsedByMTS

Forward Mapping Semantics:

Verb HasPartitions

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb UsedByMTS

Cardinality

1..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWRD2

Storage View File Name

ewsr74

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkMRDStgrpInRDBSys**Cache View Name**

DSRLinkMRDStgrpInRDBSys

Storage View Name

DSRLinkMRDStgrpInRDBSys

Description

An instance of this relationship represents DSRRDBSystem and MRDStoragegroup.

Source Information:**Class Name**

DSRMRDStoragegroup

Attribute

pContainedInRDBS

Target Information:**Class Name**

DSRMRDStoragegroup

Attribute

pContainedInRDBS

Forward Mapping Semantics:**Verb** ContainedInRDBSys**Cardinality**

1..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** HasStoragegroup**Cardinality**

0..m

Ordered

No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr75

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkTSInMRDDatabase

Cache View Name
DSRLinkTSInMRDDatabase

Storage View Name
DSRLinkTSInMRDDatabase

Description
An instance of this relationship represents MRDTablespace and MRDDatabase.

Source Information:

Class Name
DSRMRDTablespace

Attribute
pInMRDDatabase

Target Information:

Class Name
DSRMRDDatabase

Attribute
pHasTablespace

Forward Mapping Semantics:

Verb InMRDDatabase

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasTablespace

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr77

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRInMRDDatabase

Cache View Name
DSRInMRDDatabase

Storage View Name
DSRInMRDDatabase

Description
An instance of this relationship represents MRDView and MRDDatabase.

Source Information:

Class Name
DSRMRDView

Attribute
pInMRDDatabase

Target Information:

Class Name
DSRMRDDatabase

Attribute
pContainsMRDView

Forward Mapping Semantics:

Verb InMRDDatabase

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ContainsMRDView

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr2u

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkForMRDTable

Cache View Name
DSRLinkForMRDTable

Storage View Name
DSRLinkForMRDTable

Description
An instance of this relationship represents MRDIndex and MRDTable.

Source Information:

Class Name
DSRMRDIndex

Attribute
pForMRDTable

Target Information:

Class Name
DSRMRDTable

Attribute
pMRDIndexedBy

Forward Mapping Semantics:

Verb ForMRDTable

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** MRDIndexedBy**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr2z

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkMRDStoredIn

Cache View Name

DSRLinkMRDStoredIn

Storage View Name

DSRLinkMRDStoredIn

Description

An instance of this relationship represents MRDIndex and MRDIndexspace.

Source Information:**Class Name**

DSRMRDIndex

Attribute

pMRDStoredIn

Target Information:**Class Name**

DSRMRDIndexspace

Attribute
pMRDStoresIndex

Forward Mapping Semantics:

Verb MRDStoredIn

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb MRDStoresIndex

Cardinality
1..1

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr90

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkMRDDbaseInRColl

Cache View Name
DSRLinkMRDDbaseInRColl

Storage View Name
DSRLinkMRDDbaseInRColl

Description
An instance of this relationship represents RCollection and MRDDatabase.

Source Information:

Class Name
DSRMRDDatabase

Attribute
pOwnedByRC

Target Information:**Class Name**

DSRRCollection

Attribute

pOwnsMRDDatabase

Forward Mapping Semantics:**Verb** OwnedByRColl**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** OwnsMRDDatabase**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr91

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkMRDStgrpInRColl

Cache View Name

DSRLinkMRDStgrpInRColl

Storage View Name

DSRLinkMRDStgrpInRColl

Description

An instance of this relationship represents RCollection and MRDStoragegroup.

Source Information:

Class Name
DSRMRDStoragegroup

Attribute
pOwnedByRC

Target Information:

Class Name
DSRRCollection

Attribute
pOwnsMRDStogroup

Forward Mapping Semantics:

Verb OwnedByRColl

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb OwnsMRDStogroup

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr92

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkMRDTblspInRColl

Cache View Name
DSRLinkMRDTblspInRColl

Storage View Name
DSRLinkMRDTblspInRColl

Description

An instance of this relationship represents RCollection and MRDTablespace.

Source Information:**Class Name**

DSRMRDTablespace

Attribute

pOwnedByRC

Target Information:**Class Name**

DSRRCollection

Attribute

pOwnsMRDTablespace

Forward Mapping Semantics:

Verb OwnedByRColl

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb OwnsMRDTablespace

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr93

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkPartitionsTS

Cache View Name

DSRLinkPartitionsTS

Storage View Name

DSRLinkPartitionsTS

Description

An instance of this relationship represents MRDIndexspace and MRDTablespace.

Source Information:**Class Name**

DSRMRDIndexspace

Attribute

pPartitionsTS

Target Information:**Class Name**

DSRMRDTablespace

Attribute

pPartitionedByIX

Forward Mapping Semantics:

Verb PartitionsTS

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb PartitionedByIX

Cardinality

0..1

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr95

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkRefersBuffer

Cache View Name

DSRLinkRefersBuffer

Storage View Name

DSRLinkRefersBuffer

Description

An instance of this relationship represents MRDTablespace and MRDBufferpool.

Source Information:**Class Name**

DSRMRDTablespace

Attribute

pRefersBuffer

Target Information:**Class Name**

DSRMRDBufferpool

Attribute

pReferredByMTS

Forward Mapping Semantics:

Verb RefersBuffer

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb ReferredByMTS

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr97

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkRefersBufferpool

Cache View Name

DSRLinkRefersBufferpool

Storage View Name

DSRLinkRefersBufferpool

Description

An instance of this relationship represents MRDIndexspace and MRDBufferpool.

Source Information:

Class Name

DSRMRDIndexspace

Attribute

pRefersBufferpool

Target Information:

Class Name

DSRMRDBufferpool

Attribute

pReferredByMIS

Forward Mapping Semantics:

Verb RefersBufferpool

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb ReferredByMIS

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr98

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkRefersCatalog**Cache View Name**

DSRLinkRefersCatalog

Storage View Name

DSRLinkRefersCatalog

Description

An instance of this relationship represents MRDStoragegroup and MRDICFCatalog.

Source Information:**Class Name**

DSRMRDStoragegroup

Attribute

pRefersCatalog

Target Information:**Class Name**

DSRMRDICFCatalog

Attribute

pReferredByMSG

Forward Mapping Semantics:**Verb** RefersCatalog**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** ReferredByMSG**Cardinality**

0..m

Ordered

No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr99

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkRefersStogroup

Cache View Name
DSRLinkRefersStogroup

Storage View Name
DSRLinkRefersStogroup

Description
An instance of this relationship represents MRDPartition and MRDStoragegroup.

Source Information:

Class Name
DSRMRDPartition

Attribute
pRefersStogroup

Target Information:

Class Name
DSRMRDStoragegroup

Attribute
pForMP

Forward Mapping Semantics:

Verb RefersStogroup

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ForMP

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr0b

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

eraseRule	
------------------	--

This attribute shows the action to be performed on the DB2 defined data set.

- Y - Erase
- N - Not Erase

Data Type
char

Valid Values
'Y', 'N'

Default
null

priQty	This attribute shows the primary space allocation for DB2 defined data sets.
---------------	--

Data Type
long

Valid Values
0 to 4194304

Default
-1

secQty	
---------------	--

This attribute shows the secondary space allocation for DB2 defined data sets.

Data Type
long

Valid Values
0 to 131068

Default
-1

Cache View Methods

Name	Description
------	-------------

DSRLinkRefersStogroupInit

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Relationship Class: DSRLinkRefinedByMCDE**Cache View Name**

DSRLinkRefinedByMCDE

Storage View Name

DSRLinkRefinedByMCDE

Description

An instance of this relationship represents ColumnDef and MRDColumnDefExt.

Source Information:**Class Name**

DSRColumnDefinition

Attribute

pRefinedByMCDE

Target Information:**Class Name**

DSRMRDColumnDefExt

Attribute

pRefinesCD

Forward Mapping Semantics:

Verb RefinedByMCDE

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb RefinesCD

Cardinality

1..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWRD2

Storage View File Name

ewsrOf

DLL Name

ewsrdb

Attributes

Name Description**shareFlag**

This attribute is a flag to indicate how the database will be shared using shared read-only data.

- O - Owner (Target RDB system is the owner and can update the database)
- R - Read (Target RDB system has readonly access to the database)

Data Type

char

Valid Values

'O', 'R'

Default*null*

Cache View Methods

Name Description**DSRLinkRefinedByMCDEInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Relationship Class: DSRLinkROShare

Cache View Name

DSRLinkROShare

Storage View Name

DSRLinkROShare

Description

An instance of this relationship class represents a ReadOnly data sharing relationship between MRDDatabase and DSRRDBSystem objects.

Source Information:**Class Name**

DSRMRDDatabase

Attribute

pROShare

Target Information:**Class Name**

DSRRDBSystem

Attribute

pROSharedBy

Forward Mapping Semantics:**Verb** ROShare**Cardinality**
0..m**Ordered**
No**Controlling**
No**Inverse Mapping Semantics:****Verb** ROSharedBy**Cardinality**
0..m**Ordered**
No**Controlling**
No**Cache View File Name**
EWSWRD2**Storage View File Name**
ewsr0o**DLL Name**
ewsrdb**Attributes**

None

Cache View Methods

None

Relationship Class: DSRLinkSpaceRefersStogroup**Cache View Name**
DSRLinkSpaceRefersStogroup**Storage View Name**
DSRLinkSpaceRefersStogroup**Description**
An instance of this relationship represents MRDSpace and MRDStoragegroup.**Source Information:****Class Name**
DSRMRDSpace

Attribute
pRefersStogroup

Target Information:

Class Name
DSRMRDStoragegroup

Attribute
pForSpace

Forward Mapping Semantics:

Verb RefersStogroup

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ForSpace

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr7b

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

eraseRule

This attribute shows the action to be performed on the DB2 defined data set.

- Y - Erase
- N - Not Erase

Data Type
char

Valid Values
'Y', 'N'

Default
null

priQty This attribute shows the primary space allocation for DB2 defined data sets.

Data Type

long

Valid Values

0 to 4194304

Default

-1

secQty

This attribute shows the secondary space allocation for DB2 defined data sets.

Data Type

long

Valid Values

0 to 131068

Default

-1

exclusive

This attribute represents an exclusive use flag. If it is set to *true*, the storage group is exclusively reserved for the assigned tablespace.

Data Type

boolean

Valid Values*true* or *false***Default***false*

Cache View Methods

Name Description**DSRLinkSpaceRefersStogroupInit**

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Relationship Class: DSRLinkSpaceUsesCatalog

Cache View Name

DSRLinkSpaceUsesCatalog

Storage View Name

DSRLinkSpaceUsesCatalog

Description

An instance of this relationship represents MRDPartition and MRDICFCatalog.

Source Information:**Class Name**

DSRMRDSpace

Attribute
pUsesCatalog

Target Information:

Class Name
DSRMRDICFCatalog

Attribute
pForSpace

Forward Mapping Semantics:

Verb UsesCatalog

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ForSpace

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr7c

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkUsesBufferpool

Cache View Name
DSRLinkUsesBufferpool

Storage View Name
DSRLinkUsesBufferpool

Description

An instance of this relationship represents MRDDatabase and MRDBufferpool.

Source Information:**Class Name**

DSRMRDDatabase

Attribute

pUsesBufferpool

Target Information:**Class Name**

DSRMRDBufferpool

Attribute

pUsedByMDB

Forward Mapping Semantics:

Verb UsesBufferpool

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb UsedByMDB

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr0q

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkUsesCatalog

Cache View Name

DSRLinkUsesCatalog

Storage View Name

DSRLinkUsesCatalog

Description

An instance of this relationship represents MRDPartition and MRDICFCatalog.

Source Information:**Class Name**

DSRMRDPartition

Attribute

pUsesCatalog

Target Information:**Class Name**

DSRMRDICFCatalog

Attribute

pForMP

Forward Mapping Semantics:

Verb UsesCatalog

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb ForMP

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr0r

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkUsesDatabase

Cache View Name

DSRLinkUsesDatabase

Storage View Name

DSRLinkUsesDatabase

Description

An instance of this relationship represents MRDTable and MRDDatabase.

Source Information:**Class Name**

DSRMRDTable

Attribute

pUsesDatabase

Target Information:**Class Name**

DSRMRDDatabase

Attribute

pUsedByMTB

Forward Mapping Semantics:

Verb UsesDatabase

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb UsedByMTB

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr0s

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkUsesStogroup

Cache View Name

DSRLinkUsesStogroup

Storage View Name

DSRLinkUsesStogroup

Description

An instance of this relationship represents MRDDatabase and MRDStoragegroup.

Source Information:

Class Name

DSRMRDDatabase

Attribute

pUsesStogroup

Target Information:

Class Name

DSRMRDStoragegroup

Attribute

pUsedByMDB

Forward Mapping Semantics:

Verb UsesStogroup

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb UsedByMDB

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr0x

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkUsesTablespace**Cache View Name**

DSRLinkUsesTablespace

Storage View Name

DSRLinkUsesTablespace

Description

An instance of this relationship represents MRDTable and MRDTablespace.

Source Information:**Class Name**

DSRMRDTable

Attribute

pUsesTablespace

Target Information:**Class Name**

DSRMRDTablespace

Attribute

pUsedByMTB

Forward Mapping Semantics:**Verb** UsesTablespace**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** UsedByMTB**Cardinality**

0..m

Ordered

No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr2c

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkUsesVolume

Cache View Name
DSRLinkUsesVolume

Storage View Name
DSRLinkUsesVolume

Description
An instance of this relationship represents MRDStoragegroup and MRDVolume.

Source Information:

Class Name
DSRMRDStoragegroup

Attribute
pUsesVolume

Target Information:

Class Name
DSRMRDVolume

Attribute
pUsedByMSG

Forward Mapping Semantics:

Verb UsesVolume

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb UsedByMSG

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr0z

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkTableInMPDesign

Cache View Name
DSRLinkTableInMPDesign

Storage View Name
DSRLinkTableInMPDesign

Description
An instance of this relationship represents MRDPhysicalDesign and MRDTable

Source Information:

Class Name
DSRMRDTable

Attribute
pInPhysicalDesign

Target Information:

Class Name
DSRMRDPhysicalDesign

Attribute
pHasTables

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr7h

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkTblspaceInMPDesign

Cache View Name
DSRLinkTblspaceInMPDesign

Storage View Name
DSRLinkTblspaceInMPDesign

Description
An instance of this relationship represents MRDPhysicalDesign and MRDTablespace

Source Information:

Class Name
DSRMRDTablespace

Attribute
pInPhysicalDesign

Target Information:

Class Name
DSRMRDPhysicalDesign

Attribute
pHasTablespaces

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr7i

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkStogroupInMPDesign

Cache View Name
DSRLinkStogroupInMPDesign

Storage View Name
DSRLinkStogroupInMPDesign

Description
An instance of this relationship represents MRDPhysicalDesign and MRDStoragegroup

Source Information:

Class Name
DSRMRDStoragegroup

Attribute
pInPhysicalDesign

Target Information:

Class Name
DSRMRDPhysicalDesign

Attribute

pHasStogroup

Forward Mapping Semantics:**Verb** In**Cardinality**
0..m**Ordered**
No**Controlling**
No**Inverse Mapping Semantics:****Verb** Has**Cardinality**
0..m**Ordered**
No**Controlling**
No**Cache View File Name**

EWSWRD2

Storage View File Name

ewsr7j

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkViewInMPDesign

Cache View Name

DSRLinkViewInMPDesign

Storage View Name

DSRLinkViewInMPDesign

Description

An instance of this relationship represents MRDPhysicalDesign and MRD

Source Information:**Class Name**

DSRMRDView

Attribute

pInPhysicalDesign

Target Information:**Class Name**

DSRMRDPhysicalDesign

Attribute

pHasViews

Forward Mapping Semantics:**Verb** In**Cardinality**

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** Has**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr7k

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkIndexInMPDesign**Cache View Name**

DSRLinkIndexInMPDesign

Storage View Name

DSRLinkIndexInMPDesign

Description

An instance of this relationship represents MRDPhysicalDesign and MRDIndex

Source Information:

Class Name
DSRMRDIndex

Attribute
pInPhysicalDesign

Target Information:

Class Name
DSRMRDPhysicalDesign

Attribute
pHasIndexes

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD2

Storage View File Name
ewsr7l

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkDatabaseInMPDesign

Cache View Name
DSRLinkDatabaseInMPDesign

Storage View Name
DSRLinkDatabaseInMPDesign

Description

An instance of this relationship represents MRDPhysicalDesign and MRDDatabase

Source Information:**Class Name**

DSRMRDDatabase

Attribute

pInPhysicalDesign

Target Information:**Class Name**

DSRMRDPhysicalDesign

Attribute

pHasDatabases

Forward Mapping Semantics:

Verb In

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Has

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr7m

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkBufpoolInMPDesign

Cache View Name

DSRLinkBufpoolInMPDesign

Storage View Name

DSRLinkBufpoolInMPDesign

Description

An instance of this relationship represents MRDPhysicalDesign and MRDBufferpool.

Source Information:**Class Name**

DSRMRDBufferpool

Attribute

pInPhysicalDesign

Target Information:**Class Name**

DSRMRDPhysicalDesign

Attribute

pHasBufferpools

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name

EWSWRD2

Storage View File Name

ewsr7n

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Chapter 11. Relational Database: Common Server Submodel

Object Class: DSRORDDatabase

Cache View Name

DSRORDDatabase

Storage View Name

DSRORDDatabase

Inherits From

DSRDatabase

Description

An instance of this class represents a Database in Database2 for OS/2.

The abbreviated name of this class (used in relationship classes)=ORDDatbs

Constraint - The product code for an instance of this class related to the RDBSystem class via RCollection class must be ORD.

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWRD3

Storage View File Name

ewsr26

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

codePage	
-----------------	--

This attribute represents a code page for the comments entered for the Database description.

Data Type

string

Valid Values

a character string

Default

null

driveLetter	
--------------------	--

This attribute represents a one character letter for the drive where Database is created.

Data Type

char

Default
null

pHasOIndex
This attribute specifies an instance of the relationship
DSRORDDatabase_HasOIndex_Index.

Data Type
relationship

pHasOTable
This attribute specifies an instance of the relationship
DSRORDDatabase_HasOTable_ORDTable.

Data Type
relationship

pHasOView
This attribute specifies an instance of the relationship
DSRORDDatabase_HasOView_ORDView.

Data Type
relationship

pInPhysicalDesign
This attribute specifies an instance of the relationship
DSRORDDatbs_In_ORDPDsgn

Data Type
relationship

Cache View Methods

Name	Description
DSRORDDatabaseInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRORDIndex

Cache View Name
DSRORDIndex

Storage View Name
DSRORDIndex

Inherits From
DSRIndex

Description
An instance of this object class represents an Index in DB2/2.

The abbreviated name of this class (used in relationship
classes)=ORDIndex

Constraint - The product code for an instance of this class related to the
RDBSystem class via RCollection class must be ORD.

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD3

Storage View File Name
ewsr2s

DLL Name
ewsrdb

Attributes

Name	Description
pContainedInORDD	This attribute specifies an instance of the relationship DSRORDIndex_IndexHasORDD_ORDDatbs.
Data Type	relationship
pForORDTable	This attribute specifies an instance of the relationship DSRORDIndex_ForORDTable_ORDTable.
Data Type	relationship
pInPhysicalDesign	This attribute specifies an instance of the relationship DSRORDIndex_In_ORDPDsgn
Data Type	relationship

Cache View Methods

Name	Description
DSRORDIndexInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRORDPhysicalDesign

Cache View Name
DSRORDPhysicalDesign

Storage View Name
DSRORDPhysicalDesign

Inherits From

DSRPhysicalDesign

Description

An instance of this object class represent a PhysicalDesign in Database2 for MVS.

The abbreviated name of this class (used in relationship classes)=ORDPDsgn

Public Class

Yes

Extends

None

Class Extent

Versioned

Cache View File Name

EWSWRD3

Storage View File Name

ewsr8a

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

pHasTables

This attribute specifies an instance of the relationship ORDPDsgn_Has_ORDTable.

Data Type

relationship

pHasViews

This attribute specifies an instance of the relationship ORDPDsgn_Has_ORDView.

Data Type

relationship

pHasIndexes

This attribute specifies an instance of the relationship ORDPDsgn_Has_ORDIndex.

Data Type

relationship

pHasDatabases

This attribute specifies an instance of the relationship ORDPDsgn_Has_ORDDatbs.

Data Type

relationship

pHasTablespaces

This attribute specifies an instance of the relationship ORDPDsgn_Has_ORDTblspace

Data Type
relationship

Cache View Methods

Name **Description**

DSRORDPhysicalDesignInit

This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Object Class: DSRORDTable

Cache View Name
DSRORDTable

Storage View Name
DSRORDTable

Inherits From
DSRTable

Description

An instance of this class represents a Table in Database2 for OS/2.

The abbreviated name of this class (used in relationship classes)=ORDTable

Constraint - Product code in RDBSystem class related to an instance of this class via RCollection must be ORD.

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD3

Storage View File Name
ewsr27

DLL Name
ewsrdb

Attributes

Name **Description**

pContainedInORDD

This attribute specifies an instance of the relationship DSRORDTable_TableHasORDD_ORDDatbs.

Data Type
relationship

pInPhysicalDesign

This attribute specifies an instance of the relationshipDSRORDTable_In_ORDPDsgn

Data Type

relationship

pORDIndexedBy

This attribute specifies an instance of the relationshipDSRORDTable_ORDIndexedBy_ORDIndex.

Data Type

relationship

dataCaptureFlag

This attribute specified whether the logging of SQL INSERT, UPDATE, and DELETE operations on the table is augmented by additional information. N - None C - Changes

Data Type

char

Valid Values

'N' or 'C'

Default

null

pTablespace

This attribute specifies an instance of the relationshipORDTable_HasPrimary_ORDTblspce. It identifies the tablespace in which the table data will be stored

Data Type

relationship

pIndexInTS

This attribute specifies an instance of the relationshipORDTable_HasIndexIn_ORDTblspce. It identifies the tablespace in which the table index data will be stored

Data Type

relationship

pLongInTS

This attribute specifies an instance of the relationshipORDTable_HasLongIn_ORDTblspce. It identifies the tablespace in which the LOB table data will be stored

Data Type

relationship

Cache View Methods

Name	Description
DSRORDTableInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSRORDView

Cache View Name
DSRORDView

Storage View Name
DSRORDView

Inherits From
DSRView

Description
An instance of this class represents a View in Database2 for OS/2.
The abbreviated name of this class (used in relationship classes)=ORDView
Constraint - The product code for an instance of this class related to the RDBSystem class via RCollection class must be ORD.

Public Class
Yes

Extends
None

Class Extent
Versioned

Cache View File Name
EWSWRD3

Storage View File Name
ewsr28

DLL Name
ewsrdb

Attributes

Name	Description
pContainedInORDD	This attribute specifies an instance of the relationship DSRORDView_ViewHasORDD_ORDDatbs.

Data Type
relationship

pInPhysicalDesign	This attribute specifies an instance of the relationship DSRORDView_In_ORDPDsgn
--------------------------	---

Data Type
relationship

Cache View Methods

Name	Description
DSRORDViewInit	This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Object Class: DSRWRDColumnDefExt

Cache View Name
DSRWRDColumnDefExt

Storage View Name
DSRWRDColumnDefExt

Inherits From
DSRView

Description
An instance of this class represents the extensions to the columnDefinition needed for the Database2 for OS/2 or Database2 for AIX column definition.

The abbreviated name of this class (used in relationship classes)=WRDCImnD

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWRD3

Storage View File Name
ewsr48

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

pRefinesCD	This attribute specifies an instance of the relationship WRDCImnD_RefinesCD_CImnDfnt.
-------------------	---

Data Type	relationship
------------------	--------------

primaryFlag	This attribute indicates whether the column is in a primary key or not. Y - In primary key N - Not in primary key
--------------------	---

Data Type	char
------------------	------

Valid Values	'Y', 'N'
---------------------	----------

Default	<i>null</i>
----------------	-------------

loggedLOB

This attribute indicates whether or not changes to this LOB column are to be written to the log. Yes (TRUE) is normally the default.

Data Type

boolean

compactLOB

This attribute indicates whether or not data in this LOB column should take up minimal disk space. NO (FALSE) is the default.

Data Type

boolean

Cache View Methods

Name	Description
------	-------------

DSRWRDColumnDefExtInit	
-------------------------------	--

	This instance method initializes the attributes of this class.
--	--

Return Type	
--------------------	--

	void
--	------

Parameters	
-------------------	--

	None
--	------

Object Class: DSORDTablespace

Cache View Name

DSORDTablespace

Storage View Name

DSRORDTablespace

Inherits From

DSTechnologyObject

Description

An instance of this class represents a Tablespace used in Database2 for common servers.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD3

Storage View File Name

ewsr29

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

TSType

This attribute indicates the type of tablespace. The allowed values are: R - REGULAR L - Long (can hold LOB data) T - Temporary (only for temporary tables)

Data Type

char

Valid Values

'R', 'L', 'T'

Default

null

managedBy

This attribute indicates whether tablespace management is System (S) or Database (D) Allowed Values: S or D

Data Type

char

Valid Values

'S', 'D'

Default

null

managedByContainer

This attribute holds the container string portion of the MANAGED BY clause

Data Type

char

Default

null

comment

This attribute holds the ON COMMENT text for a tablespace ot.

Data Type

string

Default

null

pTable

This attribute specifies an instance of the relationship ORDTblspace_IsPrimaryFor_ORDTable.

Data Type

relationship

pIndexTSFor

This attribute specifies an instance of the relationship ORDTblspace_HasIndexesFor_ORDTable.

Data Type

relationship

pLongTSFor

This attribute specifies an instance of the relationship ORDTblspace_HasLongFor_ORDTable.

Data Type

relationship

pInPhysicalDesign

This attribute specifies an instance of the relationship
DSORDTblsp_In_ORDPDsgn

Data Type

relationship

pRDBSystem

This attribute specifies an instance of the
relationshipDSORDTblsp_In_RDBSystem

Data Type

relationship

Cache View Methods

None

Relationship Class: DSRLinkIndexHasORDD

Cache View Name

DSRLinkIndexHasORDD

Storage View Name

DSRLinkIndexHasORDD

Description

An instance of this relationship represents DSRIndex and
DSRORDDatabase.

Source Information:**Class Name**

DSRORDIndex

Attribute

FWDATTR

Target Information:**Class Name**

DSRORDDatabase

Attribute

pHasOIndex

Forward Mapping Semantics:

Verb IndexHasORDD

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb HasOIndex

Cardinality

0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD3

Storage View File Name
ewsr71

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkTableHasORDD

Cache View Name
DSRLinkTableHasORDD

Storage View Name
DSRLinkTableHasORDD

Description
An instance of this relationship represents DSRORDTable and DSRORDDatabase.

Source Information:

Class Name
DSRORDTable

Attribute
FWDATTR

Target Information:

Class Name
DSRORDDatabase

Attribute
pHasOTable

Forward Mapping Semantics:

Verb TableHasORDD

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasOTable

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD3

Storage View File Name
ewsr72

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkViewHasORDD

Cache View Name
DSRLinkViewHasORDD

Storage View Name
DSRLinkViewHasORDD

Description
An instance of this relationship represents DSRORDView and DSRORDDatabase.

Source Information:

Class Name
DSRORDView

Attribute
FWDATTR

Target Information:

Class Name
DSRORDDatabase

Attribute
pHasOView

Forward Mapping Semantics:

Verb ViewHasORDD

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasOView

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD3

Storage View File Name
ewsr73

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkForORDTable

Cache View Name
DSRLinkForORDTable

Storage View Name
DSRLinkForORDTable

Description
An instance of this relationship represents DSRORDIndex and DSRORDTable.

Source Information:

Class Name
DSRORDIndex

Attribute
FWDATTR

Target Information:

Class Name
DSRORDTable

Attribute
pORDIndexedBy

Forward Mapping Semantics:

Verb ForORDTable

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb ORDIndexedBy

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD3

Storage View File Name
ewsr3a

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkRefinedByWCDE

Cache View Name
DSRLinkRefinedByWCDE

Storage View Name
DSRLinkRefinedByWCDE

Description
An instance of this relationship represents ColumnDef and WRDColDefExt.

Source Information:

Class Name
DSRColumnDefinition

Attribute
pRefinedByWCDE

Target Information:

Class Name
DSRWRDColumnDefExt

Attribute
pRefinesCD

Forward Mapping Semantics:

Verb RefinedByWCDE

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb RefinesCD

Cardinality
1..1

Ordered
No

Controlling
Yes

Cache View File Name
EWSWRD3

Storage View File Name
ewsr0j

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

databaseType	
---------------------	--

This attribute indicates the type of database to which the Package belong.

- O - Database2 for OS/2
- X - Database2 for AIX

Data Type
char

Valid Values
'O', 'X'

Default
null

Cache View Methods

Name	Description
------	-------------

DSRLinkRefinedByWCDEInit	
---------------------------------	--

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Relationship Class: DSRLinkTableInOPDesign**Cache View Name**

DSRLinkTableInOPDesign

Storage View Name

DSRLinkTableInOPDesign

Description

An instance of this relationship represents ORDPhysicalDesign and ORDTTable

Source Information:**Class Name**

DSRORDTable

Attribute

pInPhysicalDesign

Target Information:**Class Name**

DSRORDPhysicalDesign

Attribute

pHasTables

Forward Mapping Semantics:**Verb** In**Cardinality**

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** Has**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD3

Storage View File Name

ewsr8b

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkViewInOPDesign

Cache View Name
DSRLinkViewInOPDesign

Storage View Name
DSRLinkViewInOPDesign

Description
An instance of this relationship represents ORDPhysicalDesign and ORD

Source Information:

Class Name
DSRORDView

Attribute
pInPhysicalDesign

Target Information:

Class Name
DSRORDPhysicalDesign

Attribute
pHasViews

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD3

Storage View File Name
ewsr8c

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkIndexInOPDesign

Cache View Name
DSRLinkIndexInOPDesign

Storage View Name
DSRLinkIndexInOPDesign

Description
An instance of this relationship represents ORDPhysicalDesign and ORDIndex

Source Information:

Class Name
DSRORDIndex

Attribute
pInPhysicalDesign

Target Information:

Class Name
DSRORDPhysicalDesign

Attribute
pHasIndexes

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD3

Storage View File Name
ewsr8d

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkDatabaseInOPDesign

Cache View Name
DSRLinkDatabaseInOPDesign

Storage View Name
DSRLinkDatabaseInOPDesign

Description
An instance of this relationship represents ORDPhysicalDesign and ORDDatabase

Source Information:

Class Name
DSRORDDatabase

Attribute
pInPhysicalDesign

Target Information:

Class Name
DSRORDPhysicalDesign

Attribute
pHasDatabases

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:**Verb** Has**Cardinality**
0..m**Ordered**
No**Controlling**
No**Cache View File Name**
EWSWRD3**Storage View File Name**
ewsr8e**DLL Name**
ewsrdb**Attributes**

None

Cache View Methods

None

Relationship Class: DSDSRLinkORDTblInTblspace**Cache View Name**
DSDSRLinkORDTblInTblspace**Storage View Name**
DSRLinkORDTblInTblspace**Description**
An instance of this relationship represents ORDTTable and ORDTTablespace**Source Information:****Class Name**
DSRORDTable**Attribute**
pTablespace**Target Information:****Class Name**
DSORDTablespace**Attribute**
pTable**Forward Mapping Semantics:****Verb** HasPrimary**Cardinality**
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb IsPrimaryFor

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD3

Storage View File Name
ewsr8f

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSDSRLinkORDTblIndexInTSpace

Cache View Name
DSDSRLinkORDTblIndexInTSpace

Storage View Name
DSRLinkORDTblIndexInTSpace

Description
An instance of this relationship represents ORDTTable and ORDTTablespace

Source Information:

Class Name
DSRORDTable

Attribute
pIndexInTS

Target Information:

Class Name
DSORDTablespace

Attribute
pIndexTSFor

Forward Mapping Semantics:

Verb HasIndexIn

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb HasIndexesFor

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD3

Storage View File Name
ewsr8g

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSDSRLinkORDTblLongInTSpace

Cache View Name
DSDSRLinkORDTblLongInTSpace

Storage View Name
DSRLinkORDTblLongInTSpace

Description
An instance of this relationship represents ORDTTable and ORDTTablespace

Source Information:

Class Name
DSRORDTable

Attribute
pLongInTS

Target Information:

Class Name
DSORDTablespace

Attribute

pLongTSFor

Forward Mapping Semantics:**Verb** HasLongIn**Cardinality**

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** HasLongFor**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD3

Storage View File Name

ewsr8h

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSDSRLinkTblspaceInOPDesign

Cache View Name

DSDSRLinkTblspaceInOPDesign

Storage View Name

DSRLinkTblspaceInOPDesign

Description

An instance of this relationship represents ORDDTable and
ORDPhysicalDesign

Source Information:**Class Name**

DSORDTablespace

Attribute
pInPhysicalDesign

Target Information:

Class Name
DSRORDPhysicalDesign

Attribute
pHasTablespaces

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD3

Storage View File Name
ewsr8i

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSDSRLinkORDTblspaceInRDBSys

Cache View Name
DSDSRLinkORDTblspaceInRDBSys

Storage View Name
DSRLinkORDTblspaceInRDBSys

Description

This attribute specifies an instance of the relationship
DSORDTblspace_In_RDBSystem

Source Information:**Class Name**

DSORDTablespace

Attribute

pRDBSystem

Target Information:**Class Name**

DSRRDBSystem

Attribute

pORDTblspace

Forward Mapping Semantics:

Verb In

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Contains

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD3

Storage View File Name

ewsr8j

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Chapter 12. Relational Database: Oracle Submodel

Object Class: DSOOracleIndex

Cache View Name

DSOracleIndex

Storage View Name

DSROracleIndex

Inherits From

DSRIndex

Description

An instance of this object class represents an Index in Oracle

The abbreviated name of this class (used in relationship classes)=ORAIndex

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD4

Storage View File Name

ewsr6o

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

pTable	This attribute specifies an instance of the relationship ORAIndex_For_ORATable.
---------------	---

Data Type	relationship
------------------	--------------

pPhysicalDesign	This attribute specifies an instance of the relationshipDSROracleIndex_In_PDesign
------------------------	---

Data Type	relationship
------------------	--------------

pTablespace	This attribute specifies an instance of the relationshipDSROracleIndex_In_Tablespace
--------------------	--

Data Type	relationship
------------------	--------------

Cache View Methods

Name	Description
DSROracleIndexInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSOraclePhysicalDesign

Cache View Name	DSOraclePhysicalDesign
Storage View Name	DSOraclePhysicalDesign
Inherits From	DSRPhysicalDesign
Description	An instance of this object class represent a PhysicalDesign in Oracle. The abbreviated name of this class (used in relationship classes)=OraclePDesign
Public Class	Yes
Extends	None
Class Extent	None
Cache View File Name	EWSWRD4
Storage View File Name	ewsr6q
DLL Name	ewsrdb

Attributes

Name	Description
pTable	This attribute specifies an instance of the relationship OraclePDesign_Has_OracleTable.
Data Type	relationship
pView	This attribute specifies an instance of the relationship OraclePDsgn_Has_OracleView.
Data Type	relationship

pIndex

This attribute specifies an instance of the relationship OraclePDsgn_Has_OracleIndex.

Data Type

relationship

pTablespace

This attribute specifies an instance of the relationship OraclePDesign_Has_OracleTbSpace.

Data Type

relationship

Cache View Methods

Name	Description
DSOraclePhysicalDesignInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: DSOracleTable

Cache View Name

DSOracleTable

Storage View Name

DSOracleTable

Inherits From

DSRTable

Description

An instance of this class represents a Table in Database2 for OS/2.

The abbreviated name of this class (used in relationship classes)=ORDTable

Constraint - Product code in RDBSystem class related to an instance of this class via RCollection must be ORD.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD4

Storage View File Name

ewsr6n

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

pPhysicalDesign	
------------------------	--

This attribute specifies an instance of the relationshipDSROracleTable_In_PDesign

Data Type	
------------------	--

relationship

pIndex	
---------------	--

This attribute specifies an instance of the relationshipDSROracleTable_Has_ORAIndex.

Data Type	
------------------	--

relationship

pTablespace	
--------------------	--

This attribute specifies an instance of the relationshipOracleTable_In_Tblspace. It identifies the tablespace in which the table data will be stored

Data Type	
------------------	--

relationship

Cache View Methods

Name	Description
------	-------------

DSROracleTableInit	
---------------------------	--

This instance method initializes the attributes of this class.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

None

Object Class: DSORacleView

Cache View Name	
------------------------	--

DSORacleView

Storage View Name	
--------------------------	--

DSORacleView

Inherits From	
----------------------	--

DSRView

Description	
--------------------	--

An instance of this class represents a View in Oracle

The abbreviated name of this class (used in relationship classes)=DSROracleView

Public Class	
---------------------	--

Yes

Extends	
----------------	--

None

Class Extent	
---------------------	--

None

Cache View File Name
EWSWRD4

Storage View File Name
ewsr6p

DLL Name
ewsrdb

Attributes

Name **Description**

force This attribute indicates whether or not the FORCE clause is to be generated in the Oracle CREATE VIEW DDL.

Data Type
boolean

orReplace

This attribute indicates whether or not the OR REPLACE clause is to be generated in the Oracle CREATE VIEW DDL.

Data Type
boolean

constraintName

This attribute indicates contains the constraint name that is part WITH CHECK OPTION clause. If supplied and wOption has a value of CO, it will appear in the Oracle CREATE VIEW DDL.

Data Type
string

wOption

This attribute indicates whether or not the WITH clause is to be generated for this VIEW and the value to be set. If this attribute is null, no clause will be generated. Other values are 'RO' - READ ONLY or 'CO' - CHECK OPTION

Data Type
char

pPhysicalDesign

This attribute specifies an instance of the relationshipDSROracleView_In_PDesign

Data Type
relationship

Cache View Methods

Name **Description**

DSROracleViewInit

This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Object Class: DSOOracleTablespace

Cache View Name

DSOracleTablespace

Storage View Name

DSROracleTablespace

Inherits From

DSTechnologyObject

Description

An instance of this class represents a Tablespace used in Oracle.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD4

Storage View File Name

ewsr6r

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

pTable

This attribute specifies an instance of the relationship OracleTblspace_For_OracleTable.

Data Type

relationship

pIndex

This attribute specifies an instance of the relationship OracleTblspace_For_OracleIndex.

Data Type

relationship

pPhysicalDesign

This attribute specifies an instance of the relationship OracleTblspace_In_PDesign

Data Type

relationship

pRDBSystem

This attribute specifies an instance of the relationship DSOOracleTblspace_In_RDBSystem

Data Type

relationship

Cache View Methods

Name	Description
------	-------------

DSROracleTablespaceInit	
--------------------------------	--

This instance method initializes the attributes of this class.

Return Type	
--------------------	--

void

Parameters	
-------------------	--

None

Relationship Class: DSRLinkOclIndexForTable

Cache View Name	
------------------------	--

DSRLinkOclIndexForTable

Storage View Name	
--------------------------	--

DSRLinkOclIndexForTable

Description	
--------------------	--

An instance of this relationship represents DSROracleIndex and DSROracleTable

Source Information:	
----------------------------	--

Class Name	
-------------------	--

DSOracleIndex

Attribute	
------------------	--

pTable

Target Information:	
----------------------------	--

Class Name	
-------------------	--

DSOracleTable

Attribute	
------------------	--

pIndex

Forward Mapping Semantics:	
-----------------------------------	--

Verb	For
-------------	-----

Cardinality	
--------------------	--

0..1

Ordered	
----------------	--

No

Controlling	
--------------------	--

No

Inverse Mapping Semantics:	
-----------------------------------	--

Verb	IndexedBy
-------------	-----------

Cardinality	
--------------------	--

0..m

Ordered	
----------------	--

No

Controlling	
--------------------	--

No

Cache View File Name
EWSWRD4

Storage View File Name
ewsr6m

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkOclTableInPDesign

Cache View Name
DSRLinkOclTableInPDesign

Storage View Name
DSRLinkOclTableInPDesign

Description
An instance of this relationship represents DSROracleTable and DSROraclePhysicalDesign

Source Information:

Class Name
DSOracleTable

Attribute
pPhysicalDesign

Target Information:

Class Name
DSOraclePhysicalDesign

Attribute
pTable

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD4

Storage View File Name
ewsr6f

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkOclIndexInPDesign

Cache View Name
DSRLinkOclIndexInPDesign

Storage View Name
DSRLinkOclIndexInPDesign

Description
An instance of this relationship represents DSROracleIndex and DSROraclePhysicalDesign

Source Information:

Class Name
DSOracleIndex

Attribute
pPhysicalDesign

Target Information:

Class Name
DSOraclePhysicalDesign

Attribute
pIndex

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:**Verb** Has**Cardinality**
0..m**Ordered**
No**Controlling**
No**Cache View File Name**
EWSWRD4**Storage View File Name**
ewsr6g**DLL Name**
ewsrdb**Attributes**

None

Cache View Methods

None

Relationship Class: DSRLinkOclTSpaceInPDesign**Cache View Name**
DSRLinkOclTSpaceInPDesign**Storage View Name**
DSRLinkOclTSpaceInPDesign**Description**
An instance of this relationship represents DSROracleTable and DSROraclePhysicalDesign**Source Information:****Class Name**
DSOracleTablespace**Attribute**
pPhysicalDesign**Target Information:****Class Name**
DSOraclePhysicalDesign**Attribute**
pTablespace**Forward Mapping Semantics:****Verb** In**Cardinality**
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD4

Storage View File Name
ewsr6i

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkOclViewInPDesign

Cache View Name
DSRLinkOclViewInPDesign

Storage View Name
DSRLinkOclViewInPDesign

Description
An instance of this relationship represents DSROracleView and DSROraclePhysicalDesign

Source Information:

Class Name
DSOracleView

Attribute
pPhysicalDesign

Target Information:

Class Name
DSOraclePhysicalDesign

Attribute
pView

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD4

Storage View File Name
ewsr6h

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkOclTableInTblspace

Cache View Name
DSRLinkOclTableInTblspace

Storage View Name
DSRLinkOclTableInTblspace

Description
An instance of this relationship represents DSROracleTable and DSROracleTablespace

Source Information:

Class Name
DSOracleTable

Attribute
pTablespace

Target Information:

Class Name
DSOracleTablespace

Attribute
pTable

Forward Mapping Semantics:

Verb In

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb For

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD4

Storage View File Name
ewsr6k

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkOclIndexInTblspace

Cache View Name
DSRLinkOclIndexInTblspace

Storage View Name
DSRLinkOclIndexInTblspace

Description
An instance of this relationship represents DSROracleIndex and DSROracleTablespace

Source Information:

Class Name
DSOracleIndex

Attribute
pTablespace

Target Information:

Class Name
DSOracleTablespace

Attribute
pIndex

Forward Mapping Semantics:

Verb In

Cardinality
0..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb For

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD4

Storage View File Name
ewsr6l

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkOclTblspcInRDBSys

Cache View Name
DSRLinkOclTblspcInRDBSys

Storage View Name
DSRLinkOclTblspcInRDBSys

Description

An instance of this relationship represents DSROracleTablespace and DSRRDBSystem

Source Information:**Class Name**

DSOracleTablespace

Attribute

pRDBSystem

Target Information:**Class Name**

DSRRDBSystem

Attribute

pOracleTS

Forward Mapping Semantics:

Verb In

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Contains

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD4

Storage View File Name

ewsr6j

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Chapter 13. Relational Database: Designer Extensions to the MVS Submodel

Object Class: DSRMRDTablePartition

Cache View Name

DSRMRDTablePartition

Storage View Name

DSRMRDTablePartition PARENTLINE

Description

An instance of this object class represents a table partition. A table partition is used for the physical database design. It is the bases for the proposal for a tablespace partition.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWRD5

Storage View File Name

ewsr5k

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

partitionNumber	
------------------------	--

This attribute represents the table partition number.

Data Type

short

Valid Values

0 to 64

Default

-1

alterTableFrequency	
----------------------------	--

This attribute specifies how frequently an alter table is executed on the table.

Data Type

short

Valid Values

0 to 10

Default

-1

concurrency

This attribute specifies the concurrency requirements.

- L - Low
- M - Medium
- H - High

Data Type

char

Valid Values

'L', 'M', 'H'

Default

null

deleteFrequency

This attribute specifies how frequently a delete statement is executed on the table.

Data Type

short

Valid Values

0 to 10

Default

-1

insertFrequency

This attribute specifies how frequently an insert statement is executed on the table.

Data Type

short

Valid Values

0 to 10

Default

-1

reorgFrequency

This attribute specifies how frequently an reorg is executed on the table.

Data Type

short

Valid Values

0 to 10

Default

-1

selectFrequency

This attribute specifies how frequently a select statement is executed on the table.

Data Type

short

Valid Values

0 to 10

Default

-1

unloadReloadFrequency

This attribute specifies how frequently an unload and reload operation is executed on the table.

Data Type

short

Valid Values

0 to 10

Default

-1

updateFrequency

This attribute specifies how frequently an update statement is executed on the table.

Data Type

short

Valid Values

0 to 10

Default

-1

confidenceFactor

This attribute specifies the confidence factor of the growth and delete rate values. It has an influence on the secondary quantity calculations. It is specified as a percentage.

Data Type

short

Valid Values

0 to 100

Default

-1

deletePeriod

This attribute specifies the period of the delete rate.

- D - Daily
- W - Weekly
- M - Monthly
- Q - Quarterly

Data Type

char

Valid Values

'D', 'W', 'M', or 'Q'

Default

null

deleteRate

This attribute specifies the delete rate of the table.

Data Type

short

Default

-1

growthPeriod

This attribute specifies the period of the growth rate.

- D - Daily
- W - Weekly
- M - Monthly
- Q - Quarterly

Data Type

char

Valid Values

'D', 'W', 'M', or 'Q'

Default

null

growthRate

This attribute specifies the growth rate of the table.

Data Type

short

Default

-1

initialNoOfRows

This attribute specifies the initial number of rows in the table.

Data Type

long

Default

-1

maintenancePeriod

This attribute specifies the amount of time for which the growth and delete rates are calculated.

Data Type

short

Default

-1

securityOptionSet

This attribute specifies whether the data is security sensitive. It is the basis of the setting of the 'Erase option'.

Data Type

boolean

Default

false

pMRDPartitionOf

This attribute specifies an instance of the relationship
MRDTP_PartitionOf_MRDTable

Data Type

relationship

pMRDTPHasColumns

This attribute specifies an instance of the relationship
MRDTP_HasColumns_ClmnDef

Data Type
relationship

Cache View Methods

Name	Description
------	-------------

DSRMRDTablePartitionInit	
---------------------------------	--

This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Object Class: DSRTodoList

Cache View Name
DSRTodoList

Storage View Name
DSRTodoList PARENTLINE

Description

An instance of this object class represents a list of PhysicalDesign objects that have pending work in the Physical Data Designer.

The abbreviated name of this class (used in relationship classes)=ToDoList

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWRD5

Storage View File Name
ewsr7o

DLL Name
ewsrdb

Attributes

Name	Description
------	-------------

Date	This attribute represents the date (yymmdd) when the entry was made, closed, or rejected in the ToDoList.
-------------	---

Data Type
string<6>

Default
null

Status
This attribute represents the status of the entry in the ToDoList.

- C - Closed

- O - Open
- R - Rejected

Data Type

char

Valid Values

'C', 'O', 'R', or *null*

Default

null

Created_by

This attribute specifies who created the entry in the ToDoList.

- SYSTEM - entry was created by DataAtlas Designer
- <userid> - Userid of the logged on user

Data Type

string<30>

Valid Values

'SYSTEM', <userid>, or *null*

Default

null

Todo_entry

This attribute represents the action which has to be done.

Data Type

string<256>

Default

null

Rule This attribute represents the rule id which is assigned to the ToDoList entry.

Data Type

string<8>

Default

null

pHasAltNames

This attribute specifies an instance of the relationship
ToDoList_Has_AltName.

Data Type

relationship

pHasTables

This attribute specifies an instance of the relationship
ToDoList_Has_MRDTable.

Data Type

relationship

pHasTablespaces

This attribute specifies an instance of the relationship
ToDoList_Has_MRDTblsp.

Data Type

relationship

pHasStogroup

This attribute specifies an instance of the relationship
ToDoList_Has_MRDCStrgg.

Data Type

relationship

pHasViews

This attribute specifies an instance of the relationship
ToDoList_Has_MRDCView.

Data Type

relationship

pHasIndexes

This attribute specifies an instance of the relationship
ToDoList_Has_MRDCIndex.

Data Type

relationship

pHasDatabases

This attribute specifies an instance of the relationship
ToDoList_Has_MRDCDatbs.

Data Type

relationship

pHasBufferpools

This attribute specifies an instance of the relationship
ToDoList_Has_MRDCBffrp.

Data Type

relationship

pInPDesign

This attribute specifies an instance of the relationship
ToDoList_In_PhysDsgn.

Data Type

relationship

Cache View Methods

Name	Description
DSRToDoListInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Relationship Class: DSRLinkMRDHasPartitions

Cache View Name

DSRLinkMRDHasPartitions

Storage View Name

DSRLinkMRDHasPartitions

Description

An instance of this relationship represents MRDTable and MRDTablePartition.

Source Information:**Class Name**

DSRMRDTable

Attribute

pMRDHasPartitions

Target Information:**Class Name**

DSRMRDTablePartition

Attribute

pMRDPartitionOf

Forward Mapping Semantics:

Verb HasPartitions

Cardinality

0..n

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb PartitionOf

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWRD5

Storage View File Name

ewsr5x

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkMRDTPHasColumns

Cache View Name

DSRLinkMRDTPHasColumns

Storage View Name

DSRLinkMRDTPHasColumns

Description

An instance of this relationship represents MRDTablePartition and ColumnDefinition.

Source Information:**Class Name**

DSRMRDTablePartition

Attribute

pMRDTPHasColumns

Target Information:**Class Name**

DSRColumnDefinition

Attribute

pColumnOfMRDTP

Forward Mapping Semantics:

Verb HasColumns

Cardinality

0..n

Ordered

Yes

Controlling

No

Inverse Mapping Semantics:

Verb ColumnOf

Cardinality

0..n

Ordered

No

Controlling

No

Cache View File Name

EWSWRD5

Storage View File Name

ewsr6c

DLL Name

ewsrdb

Attributes

Name	Description
------	-------------

rangeValue

This attribute is used together with all other range values for the table partition to define the highest key value of the table partition. The concatenation of all range values is used to define the corresponding partition of the index.

Data Type

string<512>

Default

null

Cache View Methods

Name	Description
------	-------------

DSRLinkMRDTPHasColumnsInit	
-----------------------------------	--

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Relationship Class: DSRLinkToDoListHasAltNames

Cache View Name

DSRLinkToDoListHasAltNames

Storage View Name

DSRLinkToDoListHasAltNames

Description

An instance of this relationship represents ToDoList and AlternateName.

Source Information:**Class Name**

DSRToDoList

Attribute

pHasAltNames

Target Information:**Class Name**

DSRAlternateName

Attribute

pInToDoList

Forward Mapping Semantics:

Verb Has

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** In**Cardinality**
0..m**Ordered**
No**Controlling**
No**Cache View File Name**
EWSWRD5**Storage View File Name**
ewsr7q**DLL Name**
ewsrdb**Attributes**

None

Cache View Methods

None

Relationship Class: DSRLinkToDoListHasTables**Cache View Name**
DSRLinkToDoListHasTables**Storage View Name**
DSRLinkToDoListHasTables**Description**
An instance of this relationship represents ToDoList and MRDTable**Source Information:****Class Name**
DSRToDoList**Attribute**
pHasTables**Target Information:****Class Name**
DSRMRDTable**Attribute**
pInToDoList**Forward Mapping Semantics:****Verb** Has**Cardinality**
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD5

Storage View File Name
ewsr7r

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkToDoListHasTblspaces

Cache View Name
DSRLinkToDoListHasTblspaces

Storage View Name
DSRLinkToDoListHasTblspaces

Description
An instance of this relationship represents ToDoList and MRDTablespace

Source Information:

Class Name
DSRToDoList

Attribute
pHasTablespaces

Target Information:

Class Name
DSRMRDTablespace

Attribute
pInToDoList

Forward Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD5

Storage View File Name
ewsr7s

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkToDoListHasStogroup

Cache View Name
DSRLinkToDoListHasStogroup

Storage View Name
DSRLinkToDoListHasStogroup

Description
An instance of this relationship represents ToDoList and MRDStoragegroup

Source Information:

Class Name
DSRToDoList

Attribute
pHasStogroup

Target Information:

Class Name
DSRMRDStoragegroup

Attribute

pInToDoList

Forward Mapping Semantics:**Verb** Has**Cardinality**
0..m**Ordered**
No**Controlling**
No**Inverse Mapping Semantics:****Verb** In**Cardinality**
0..m**Ordered**
No**Controlling**
No**Cache View File Name**
EWSWRD5**Storage View File Name**
ewsr7t**DLL Name**
ewsrdb**Attributes**

None

Cache View Methods

None

Relationship Class: DSRLinkToDoListHasViews**Cache View Name**
DSRLinkToDoListHasViews**Storage View Name**
DSRLinkToDoListHasViews**Description**
An instance of this relationship represents ToDoList and MRD**Source Information:****Class Name**
DSRToDoList**Attribute**
pHasViews

Target Information:**Class Name**

DSRMRDView

Attribute

pInToDoList

Forward Mapping Semantics:**Verb** Has**Cardinality**

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** In**Cardinality**

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD5

Storage View File Name

ewsr7u

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkToDoListHasIndexes**Cache View Name**

DSRLinkToDoListHasIndexes

Storage View Name

DSRLinkToDoListHasIndexes

Description

An instance of this relationship represents ToDoList and MRDIndex

Source Information:

Class Name
DSRToDoList

Attribute
pHasIndexes

Target Information:

Class Name
DSRMRDIndex

Attribute
pInToDoList

Forward Mapping Semantics:

Verb Has

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWRD5

Storage View File Name
ewsr7v

DLL Name
ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkToDoListHasDatabases

Cache View Name
DSRLinkToDoListHasDatabases

Storage View Name
DSRLinkToDoListHasDatabases

Description

An instance of this relationship represents ToDoList and MRDDatabase

Source Information:**Class Name**

DSRToDoList

Attribute

pHasDatabases

Target Information:**Class Name**

DSRMRDDatabase

Attribute

pInToDoList

Forward Mapping Semantics:

Verb Has

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb In

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD5

Storage View File Name

ewsr7w

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRLinkToDoListHasBufpools

Cache View Name

DSRLinkToDoListHasBufpools

Storage View Name

DSRLinkToDoListHasBufpools

Description

An instance of this relationship represents ToDoList and MRDBufferpool.

Source Information:**Class Name**

DSRToDoList

Attribute

pHasBufferpools

Target Information:**Class Name**

DSRMRDBufferpool

Attribute

pInToDoList

Forward Mapping Semantics:

Verb Has

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb In

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWRD5

Storage View File Name

ewsr7x

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Relationship Class: DSRPDesignHasToDoList

Cache View Name

DSRPDesignHasToDoList

Storage View Name

DSRPDesignHasToDoList

Description

An instance of this relationship represents PhysicalDesign and ToDoList.

Source Information:**Class Name**

DSRPhysicalDesign

Attribute

pHasToDoList

Target Information:**Class Name**

DSRToDoList

Attribute

pInPDesign

Forward Mapping Semantics:

Verb Has

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb In

Cardinality

0..1

Ordered

No

Controlling

Yes

Cache View File Name

EWSWRD5

Storage View File Name

ewsr7p

DLL Name

ewsrdb

Attributes

None

Cache View Methods

None

Chapter 14. Transform Mapping Model

Object Class: cdmModel

Cache View Name

cdmModel

Storage View Name

cdmModel

Inherits From

TCPart

Description

An instance of this object class represents a Data Model containing entities and relationships that define a logical model of a relational database.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

CDMWCPM

Storage View File Name

cdmmod

DLL Name

ewswcdm

Attributes

Name	Description
------	-------------

data	This attribute represents the data contained in the data model in a CDM internal representation.
-------------	--

Data Type

sequence(octet)

fileName

This attribute represents the data model file name which is materialized in a file (without extension; extension is always .IR).

Data Type

string(8)

modelType

This attribute indicates the type of the data model:

0 - an SERM data model

1 - an ERM data model

Data Type

short

fLocked

This attribute indicates whether or not the data is currently materialized in a file.

Data Type

boolean (short)

transformedRelDesigns

This attribute collects the results of the transformation processes from the data model to a DSRRelationalDesign (forward engineering) and vice versa (reverse engineering).

Data Type

relationship

size

This attribute indicates the size in bytes of the data model blob representation (attribute data).

Data Type

integer

pUsesDE

This attribute locates the DSDataElements that are used/linked to a data model.

Data Type

relationship

Cache View Methods

Name	Description
cdmModelInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: cdmTransformProcess

Cache View Name

DSOraclePhysicalDesign

Storage View Name

DSOraclePhysicalDesign

Inherits From

TCPart

Description

An instance of this class represents the results of a transformation process from a CDM data model to an ATLAS relational design (forward engineering) or vice versa (reverse engineering).

Public Class

Yes

Extends

None

Class Extent
None

Cache View File Name
EWSWCDM

Storage View File Name
cdmtra

DLL Name
ewswcdm

Attributes

Name	Description
------	-------------

pModel

This attribute locates the CDM data model which is the source (target) of the transformation.

Data Type

relationship

pTransformedToRD

This attribute locates the RelationalDesign which was the target of the transformation.

Data Type

relationship

RDBSystemName

Name of the RDBSystem that was target of the transformation.

Data Type

char(254)

PhysicalDesignName

Name of the PhysicalDesign that was target of the transformation.

Data Type

char(254)

DeleteRuleOptionalSetting

This attribute represents the foreign key delete rule that is set for optional relationships of the transformation.

Data Type

char

Valid Values

N	ON_DELETE_SET_NULL
R	ON_DELETE_RESTRICT
null	ON_DELETE_NONE

DeleteRuleMandatorySetting

This attribute represents the foreign key delete rule that is set for mandatory relationships of the transformation.

Data Type

char

Valid Values

N	ON_DELETE_SET_NULL
----------	--------------------

R	ON_DELETE_RESTRICT
null	ON_DELETE_NONE

fGenerateIndexOnPK

Flag whether primary key indices were created during the transformation.

Data Type

boolean

NameReductionRules

This attribute represents the name reduction rules that were used during the transformation.

Data Type

integer

TablePrefix

This attribute represents the table prefix character string that was used during the transformation.

Data Type

char(254)

DatabaseName

Name of the SQL data base that was target of the transformation.

Data Type

char(254)

RCollectionName

Name of the RCollection that was target of the transformation.

Data Type

char(254)

TargetDBMS

Name of the RDBSystem that was target of the transformation.

Data Type

char(254)

Valid Values

0	no RDB System
1	DB2 for MVS
2	SQL/DS
3	DB2 for OS/2
4	SQL/400
5	ANSI SQL
6	DB2 for AIX
7	

Cache View Methods

Name Description

cdmTransformProcessInit

This instance method initializes the attributes of this class.

Return Type

void

Parameters

None

Object Class: cdmAttributeToColumnMapping**Cache View Name**

cdmAttributeToColumnMapping

Storage View Name

cdmAttributeToColumnMapping

Inherits From

cdmMapping

Description

An instance of this class represents the mapping between a CDM attribute and the transformed ATLAS column.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWCDM

Storage View File Name

cdma2c

DLL Name

ewswcdm

Attributes**Name Description****pLinkE2TDMapping**

This attribute locates the entity to table mapping object the attribute belongs to.

Data Type

relationship

attributeName

This attribute represents the name of the CDM attribute.

Data Type

char(64)

index This attribute represents index of this attribute within its entity.

Data Type

short

pLinkedToColumnDefinition

This attribute represents the link between a RDB column definition and a CDM attribute.

Data Type
relationship

Cache View Methods

Name	Description
cdmAttributeToColumnMappingInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Object Class: cdmEntityToTableMapping

Cache View Name	cdmEntityToTableMapping
Storage View Name	cdmEntityToTableMapping
Inherits From	cdmMapping
Description	An instance of this class represents the mapping between a CDM entity and the transformed ATLAS table.
Public Class	Yes
Extends	None
Class Extent	None
Cache View File Name	EWSWCDM
Storage View File Name	cdme2t
DLL Name	ewswcdm

Attributes

Name	Description
pTransformProcess	This attribute locates the transform process where this mapping rule is a result of.DL.
Data Type	relationship
entityName	This attribute represents the name of the CDM entity.

Data Type
char(64)

pLinkedToTableDefinition

This attribute represents the link between a RDB table definition and a CDM entity.

Data Type
relationship

pLinkedToTable

This attribute represents the link between a RDB table and a CDM entity.

Data Type
relationship

pHasR2FKMapping

This attribute locates a relationship to foreign key mapping instance that belongs to this entity (located by cdmID)

Data Type
relationship

pHasA2CMapping

This attribute locates an attribute to column mapping instance that belongs to this entity (located by cdmId).

Data Type
relationship

Cache View Methods

Name	Description
------	-------------

cdmEntityToTableMappingInit	
------------------------------------	--

	This instance method initializes the attributes of this class.
--	--

Return Type	void
--------------------	------

Parameters	None
-------------------	------

Object Class: cdmMapping

Cache View Name
cdmMapping

Storage View Name
cdmMapping

Inherits From
ADObject

Description

This class is the superclass for all kinds of mappings between CDM objects and ATLAS objects. Because the CDM data model is not stored on a fine grained level in TC the relationship is kept via the unique CDM Ids of the participating CDM instances.

Public Class
Yes

Extends
None

Class Extent
None

Cache View File Name
EWSWCDM

Storage View File Name
cdmmap

DLL Name
ewswcdm

Attributes

Name **Description**

cdmId This attribute represents to the unique object Id of the CDM object in the internal representation.

Data Type
integer

fVisited

This attribute represents a flag which is needed during retransformation of a data model into the same relational design (or vice versa). It indicates whether the objects located by their Ids have already been visited during a transformation process or not.

Data Type
short

Valid Values

0	not visited
1	visited

Cache View Methods

Name **Description**

cdmMappingInit
This instance method initializes the attributes of this class.

Return Type
void

Parameters
None

Object Class: cdmRelshipToFKeyMapping

Cache View Name
cdmRelshipToFKeyMapping

Storage View Name
cdmRelshipToFKeyMapping

Inherits From
cdmMapping

Description

An instance of this class represents the mapping between a CDM relationship and the transformed ATLAS foreign key.

Public Class

Yes

Extends

None

Class Extent

None

Cache View File Name

EWSWCDM

Storage View File Name

cdmr2f

DLL Name

ewswcdm

Attributes

Name	Description
pLinkE2TDMapping	This attribute locates the entity to table mapping object the relationship belongs to.
Data Type	relationship
RelationshipName	The name of the relationship.
Data Type	char(64)
pLinkedToKeyDefinition	This attribute represents the link between a RDB key definition and a CDM relationship.
Data Type	relationship

Cache View Methods

Name	Description
cdmRelshipToFKeyMappingInit	This instance method initializes the attributes of this class.
Return Type	void
Parameters	None

Relationship Class: cdmA2CMPartOfE2TM

Cache View Name

cdmA2CMPartOfE2TM

Storage View Name

cdmA2CMPartOfE2TM

Description

An instance of this relationship represents cdmEntityToTableMapping and cdmAttributeToColumnMapping

Source Information:**Class Name**

cdmEntityToTableMapping

Attribute

pHasA2CMapping

Target Information:**Class Name**

cdmAttributeToColumnMapping

Attribute

pLinkE2TDMapping

Forward Mapping Semantics:

Verb For

Cardinality

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb IndexedBy

Cardinality

1..1

Ordered

No

Controlling

No

Cache View File Name

EWSWCDM

Storage View File Name

cdmI07

DLL Name

ewswcdm

Attributes

None

Cache View Methods

None

Relationship Class: cdmLinkAttribute2CDef

Cache View Name

cdmLinkAttribute2CDef

Storage View Name

cdmLinkAttribute2CDef

Description

An instance of this relationship represents cdmAttributeToColumnMapping and DSRColumnDefinition

Source Information:**Class Name**

cdmAttributeToColumnMapping

Attribute

pLinkedToColumnDefinition

Target Information:**Class Name**

DSColumnDefinition

Attribute

pBasedOnAttribute

Forward Mapping Semantics:

Verb In

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb Has

Cardinality
0..1

Ordered
No

Controlling
No

Cache View File Name

EWSWCDM

Storage View File Name

cdmI05

DLL Name

ewswcdm

Attributes

None

Cache View Methods

None

Relationship Class: cdmLinkEntity2Table

Cache View Name

cdmLinkEntity2Table

Storage View Name

cdmLinkEntity2Table

Description

An instance of this relationship represents cdmEntityToTableMapping and DSRTTable

Source Information:**Class Name**

cdmEntityToTableMapping

Attribute

pLinkedToTable

Target Information:**Class Name**

DSRTTable

Attribute

pBasedOnEntity

Forward Mapping Semantics:

Verb In

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Has

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWCDM

Storage View File Name

cdmI04

DLL Name

ewswcdm

Attributes

None

Cache View Methods

None

Relationship Class: cdmLinkEntity2TDef

Cache View Name

cdmLinkEntity2TDef

Storage View Name

cdmLinkEntity2TDef

Description

An instance of this relationship represents cdmEntityToTableMapping and DSRTTableDefinition

Source Information:

Class Name

cdmEntityToTableMapping

Attribute

pLinkedToTableDefinition

Target Information:

Class Name

DSRTTableDefinition

Attribute

pBasedOnEntity

Forward Mapping Semantics:

Verb In

Cardinality

0..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb Has

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWCDM

Storage View File Name

cdmI03

DLL Name

ewswcdm

Attributes

None

Cache View Methods

None

Relationship Class: cdmLinkModelTransformP**Cache View Name**

cdmLinkModelTransformP

Storage View Name

cdmLinkModelTransformP

Description

Relationship that connects a transform process to a CDM data model. A data model can be the source of many transformations into different relational designs. Note: If a CDM data model is transformed twice into the same relational design, the existing transformation process is reused. and cdmTransformProcess

Source Information:**Class Name**

cdmModel

Attribute

transformedRelDesigns

Target Information:**Class Name**

cdmTransformProcess

Attribute

pModel

Forward Mapping Semantics:**Verb** In**Cardinality**
0..m**Ordered**
No**Controlling**
No**Inverse Mapping Semantics:****Verb** Has**Cardinality**
1..1

Ordered
No

Controlling
No

Cache View File Name
EWSWCDM

Storage View File Name
cdmInt

DLL Name
ewswcdm

Attributes

None

Cache View Methods

None

Relationship Class: cdmLinkRelship2FKey

Cache View Name
cdmLinkRelship2FKey

Storage View Name
cdmLinkRelship2FKey

Description
An instance of this relationship represents cdmRelshipToFKeyMapping and DSRKeyDefinition

Source Information:

Class Name
cdmRelshipToFKeyMapping

Attribute
pLinkedToKeyDefinition

Target Information:

Class Name
DSRKeyDefinition

Attribute
pBasedOnRelationship

Forward Mapping Semantics:

Verb In

Cardinality
1..1

Ordered
No

Controlling
No

Inverse Mapping Semantics:**Verb** Has**Cardinality**
0..1**Ordered**
No**Controlling**
No**Cache View File Name**
EWSWCDM**Storage View File Name**
cdmI02**DLL Name**
ewswcdm**Attributes**

None

Cache View Methods

None

Relationship Class: cdmLinkTransformPEnt2Tab**Cache View Name**
cdmLinkTransformPEnt2Tab**Storage View Name**
cdmLinkTransformPEnt2Tab**Description**
An instance of this relationship represents cdmTransformProcess and cdmEntityToTableMapping**Source Information:****Class Name**
cdmTransformProcess**Attribute**
entityToTableMappings**Target Information:****Class Name**
cdmEntityToTableMapping**Attribute**
pTransformProcess**Forward Mapping Semantics:****Verb** In**Cardinality**
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb For

Cardinality
1..1

Ordered
No

Controlling
No

Cache View File Name
EWSWCDM

Storage View File Name
cdmlte

DLL Name
ewswcdm

Attributes

None

Cache View Methods

None

Relationship Class: cdmModelUsesDE

Cache View Name
cdmModelUsesDE

Storage View Name
cdmModelUsesDE

Description
Relationship that connects the classes cdmModel and DSDataElement. This relationship has an attribute that contains the CDM internal key of the corresponding data element.

Source Information:

Class Name
cdmModel

Attribute
pUsedDE

Target Information:

Class Name
DSDataElement

Attribute
pUsedInDM

Forward Mapping Semantics:

Verb In

Cardinality
0..m

Ordered
No

Controlling
No

Inverse Mapping Semantics:

Verb For

Cardinality
0..m

Ordered
No

Controlling
No

Cache View File Name
EWSWCDM

Storage View File Name
cdmdmde

DLL Name
ewswcdm

Attributes

None

Cache View Methods

None

Relationship Class: cdmR2FKPartOfE2TM

Cache View Name
cdmR2FKPartOfE2TM

Storage View Name
cdmR2FKPartOfE2TM

Description
An instance of this relationship represents cdmEntityToTableMapping and cdmRelshipToFKeyMapping

Source Information:

Class Name
cdmEntityToTableMapping

Attribute
pHasR2FKMapping

Target Information:

Class Name

cdmRelshipToFKeyMapping

Attribute

pLinkE2TDMapping

Forward Mapping Semantics:**Verb** In**Cardinality**

0..m

Ordered

No

Controlling

No

Inverse Mapping Semantics:**Verb** For**Cardinality**

1..1

Ordered

No

Controlling

No

Cache View File Name

EWSWCDM

Storage View File Name

cdmI06

DLL Name

ewswcdm

Attributes

None

Cache View Methods

None

Relationship Class: cdmTPTransformed2RD**Cache View Name**

cdmTPTransformed2RD

Storage View Name

cdmTPTransformed2RD

Description

Relationship that connects a CDM transform process with a RDB relational design. A relational design can be the target of many transform processes. E.g. you can transform a data model 'dm1' into the relational design 'rd' and after that transform a second data model 'dm2' into the relational design 'rd'.

Source Information:**Class Name**

cdmTransformProcess

Attribute

pTransformedToRD

Target Information:**Class Name**

DSRRelationalDesign

Attribute

pTargetofTP

Forward Mapping Semantics:

Verb In

Cardinality

1..1

Ordered

No

Controlling

No

Inverse Mapping Semantics:

Verb For

Cardinality

0..m

Ordered

No

Controlling

No

Cache View File Name

EWSWCDM

Storage View File Name

cdmI01

DLL Name

ewswcdm

Attributes

None

Cache View Methods

None

Customer support

Your options for IBM VisualAge TeamConnection support, as described in your License Information and Licensed Program Specifications, include electronic forums. You can use the electronic forums to access IBM VisualAge TeamConnection technical information, exchange messages with other TeamConnection users, and receive information regarding the availability of fixes. The following forums are available.

- **IBM Talklink**

Use the TEAMC CFORUM. For additional information about TalkLink, call

- United States 1-800-547-1283
- Canada 1-800-465-7999 ext. 228

- **CompuServe**

From any ! prompt, type GO SOFSOL, then select TeamConnection. For additional information, call 1-800-848-8199 and ask for representative 239.

- **Internet**

Go to the IBM homepage, <http://www.ibm.com>. Use the search function with keyword TeamConnection to go to the TeamConnection area.

If you cannot access these forums, contact your IBM representative.

There are several other support offerings available after purchasing IBM VisualAge TeamConnection. For a list of these offerings, please contact your IBM representative.

Bibliography

IBM VisualAge TeamConnection library

The following is a list of the TeamConnection publications.

- **License Information (GC34-4497):**
Contains license, service, and warranty information.
- **Administrator's Guide (GC34-4551):**
Lists the hardware and software that are required before you can install and use the IBM VisualAge TeamConnection product, provides detailed instructions for installing and configuring the TeamConnection family and build servers, and provides instructions for administering a TeamConnection family.
- **Getting Started (SC34-4552):**
Tells first-time users how to install the TeamConnection clients on their workstations, and familiarizes them with the command line and graphical user interfaces.
- **User's Guide (SC34-4499):**
A comprehensive guide for TeamConnection administrators and client users that helps them install and use TeamConnection.
- **Commands Reference (SC34-4501):**
Describes the TeamConnection commands, their syntax, and the authority required to issue each command. This book also provides examples of how to use the various commands.
- **Quick Commands Reference (GC34-4500):**
Lists the TeamConnection commands along with their syntax.
- **Staying on Track with TeamConnection Processes (83H9677):**
Poster showing how objects flow through the states defined for each TeamConnection process.
- The following publications can be ordered as a set (SBOF-8560):
 - Administrator's Guide**
 - Getting Started**
 - User's Guide**
 - Commands Reference**
 - Quick Commands Reference**
 - Staying on Track with TeamConnection Processes**

Tool Builder's Development Kit

The following publications are part of the Tool Builder's Development Kit feature:

- **Tool Builder's Development Guide (SC34-4553):**
Explains how to create and extend tools for accessing objects in the TeamConnection database. It contains guidance and reference information.
- **Information Model Reference (SC34-4554):**
Details the TeamConnection information model. This publication is available in softcopy only.

ObjectStore

The following publications are part of the ObjectStore library of documents and are available for order from Object Design, Inc. To order these documents call (617) 674-5000, Monday through Friday from 8:30 AM to 5:30 PM Eastern Time.

- **ObjectStore C++ Installation:**

Contains step-by-step procedures for installing the latest release of ObjectStore on a specific platform:

- **310-100-40 I**

- UNIX

- **310-310-40 I**

- Windows

- **310-320-40 I**

- OS/2

- **ObjectStore C++ API User Guide (310-000-40 U):**

Provides information about the application programming interface for application programmers.

- **ObjectStore C++ API Reference (310-000-40 R):**

Describes the API to the features provided by ObjectStore for application programmers.

- **ObjectStore C++ Building Applications (310-000-40 B):**

Provides information and instructions for compiling code, generating schemas, and linking files using all supported compilers; and provides instructions for developing ObjectStore client applications for use on multiple platforms.

- **ObjectStore Management (310-000-40 M):**

Provides information and instructions for performing management tasks on ObjectStore server and client systems. It includes server parameters, environment variables, and database utilities.

- **ObjectStore C++ Performance (310-000-40 P):**

Explains the fundamentals of designing and tuning ObjectStore applications for optimal performance.

IBM Exchange library

The publications listed below can be ordered as a set (SBOF-6098) or separately as indicated below. IBM Exchange will be available at a later date.

- *Licensed Programming Specification (GC34-4525):*
- *Installation Guide (SC34-4509):*
- *Bridge Builder's Guide (SC34-4508):*
- *User's Guide 1 (SC34-4506):*
- *User's Guide 2 (SC34-4507):*

Related publications

- Transmission Control Protocol/Internet Protocol (TCP/IP)
 - *TCP/IP 2.0 for OS/2: Installation and Administration (SC31-6075)*
 - *TCP/IP for MVS Planning and Customization (SC31-6085)*
- MVS

- *MVS/XA JCL User's Guide* (GC28-1351)
- *MVS/XA JCL Reference* (GC28-1352)
- *MVS/ESA JCL User's Guide* (GC28-1830)
- *MVS/ESA JCL Reference* (GC28-1829)



Part Number: 00A0000



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC34-4504-00



00A0000

