



Innovus Database Access Command Reference

Product Version 20.10
March 2020

© 2020 Cadence Design Systems, Inc. All rights reserved.
Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 1-800-862-4522.

All other trademarks are the property of their respective holders.

Restricted Print Permission: This publication is protected by copyright and any unauthorized use of this publication may violate copyright, trademark, and other laws. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. This statement grants you permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used solely for personal, informational, and noncommercial purposes;
2. The publication may not be modified in any way;
3. Any copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement; and
4. Cadence reserves the right to revoke this authorization at any time, and any such use shall be discontinued immediately upon written notice from Cadence.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. The information contained herein is the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only by Cadence's customer in accordance with, a written agreement between Cadence and its customer. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

About This Manual	53
Database Access Command Diagrams	53
Related Documents	56
Innovus Product Documentation	56
Innovus Stylus Common UI Product Documentation	58
1	59
Database Commands A - E	59
A	59
dbAddCellObs	59
dbAddCoverInst	60
dbAddDeCapAtLoc	61
dbAddToHInst	62
dbAddToSelSet	62
dbAddWireSeg	63
dbAreAnyNetWiresDangling	64
dbAreCellsPhyReplaceable	65
dbAreNetWiresOK	66
dbAssignFTermToBump	67
dbAttachBuffer	67
B	68
dbBlackBoxArea	68
dbBlackBoxAreaPerGate	69
dbBlackBoxAspectRatio	70
dbBlackBoxCell	70
dbBlackBoxGateCount	71
dbBlackBoxHeight	72
dbBlackBoxName	72
dbBlackBoxNext	73
dbBlackBoxSpecifyMethod	74
dbBlackBoxSpecifyValue	75
dbBlackBoxUtilization	75
dbBlackBoxWidth	76

dbBoxDimX	77
dbBoxDimY	78
dbBoxLL	78
dbBoxLLX	79
dbBoxLLY	80
dbBoxPtrBox	80
dbBoxUR	81
dbBoxURX	82
dbBoxURY	82
dbBumpBumpCell	83
dbBumpCellBox	84
dbBumpCellName	84
dbBumpCellNext	85
dbBumpCellNrlItem	86
dbBumpCellPtArr	87
dbBumpFTerm	87
dbBumpLoc	88
dbBumpOrient	89
dbBumpType	89
dbBusBitArr	90
dbBusBitBus	91
dbBusBitParent	91
dbBusParent	92
C	93
dbCellIBNetList	93
dbCellChangeWireGlobalStatus	93
dbCellChangeWireStatus	94
dbCellClass	95
dbCellCongestionView	95
dbCellDim	96
dbCellDoNotFlatten	97
dbCellDontKnowIsFlat	97
dbCellFixWireTermLoops	98
dbCellFlattenedNrBlock	99
dbCellFlattenedNrGate	99
dbCellFlattenedNrInst	100
dbCellFlattenedNrIo	101

dbCellFlattenedNrStdCell	101
dbCellFlattenedNrStor	102
dbCellFootPrintName	103
dbCellFPlan	103
dbCellFTermArr	104
dbCellFTermList	105
dbCellGlobalWireAddVias	105
dbCellGlobalWireToUsableWire	106
dbCellHasInternalPower	107
dbCellHasIoRing	107
dbCellHasLeakagePower	108
dbCellHierCell	109
dbCellHInst	109
dbCellIGLen	110
dbCellInstList	111
dbCellIoList	111
dbCellName	112
dbCellNetList	113
dbCellNext	113
dbCellNrBidi	114
dbCellNrBlock	115
dbCellNrFlatInst	116
dbCellNrFTerm	116
dbCellNrGate	117
dbCellNrHInsts	118
dbCellNrInput	119
dbCellNrInst	119
dbCellNrIlo	120
dbCellNrNet	121
dbCellNrPGFTerm	121
dbCellNrPhysicalInst	122
dbCellNrRef	123
dbCellNrRow	124
dbCellNrStdCell	124
dbCellNrStor	125
dbCellNrSym	126
dbCellOrigCell	126

dbCellOrigin	127
dbCellPGFTermArr	128
dbCellPhysicalNetList	128
dbCellPrev	129
dbCellPtn	130
dbCellPtnList	131
dbCellRepCell	131
dbCellRouteAlgUsed	132
dbCellSite	133
dbCellsNetList	134
dbCellSubClass	134
dbCellSymArr	135
dbCellTechSite	136
dbCellTimeLib	136
dbCellType	137
dbCellVoltage	138
dbChangeInstCell	139
dbCheckCell	139
dbCleanCellFlag	141
dbCleanInstFlag	141
dbCleanNetFlag	142
dbCleanTermFlag	143
dbClearCellFTermMarkers	144
dbClearCellInstMarkers	144
dbClearCellNetMarkers	145
dbClearCellTermMarkers	145
dbClearCellWireMarkers	146
dbClearHInstMarkers	146
dbClockName	147
dbClockNext	148
dbClockPrev	148
dbClockRate	149
dbCloneListInst	150
dbCloneListOrient	151
dbComputeHInstPDensity	151
dbConstraintBox	152
dbConstraintHInst	153

dbConstraintNext	153
dbConstraintNextInFPlan	154
dbConstraintType	155
dbCopyNetWires	156
dbCountDrc	157
dbCountFTermFlag	157
dbCountInstFlag	158
dbCountNetFlag	158
dbCountTermFlag	159
dbCreateConstraint	159
dbCreateMarker	160
dbCreateObstruct	161
dbCreateRestrictedVia	162
dbCreateScreen	163
dbCreateSpareGateGroup	164
dbCreateText	164
dbCreateVia	165
dbCreateWire	166
dbCurrentDefaultLibraryContext	167
dbCustomLayerId	167
dbCustomLayerName	168
D	168
dbDBUToMicrons	169
dbDelCellObs	169
dbDelCoverInst	170
dbDeleteAreaRow	171
dbDeleteBuffer	171
dbDeleteBumpGrid	172
dbDeleteCustomObj	173
dbDeleteLayerBlk	173
dbDeleteMarker	174
dbDeleteObj	175
dbDeleteObstruct	176
dbDeletePtnCut	177
dbDeletePtnFeed	177
dbDeletePtnPinBlk	178
dbDeletePtnPinShapes	179

dbDeletePtnPinShapesForNet	179
dbDeletePtnPinShapesForNetOnLayer	180
dbDeletePtnPinShapesOnLayer	181
dbDeleteRouteBlk	182
dbDeleteRouteBox	183
dbDeleteRowCluster	184
dbDeleteScreen	184
dbDeleteSpareGateGroup	185
dbDeleteStripBox	186
dbDeleteText	187
dbDeleteTrialRoute	187
dbDelFromHInst	188
dbDelFromSelSet	188
dbDelHiliteSet	189
dbDelHInst	190
dbDelProp	191
dbDelSelSet	191
dbDelWireSeg	192
dbDisplayUnplacedInst	193
E	193
dbEndIterAllRoutes	194
dbEndIterRoutes	196
dbEndIterViaCellRectangles	197
dbEvalAddEcoBuffer	198
dbEvalChangeCell	198
dbEvalRemoveBuffer	199
dbEvalSwitchTerm	200
dbExtRuleArrNext	200
dbExtRuleId	201
dbExtRuleLayerRuleList	202
dbExtRuleName	202
dbExtRuleNext	203
dbExtRuleNrRoutingLayerRule	204
dbExtRuleRoutingLayerRuleArr	204
dbExtRuleSpacingList	205
dbExtRuleViaCellList	206

2	207
Database Commands F - H	207
F	207
dbFindInstsByCell	207
dbFindInstsByName	208
dbFindInstsOnNet	208
dbFindModulesByName	209
dbFindNetsByName	210
dbFindTermsByName	211
dbFlattenInst	211
dbFootPrintCellList	212
dbFootPrintCellNameList	213
dbFootPrintTIsCellList	213
dbForAllCellIFTerm	214
dbForAllCellIFTermAndInternalFTerm	215
dbForAllCellNet	216
dbForAllCellIPGFTerm	217
dbForAllCellIPGTerm	218
dbForAllInstTerm	219
dbForAllVInstVTerm	220
dbForEachAreaECOScreen	221
dbForEachCellBlackBox	222
dbForEachCellBump	223
dbForEachCellIFTerm	224
dbForEachCellGeomList	225
dbForEachCellHaloBox	226
dbForEachCellInputFTerm	226
dbForEachCellInst	228
dbForEachCellIlo	229
dbForEachCellNet	230
dbForEachCellOutputFTerm	231
dbForEachCellPhyFTerm	232
dbForEachCellPtn	233
dbForEachCellSNet	234
dbForEachCellSortedInst	235
dbForEachCellSortedVInst	236

dbForEachCellStorInst	236
dbForEachCellTileInst	237
dbForEachCellTimeArc	238
dbForEachCellVInst	239
dbForEachClockDomainInst	240
dbForEachClockDomainNet	240
dbForEachConstraintBox	240
dbForEachCutLayer	241
dbForEachDomainInst	242
dbForEachExtRuleLayerRuleList	242
dbForEachFileLine	243
dbForEachFileUpdateLine	244
dbForEachFPlanAIORowCluster	244
dbForEachFPlanCellPad	245
dbForEachFPlanConstraint	246
dbForEachFPlanDefRow	247
dbForEachFPlanGlobalNetConnection	248
dbForEachFPlanGroup	249
dbForEachFPlanLayerBlk	250
dbForEachFPlanNetGroup	251
dbForEachFPlanObstruct	252
dbForEachFPlanPinGroup	253
dbForEachFPlanPtnCut	254
dbForEachFPlanPtnFeed	255
dbForEachFPlanPtnPinBlk	255
dbForEachFPlanPtnPinBlks	256
dbForEachFPlanRouteBlk	257
dbForEachFPlanRouteGuide	258
dbForEachFPlanScreen	259
dbForEachFPlanStrip	260
dbForEachFTermFPin	261
dbForEachFTermLefPort	262
dbForEachGeomListBox	263
dbForEachGeomListGeom	264
dbForEachGeomListPath	265
dbForEachGeomListPoly	266
dbForEachGroupBoxListBox	267

dbForEachGroupHInst	268
dbForEachHeadBumpCell	268
dbForEachHeadCell	269
dbForEachHeadClock	270
dbForEachHeadCustomLayer	271
dbForEachHeadHilitePtr	272
dbForEachHeadLayer	273
dbForEachHeadPath	274
dbForEachHeadPtn	275
dbForEachHeadRepCell	276
dbForEachHeadRuleList	277
dbForEachHeadSel	278
dbForEachHeadSelPtr	279
dbForEachHeadSite	280
dbForEachHeadTechSite	281
dbForEachHeadTileCell	281
dbForEachHeadTimeLib	282
dbForEachHeadTIsCell	283
dbForEachHInstHInst	284
dbForEachHInstHTerm	285
dbForEachHInstTreeHInst	286
dbForEachHInstTreeInst	287
dbForEachHInstTreeNet	288
dbForEachInstInputTerm	289
dbForEachInstOutputTerm	290
dbForEachInstTerm	291
dbForEachLayerShapeShape	291
dbForEachLefFPortLayerShape	292
dbForEachListElem	294
dbForEachListPtr	294
dbForEachMetalLayer	295
dbForEachNetGroupNet	295
dbForEachNetHNet	297
dbForEachNetMsNetGroup	298
dbForEachNetNetGroup	298
dbForEachNetOutputTerm	299
dbForEachNetTerm	300

dbForEachNetTermS	301
dbForEachNetWire	302
dbForEachObjProp	303
dbForEachPathTerm	304
dbForEachPinGroupPin	304
dbForEachPowerDomain	305
dbForEachPowerDomainPad	306
dbForEachPtnCloneList	307
dbForEachRowClusterRow	308
dbForEachSNetBox	309
dbForEachSNetStripBox	310
dbForEachTileCellTilePin	311
dbForEachTimeLibCell	311
dbForEachTimeLibOpCond	313
dbForEachVNetVTerm	314
dbFPinAccessDir	314
dbFPinADir	315
dbFPinLayer	317
dbFPinLoc	318
dbFPinNext	319
dbFPinNrAPin	320
dbFPinSide	321
dbFPinWidth	322
dbFPinZ	323
dbFPlanBox	325
dbFPlanCellPadList	325
dbFPlanConstraintList	326
dbFPlanCoreBox	327
dbFPlanDefaultTechSite	328
dbFPlanEqualizeAllPtnCellHInst	329
dbFPlanFlip	329
dbFPlanGroupList	330
dbFPlanInstIGLen	331
dbFPlanUpdateGmapByPrerouteAsObs	332
dbFreezeAllInst	333
dbFTermBox	333
dbFTermBTerm	334

dbFTermBus	335
dbFTermCell	336
dbFTermDepth	337
G	337
dbGeomBoxBox	338
dbGeomLineEndPt	338
dbGeomLineStartPt	339
dbGeomListCustLayer	340
dbGeomPathLocArr	340
dbGeomPathNrLoc	341
dbGeomPathWidth	342
dbGeomPolyLocArr	342
dbGeomPolyNrLoc	343
dbGeomTextHeight	344
dbGeomTextLoc	344
dbGeomTextText	345
dbGetBlackBoxByName	346
dbGetBottomIoPadOrient	346
dbGetBumpByName	347
dbGetBumpCellByName	348
dbGetBumpCount	348
dbGetBumpDetailCount	349
dbGetCapUnitStr	349
dbGetCellByName	350
dbGetCellByNameAndLib	351
dbGetCellByNameAndLibType	351
dbGetCellFreeLegalLoc	352
dbGetCellTimeArc	353
dbGetChildHInst	354
dbGetExtRuleByName	355
dbGetFPlanLayerHalo	356
dbGetFPlanNrRow	356
dbGetFTermByName	357
dbGetGroupAncestor	358
dbGetGroupByName	358
dbGetHaloValue	359
dbGetHInstBoxArea	360

dbGetHInstByName	360
dbGetHTermByInstTermName	361
dbGetInstByName	361
dbGetIoByName	362
dbGetIoDriverSiteCount	363
dbGetIsNetlistChangedForClink	363
dbGetLayerBlkByName	364
dbGetLayerByExtId	365
dbGetLayerByName	365
dbGetLayerShapeLayer	366
dbGetLDFLibs	367
dbGetLocAndSide	367
dbGetLocPowerDomainVoltage	368
dbGetMetalLayerById	369
dbGetNetByLocalName	369
dbGetNetByName	370
dbGetNetDrivenTermsThroughLoc	371
dbGetNetFrequency	372
dbGetNetGroupByName	372
dbGetNetRectBBox	373
dbGetNetRectID	374
dbGetNextHInst	374
dbGetNumOfloRow	375
dbGetObjByName	376
dbGetObstructByName	376
dbGetOrCreatePropByName	377
dbGetOrCreateSNetByName	378
dbGetPathById	379
dbGetPGFTermByName	379
dbGetPGTermByName	380
dbGetPinGroupByName	381
dbGetPowerDomainByName	381
dbGetPrerouteAsObs	382
dbGetPrevHInst	382
dbGetPropByName	383
dbGetPtnBorderCoords	384
dbGetPtnByName	385

dbGetPtnCutByName	385
dbGetPtnFeedByName	386
dbGetPtnFPinsInArea	387
dbGetPtnFPlanBox	387
dbGetPtnLayerHalo	388
dbGetPtnPinBlkByName	389
dbGetRawSlack	390
dbGetRouteBlkByName	390
dbGetRouteGuideByName	391
dbGetScreenByName	392
dbGetShifterCellsForPD	393
dbGetSNetByName	393
dbGetSNodeByIdx	394
dbGetSpefInFileName	395
dbGetSrcClks	395
dbGetStripByName	396
dbGetTechSiteByName	396
dbGetTermByInstTermName	397
dbGetTermByName	398
dbGetTileCellByName	399
dbGetTimeLibByName	399
dbGetTimeUnitStr	400
dbGetViaCellByName	401
dbGetVInstByName	401
dbGetVNetByName	402
dbGlobalNetConnectionConnectType	403
dbGlobalNetConnectionToGlobalNet	404
dbGlobalNetConnectionUnderModule	405
dbGlobalNetConnectionVerbose	406
dbGroundSpec	407
dbGroupBox	408
dbGroupConType	408
dbGroupDensity	409
dbGroupGroup	410
dbGroupHInstList	410
dbGroupName	411
dbGroupNext	412

dbGroupNrHInst	412
dbGroupPowerDomain	413
H	414
dbHasBlackBoxAreaValueBeenSet	414
dbHasBlackBoxXYBeenSet	414
dbHeadAspectRatio	415
dbHeadBlockAsStdCell	416
dbHeadBlockMargin	416
dbHeadBox	417
dbHeadCellList	418
dbHeadChipTopCell	419
dbHeadCoreBox	419
dbHeadCustomLayerList	420
dbHeadDBUPerIGU	421
dbHeadDBUPerloGU	421
dbHeadDBUPerMGrid	422
dbHeadExtRuleArr	423
dbHeadExtRuleList	423
dbHeadFELayerList	424
dbHeadFirstSelPtr	425
dbHeadFPlan	426
dbHeadHiliteList	426
dbHeadIoBox	427
dbHeadIoHgt	428
dbHeadLEFLayerArr	428
dbHeadLEFLayerList	429
dbHeadMicronPerDBU	430
dbHeadName	430
dbHeadNrExtRule	431
dbHeadNrExtRulePlus1	432
dbHeadNrHilite	433
dbHeadNrLayer	433
dbHeadNrLEFLayer	434
dbHeadNrSel	435
dbHeadNrWireLayer	435
dbHeadOhmPerDBU	436
dbHeadPathList	437

dbHeadPicoFPerDBU	437
dbHeadPicoSecPerDBU	438
dbHeadPropTypeList	439
dbHeadPtnList	439
dbHeadRule	440
dbHeadSelList	441
dbHeadSpDetailLen	441
dbHeadSpGlobalLen	442
dbHeadStdCellHgt	443
dbHeadTopCell	444
dbHiliteObj	444
dbHiliteObjBox	445
dbHInstArea	446
dbHInstBlockAreaWHalo	446
dbHInstBox	447
dbHInstCell	448
dbHInstCellName	448
dbHInstColorId	449
dbHInstCongest	450
dbHInstConnection	451
dbHInstConstraint	451
dbHInstDensity	451
dbHInstFenceDensity	452
dbHInstFPlanBox	453
dbHInstGroup	454
dbHInstHInstList	454
dbHInstHorOrder	455
dbHInstHTermList	456
dbHInstId	456
dbHInstIlm	457
dbHInstName	458
dbHInstNrInst	459
dbHInstParent	459
dbHInstPCell	460
dbHInstPerimList	461
dbHInstPin	461
dbHInstPrefixId	462

dbHInstPropList	463
dbHInstPtListBox	463
dbHInstPtn	464
dbHInstResetArea	465
dbHInstStdCellArea	465
dbHInstSuggestLoc	466
dbHInstSuggestOrient	467
dbHInstType	467
dbHInstUserDensity	468
dbHInstVerOrder	469
dbHTermFTerm	469
dbHTermHInst	470
dbHTermNet	471
dbHTermNext	472
dbHTermOrder	472
dbHTermSide	473
3	475
Database Commands I	475
I	475
dblIncrDelayUpdate	475
dblInfoStripBox	476
dblInfoVia	476
dblInfoViaCellLayer	477
dblInfoViaCellRegularCuts	478
dblInfoWire	478
dblInitBumpGrid	480
dblInitCellNetList	480
dblInitVCellNetList	481
dblInstBaseName	482
dblInstBox	482
dblInstCell	483
dblInstCellName	484
dblInstCluld	484
dblInstCongest	485
dblInstEffIGLen	486
dblInstGroup	487

dblInstHasObstruct	487
dblInstHierHInst	488
dblInstHInst	489
dblInstHorOrder	490
dblInstIGArea	490
dblInstIGLen	491
dblInstInternalPower	492
dblInstLeakagePower	492
dblInstLibraryContext	493
dblInstLoc	494
dblInstName	494
dblInstNext	495
dblInstNrBidi	496
dblInstNrInput	496
dblInstNrOutput	497
dblInstNrRow	498
dblInstNrTerm	498
dblInstObstruct	499
dblInstObstructLayers	500
dblInstOrient	501
dblInstPCell	502
dblInstPdefName	502
dblInstPGTermArr	503
dblInstPlacementStatus	504
dblInstPrefixId	505
dblInstPrefixName	505
dblInstPrev	506
dblInstPriority	507
dblInstRoutingHaloBottomLayer	507
dblInstRoutingHaloSize	508
dblInstRoutingHaloTopLayer	509
dblInstTempld	510
dblInstTermArr	510
dblInstTermList	511
dblInstVerOrder	512
dbloBox	512
dbloBump	513

dbloCell	514
dbloFTerm	515
dbloIndent	515
dbloInst	516
dbloLayerId	517
dbloLoc	518
dbloName	519
dbloNext	519
dbloOffset	520
dbloOrder	521
dbloOrient	522
dbloRow	523
dbloSide	523
dbloSpacing	524
dbloTdfLibName	525
dblsAllInstPlaced	526
dblsBackslashInNamesHidden	526
dblsBoxOverlappingBox	527
dblsBoxOverlappingOrTouchingBox	528
dblsBumpHilite	529
dblsBumpSel	530
dblsCaseSensitive	530
dblsCellArealo	531
dblsCellBlackBox	532
dblsCellBlock	532
dblsCellCdumpDefined	533
dblsCellClockSynthesized	534
dblsCellDisplayable	534
dblsCellDontTouch	535
dblsCellDontUse	536
dblsCellDummy	536
dblsCellECOMarked	537
dblsCellECOMarked2	538
dblsCellECOMarked4	539
dblsCellEEQCell	539
dblsCellFlat	540
dblsCellGate	541

dblSCellGRouted	541
dblSCellHier	542
dblSCellIo	543
dblSCellIoPlaced	543
dblSCellJtagCell	544
dblSCellLatch	545
dblSCellLeafCell	546
dblSCellMarked	546
dblSCellMarked2	547
dblSCellMarked3	548
dblSCellMarked4	548
dblSCellMaybeHier	549
dblSCellNLDefined	550
dblSCellNLRefed	551
dblSCellPartition	551
dblSCellPlaced	552
dblSCellPower	553
dblSCellPowerAnalyzed	553
dblSCellPrototype	554
dblSCellRCExtracted	555
dblSCellRouted	555
dblSCellScanCell	556
dblSCellScanOpted	557
dblSCellSequential	558
dblSCellSpareGate	559
dblSCellStdCell	559
dblSCellStor	560
dblSCellSuper	561
dblSCellSymDegenerate	561
dblSCellSymmetryR90	562
dblSCellSymmetryX	563
dblSCellSymmetryY	564
dblSCellTGDelayUpdated	564
dblSCellTimeDefined	565
dblSCellTop	566
dblSCellVCell	566
dblSCellVDDOnBottom	567

dbIsClockHilite	568
dbIsClockMarked	569
dbIsClockSel	569
dbIsExtractRCRandomMode	570
dbIsExtractRCRCDBMode	571
dbIsFPlanX	571
dbIsFTermAssigned	572
dbIsFTermAsyncCtrl	573
dbIsFTermBidi	574
dbIsFTermBrkLoop	574
dbIsFTermBus	575
dbIsFTermClk	576
dbIsFTermContAssignLHS	576
dbIsFTermContinuousAssign	577
dbIsFTermD	578
dbIsFTermFTerm	578
dbIsFTermHilite	579
dbIsFTermIgnored	580
dbIsFTermInput	581
dbIsFTermInternal	581
dbIsFTermMarked	582
dbIsFTermMarked2	583
dbIsFTermMarked3	583
dbIsFTermMPW	584
dbIsFTermNLDefined	585
dbIsFTermNLRefed	585
dbIsFTermOutput	586
dbIsFTermPreassigned	587
dbIsFTermQ	588
dbIsFTermScanClk	588
dbIsFTermScanIn	589
dbIsFTermScanInv	590
dbIsFTermScanOut	590
dbIsFTermSel	591
dbIsFTermSideAssigned	592
dbIsFTermSpecial	592
dbIsFTermTAClkSrc	593

dblSFTermTAlgnored	594
dblSFTermTieHi	595
dblSFTermTieLo	595
dblSFTermTimeDefined	596
dblSFTermTriCtl	597
dblSFTermUnused	597
dblSGroupHilite	598
dblSGroupInferred	599
dblSGroupPhyHier	600
dblSGroupPowerDomain	601
dblSGroupSel	602
dblSGroupUngroup	603
dblSHeadCustomLayerChanged	604
dblSHeadDesignDisplayable	604
dblSHeadDesignGRouted	605
dblSHeadDesignInMemory	606
dblSHeadDesignPlaced	606
dblSHeadDesignTGDelayUpdated	607
dblSHeadFPlanChanged	608
dblSHeadHiliteSticky	608
dblSHeadIGUSet	609
dblSHeadIlmFlattened	610
dblSHeadIlmSpecified	611
dblSHeadInHierMode	611
dblSHeadLibraryInMemory	612
dblSHeadTAFuncExtracted	613
dblSHeadTALibraryInMemory	613
dblSHInstAllOrdered	614
dblSHInstDontTouch	615
dblSHInstFPlanChanged	615
dblSHInstFPLeaf	616
dblSHInstHidden	617
dblSHInstHilite	618
dblSHInstHInst	618
dblSHInstInFPlan	619
dblSHInstJtag	620
dblSHInstJtagElem	620

dblsHInstLeafRegion	621
dblsHInstMarked	622
dblsHInstMarked2	622
dblsHInstMarked3	623
dblsHInstOnCurrLevel	624
dblsHInstOrdered	625
dblsHInstPinAssigned	625
dblsHInstPinInitiated	626
dblsHInstRegionDefined	627
dblsHInstSel	627
dblsHInstUngroup	628
dblsInstArealo	629
dblsInstBlackBox	630
dblsInstBlock	631
dblsInstDefCovered	632
dblsInstDefSrcTiming	633
dblsInstDontTouch	634
dblsInstGroupInSelSet	635
dblsInstHaloBlock	636
dblsInstHidden	637
dblsInstHilite	638
dblsInstHInst	639
dblsInstIlo	640
dblsInstIPOed	641
dblsInstJtag	642
dblsInstJtagCell	643
dblsInstJtagElem	644
dblsInstMarked	645
dblsInstMarked2	646
dblsInstMarked3	647
dblsInstMoved	647
dblsInstOrientLegal	648
dblsInstParentInSelSet	649
dblsInstPartition	650
dblsInstPhysicalOnly	651
dblsInstPlaced	651
dblsInstPreplaced	652

dblInstSel	653
dblInstSpareGate	654
dblInstStdCell	655
dblInstStor	656
dblInstStorage	656
dblInstTAlgnored	657
dblInstUnused	658
dblIsAssigned	658
dblIsBump	659
dblIsClearance	660
dblIsCorner	661
dblIsCovered	661
dblIsDefCovered	662
dblIsDummyPad	663
dblIsFixed	664
dblIsGapFixed	665
dblIsGroundPad	665
dblIsHilite	666
dblIsMarked	667
dblIsPowerPad	668
dblIsPreplaced	669
dblIsSel	669
dblLayerBlkHilite	670
dblLayerBlkOnCutLayer	671
dblLayerBlkOnLayer	672
dblLayerBlkSel	673
dblLayerH	674
dblLayerMSLayer	674
dblLayerShapeShapeList	675
dblLayerShapeVia	676
dblLayerV	677
dblLefPortClassCore	678
dblLefPortClassNone	679
dblLefPortClassUndefined	680
dblNetAnalog	682
dblNetBus	682
dblNetClock	683

dbIsNetConnected	684
dbIsNetContAssignLHS	685
dbIsNetContinuousAssign	685
dbIsNetCritical	686
dbIsNetDbWidth	687
dbIsNetDefInClock	688
dbIsNetDontTouch	688
dbIsNetExternal	689
dbIsNetGnd	690
dbIsNetGroupMsBuss	690
dbIsNetGroupMsCoax	691
dbIsNetGroupMsDiffPair	692
dbIsNetGroupMsMatchedPairs	693
dbIsNetHidden	693
dbIsNetHilite	694
dbIsNetIgnored	695
dbIsNetInInst	695
dbIsNetIPOed	696
dbIsNetIPOIgnored	697
dbIsNetMarked	697
dbIsNetMarked2	698
dbIsNetMarked3	699
dbIsNetMarked4	700
dbIsNetPhysicalOnly	700
dbIsNetPostRouteSiFix	701
dbIsNetPwr	702
dbIsNetPwrOrGnd	702
dbIsNetRectActive	703
dbIsNetRectHilite	704
dbIsNetRectHookup	705
dbIsNetRectMacroPGPin	705
dbIsNetRectMacroPGTerm	706
dbIsNetRectMarked	707
dbIsNetRectPolygon	707
dbIsNetRectRail	708
dbIsNetRectReference	709
dbIsNetRectSel	710

dbIsNetRectSignal	710
dbIsNetRectStdCellIPGTerm	711
dbIsNetRectTrunk	712
dbIsNetRectVia	712
dbIsNetRouteGuided	713
dbIsNetScanNet	714
dbIsNetSel	715
dbIsNetSkipRouting	715
dbIsNetSpecial	716
dbIsNetTAlgnored	717
dbIsNetTri	718
dbIsNetTrialRouted	718
dbIsNetWideWidth	719
dbIsObjAPin	720
dbIsObjCell	721
dbIsObjCellIPad	721
dbIsObjConstraint	722
dbIsObjFTerm	723
dbIsObjHead	724
dbIsObjHilite	724
dbIsObjHInst	725
dbIsObjIInst	726
dbIsObjLayerBlk	726
dbIsObjNet	727
dbIsObjNetRect	728
dbIsObjObstruct	728
dbIsObjPath	729
dbIsObjPowerSink	730
dbIsObjPtnCut	731
dbIsObjPtnFeed	731
dbIsObjPtnPinBlk	732
dbIsObjRaShape	733
dbIsObjRouteBlk	734
dbIsObjRow	734
dbIsObjScreen	735
dbIsObjSelected	736
dbIsObjSNet	737

dblSObjStdRow	737
dblSObjStrip	738
dblSObjStripBox	739
dblSObjTerm	739
dblSObjWire	740
dblSObjXPin	741
dblSObstructHilite	742
dblSObstructSel	743
dblSPathFalse	743
dblSPathHilite	744
dblSPathSel	745
dblSPerimViewable	745
dblSPowerDomainDefault	746
dblSPowerSinkEastCut	747
dblSPowerSinkHilite	747
dblSPowerSinkMarked	748
dblSPowerSinkNorthCut	749
dblSPowerSinkSel	749
dblSPowerSinkSouthCut	750
dblSPowerSinkWestCut	751
dblSPropTypeRegistered	752
dblSPtnBlackBox	752
dblSPtnCommit	753
dblSPtnCutHilite	754
dblSPtnCutSel	754
dblSPtnFeedHilite	755
dblSPtnFeedOnLayer	756
dblSPtnFeedSel	757
dblSPtnInTimeShell	758
dblSPtnLayerBlockedOnLayer	758
dblSPtnM1Blocked	759
dblSPtnM2Blocked	760
dblSPtnM3Blocked	760
dblSPtnM4Blocked	761
dblSPtnM5Blocked	762
dblSPtnM6Blocked	763
dblSPtnM7Blocked	763

dbIsPtnM8Blocked	764
dbIsPtnM9Blocked	765
dbIsPtnPinBlkHilite	766
dbIsPtnPinBlkOnLayer	767
dbIsPtnPinBlkSel	767
dbIsPtnPinLayerFreeOnSideOnLayer	768
dbIsRCDCBCoupling	769
dbIsRCDBFromFE	770
dbIsRCDBSperf	770
dbIsRCDBSperfUsingStarN	771
dbIsResFromRCTbl	771
dbIsRouteBlkHilite	772
dbIsRouteBlkSel	773
dbIsRouteBoxFeedthrough	774
dbIsRouteBoxHilite	775
dbIsRouteBoxSel	775
dbIsRouteGuideHilite	776
dbIsRouteGuideOptimizePin	777
dbIsRouteGuidePinBased	778
dbIsRouteGuideSel	778
dbIsRouteVia	779
dbIsRouteWire	780
dbIsRowArealo	782
dbIsRowClusterHilite	783
dbIsRowClusterSel	783
dbIsRowFlipped	784
dbIsRowHilite	785
dbIsRowMarked	785
dbIsRowSel	786
dbIsRulerFlip	787
dbIsRulerHilite	788
dbIsScreenSel	789
dbIsShapeTypePath	789
dbIsShapeTypePoly	791
dbIsShapeTypeRect	792
dbIsShapeTypeUndefined	793
dbIsSiteCore	795

dbIsStdRowArealo	796
dbIsStdRowFlipped	796
dbIsStdRowHilite	797
dbIsStdRowMarked	798
dbIsStdRowSel	799
dbIsStripBoxDrcObj	799
dbIsStripBoxHilite	800
dbIsStripBoxHookup	801
dbIsStripBoxHor	802
dbIsStripBoxRail	803
dbIsStripBoxSel	804
dbIsStripBoxSignal	804
dbIsStripBoxVer	805
dbIsStripBoxVia	806
dbIsStripBoxViaCell	806
dbIsTechSiteArealo	807
dbIsTechSiteCore	808
dbIsTechSiteGNDOnBottom	809
dbIsTechSiteMarked	810
dbIsTechSiteSymmetryR90	810
dbIsTechSiteSymmetryX	811
dbIsTechSiteSymmetryY	812
dbIsTechSiteVDDOnBottom	813
dbIsTermAsyncCtrl	814
dbIsTermBidi	814
dbIsTermBrkLoop	815
dbIsTermClk	816
dbIsTermD	817
dbIsTermFeed	817
dbIsTermFTerm	818
dbIsTermHilite	820
dbIsTermIgnored	821
dbIsTermInCluInst	821
dbIsTermInPath	822
dbIsTermInput	823
dbIsTermMarked	824
dbIsTermMarked2	825

dblsTermMarked3	825
dblsTermMPW	826
dblsTermMultiHInst	827
dblsTermMultiInst	828
dblsTermOutput	828
dblsTermQ	829
dblsTermSel	830
dblsTermSpecial	831
dblsTermTAClkSrc	832
dblsTermTAIgnored	832
dblsTermTieHi	833
dblsTermTieHiOrLo	834
dblsTermTieLo	834
dblsTermTriCtl	835
dblsTermUnused	836
dblsViaCellDefault	837
dblsViaCellFromDef	837
dblsViaCellNonDefault	838
dblsViaCellRegular	839
dblsVInstHilite	840
dblsVInstMarked	840
dblsVInstMarked2	841
dblsVInstMarked3	842
dblsVInstMarked4	842
dblsVInstSel	843
dblsVInstUnused	844
dblsVNetBNet	845
dblsVNetHilite	846
dblsVNetMarked	846
dblsVNetMarked2	847
dblsVNetSel	848
dblsVTermFTerm	848
dblsVTermTieHi	849
dblsVTermTieLo	850
dblsWireDel	851
dblsWireDup	852
dblsWireGlobal	853

dbIsWireHilite	854
dbIsWireHor	855
dbIsWireMarked	856
dbIsWireMarked2	857
dbIsWireMaster	857
dbIsWireSel	858
dbIsWireSlave	859
dbIsWireTerm	860
dbIsWireVer	860
dbIsWireViaHilite	861
dbIsWireViaSel	862
dbIsWireX	863
dblterAllRoutes	863
dblterRoutes	865
dblterViaCellRectangles	867
4	869
Database Commands L - R	869
L	869
dbLayerArrNext	869
dbLayerBlkBox	870
dbLayerBlkCutLayer	871
dbLayerBlkLayer	871
dbLayerBlkName	872
dbLayerBlkNext	873
dbLayerBlkType	874
dbLayerCap	874
dbLayerCapPerSQ	875
dbLayerDrawName	876
dbLayerExtension	876
dbLayerExtIdList	877
dbLayerFillCheckLength	878
dbLayerFillCheckStep	879
dbLayerFillCheckWidth	879
dbLayerFillMaxDensity	880
dbLayerFillMinDensity	881
dbLayerFillMinSpacing	881

dbLayerId	882
dbLayerLEFId	883
dbLayerLefName	883
dbLayerMaxCap	884
dbLayerMaxWidth	885
dbLayerMinArea	886
dbLayerMinCap	886
dbLayerMinSpace	887
dbLayerMinWidth	888
dbLayerName	888
dbLayerNextInFE	889
dbLayerNextInLEF	890
dbLayerOffset	890
dbLayerPrefDir	891
dbLayerRes	892
dbLayerResistanceTable	892
dbLayerRule2ndLowerWidthRange	893
dbLayerRule2ndUpperWidthRange	894
dbLayerRuleArrNext	894
dbLayerRuleExtension	895
dbLayerRuleHasExtension	895
dbLayerRuleHasRange	896
dbLayerRuleHasSecondRange	896
dbLayerRuleHasThreshold	897
dbLayerRuleHasWidth	897
dbLayerRuleLayer	898
dbLayerRuleLengthThreshold	898
dbLayerRuleLowerWidthRange	899
dbLayerRuleMinSpacing	899
dbLayerRuleMinWidth	900
dbLayerRuleNext	900
dbLayerRuleUpperWidthRange	901
dbLayerShapeLayer	902
dbLayerShapeType	903
dbLayerShapeViaViaCell	903
dbLayerShapeViaViaLoc	904
dbLayerSpace	906

dbLayerThickness	906
dbLayerWireId	907
dbLayerWirePitch	908
dbLayerWireSpace	908
dbLayerWireWidth	909
dbLocPtrLoc	910
dbLocPtrNext	910
dbLocX	911
dbLocY	912
M	912
dbMarkCriticalNet	912
dbMaskPtnPinLayerOnSide	913
dbmCellGetXNetByName	914
dbMergeNetRC	914
dbMergeNetWires	915
dbMicronsToDBU	916
dbMinDistLocToWire	916
dbMultiplyXCap	917
N	918
dbNameToObj	918
dbNameToType	919
dbNetBaseName	919
dbNetBox	920
dbNetBus	921
dbNetCap	921
dbNetChangeWireGlobalStatus	922
dbNetChangeWireStatus	923
dbNetCleanupWires	924
dbNetDate	925
dbNetDefName	925
dbNetDeleteAggrNearestWire	926
dbNetExtrRCSummary	926
dbNetExtRule	927
dbNetFreeWires	927
dbNetFreeWires2	928
dbNetFrequency	929
dbNetGetCritDrivingTimeArc	929

dbNetGlobalWireAddVias	930
dbNetGlobalWireToUsableWire	931
dbNetGroupMsAttr	932
dbNetGroupMsCoaxialLayers	932
dbNetGroupMsMatchOrNot	933
dbNetGroupMsOverlap	933
dbNetGroupMsResistance	934
dbNetGroupMsShieldNet	935
dbNetGroupMsShieldWidth	935
dbNetGroupMsSpacing	936
dbNetGroupMsThreshold	937
dbNetGroupMsTolerance	937
dbNetGroupMsWidth	938
dbNetGroupName	938
dbNetHorWt	939
dbNetLenX	940
dbNetLenY	940
dbNetLogicValue	941
dbNetMaxCapSlack	942
dbNetMaxFanoutSlack	942
dbNetMaxTranSlack	943
dbNetName	944
dbNetNext	944
dbNetNrFanIn	945
dbNetNrFanOut	946
dbNetNrOutTerm	946
dbNetNrTerm	947
dbNetPCell	948
dbNetPdefName	948
dbNetPower	949
dbNetPrefExtraSpace	950
dbNetPrefixId	950
dbNetPrefixName	951
dbNetPrev	952
dbNetPrintWires	952
dbNetSwitchingPower	953
dbNetTermList	954

dbNetToggleDensity	955
dbNetUserWt	955
dbNetVerWt	956
dbNetWidthX	957
dbNetWireLenX	957
dbNetWireLenY	958
dbNetWireList	959
dbNewFPin	959
dbNewHInst	960
O	961
dbObjBox	961
dbObjFPlanBox	962
dbObjName	962
dbObjPrint	963
dbObjPropList	964
dbObjPtr0	964
dbObjPtrPlus1	965
dbObjToName	966
dbObstructBox	966
dbObstructName	967
dbObstructNext	968
dbObstructType	968
dbOpCondName	969
dbOpCondNext	970
dbOpCondProc	971
dbOpCondTemp	971
dbOpCondVolt	972
P	972
dbPathFallDelay	973
dbPathId	973
dbPathLastTerm	974
dbPathNext	975
dbPathRiseDelay	975
dbPathTermList	976
dbPerimBox	977
dbPerimNext	977
dbPerimPtList	978

dbPGTermPGTerm	979
dbPGTermName	980
dbPGTermNet	980
dbPGTermPGFTerm	981
dbPinGroupCell	982
dbPinGroupName	983
dbPinGroupNext	983
dbPinGroupPinList	984
dbPinGroupPinListTail	984
dbPlaceDummyIoPad	985
dbPlaceFTerm	986
dbPlaceInst	986
dbPlaceInstAtOrigin	987
dbPowerDomainCore2Bot	988
dbPowerDomainCore2Left	989
dbPowerDomainCore2Right	990
dbPowerDomainCore2Top	990
dbPowerDomainExtGNet	991
dbPowerDomainExtIntGNet	992
dbPowerDomainExtIntPNet	993
dbPowerDomainExtPNet	993
dbPowerDomainGNet	994
dbPowerDomainGroup	995
dbPowerDomainMinGapBot	996
dbPowerDomainMinGapLeft	997
dbPowerDomainMinGapRight	997
dbPowerDomainMinGapTop	998
dbPowerDomainName	999
dbPowerDomainPNet	1000
dbPowerDomainRowFlip	1000
dbPowerDomainRowSpaceType	1001
dbPowerDomainRowSpacing	1002
dbPowerDomainRSExtBot	1003
dbPowerDomainRSExtLeft	1003
dbPowerDomainRSExtRight	1004
dbPowerDomainRSExtTop	1005
dbPowerDomainTechSite	1006

dbPowerDomainVoltage	1006
dbPowerSinkNet	1007
dbPowerSinkNext	1008
dbPowerSinkPrev	1008
dbPrintCellSym	1009
dbPrintFTermMaxCapTbl	1010
dbPrintFTermMaxTranTbl	1011
dbPrintHInstStat	1011
dbPrintInstByFlag	1012
dbPrintNetByFlag	1013
dbPrintTermByFlag	1014
dbPropDataType	1015
dbPropListDataType	1015
dbPropListName	1016
dbPropListParent	1016
dbPropListValue	1017
dbPropName	1017
dbPropNext	1018
dbPropParent	1019
dbPropTypeDataType	1020
dbPropTypeName	1020
dbPropTypeNext	1021
dbPropTypeTypId	1022
dbPropTypeUsage	1022
dbPropUsage	1023
dbPropValue	1023
dbPt	1024
dbPtnBindFPlanPtnCutToPtnList	1025
dbPtnBox	1026
dbPtnCell	1026
dbPtnCellRailWidth	1027
dbPtnCoreToBottom	1028
dbPtnCoreToLeft	1028
dbPtnCoreToRight	1029
dbPtnCoreToTop	1030
dbPtnCutBox	1031
dbPtnCutName	1031

dbPtnCutNextInFPlan	1032
dbPtNext	1033
dbPtnFeedBox	1033
dbPtnFeedLayer	1034
dbPtnFeedName	1035
dbPtnFeedNext	1036
dbPtnFreePinLayerOnSide	1036
dbPtnHInst	1037
dbPtnHorTrackOffsetOnLayer	1038
dbPtnInst	1038
dbPtnLayerBlocked	1039
dbPtnLayerHaloOnLayer	1040
dbPtnMinPinSpace	1041
dbPtnMinPinSpaceOnSide	1041
dbPtnName	1042
dbPtnNext	1043
dbPtnPCell	1043
dbPtnPinBlkBox	1044
dbPtnPinBlkLayer	1045
dbPtnPinBlkName	1046
dbPtnPinBlkNext	1046
dbPtnPlacementHaloBottomSideSize	1047
dbPtnPlacementHaloLeftSideSize	1048
dbPtnPlacementHaloRightSideSize	1049
dbPtnPlacementHaloTopSideSize	1049
dbPtnRoutingHaloBottomLayer	1050
dbPtnRoutingHaloSideSize	1051
dbPtnRoutingHaloTopLayer	1052
dbPtnStdCellHgt	1052
dbPtnVerTrackOffsetOnLayer	1053
Q	1054
dbQueryCongestAtLoc	1054
dbQueryInstInBox	1055
R	1055
dbRebindPtnCutToPtn	1055
dbReCalculateDefaultTechSite	1056
dbReclaimAreaDownsize	1056

dbRegisterPropType	1057
dbReportMsmvInstVoltage	1058
dbResetIsNetGnd	1058
dbResetIsNetPwr	1059
dbResizeInst	1060
dbRotateLocWROrigin	1060
dbRouteBlkBox	1062
dbRouteBlkLayer	1062
dbRouteBlkName	1063
dbRouteBlkType	1064
dbRouteBoxBox	1065
dbRouteBoxNext	1065
dbRouteBoxRouteGuide	1066
dbRouteBoxZ	1067
dbRouteGuideBoxList	1067
dbRouteGuideHorLayer	1068
dbRouteGuideMinSpace	1069
dbRouteGuideName	1069
dbRouteGuideNetGroup	1070
dbRouteGuideNetList	1071
dbRouteGuideNext	1071
dbRouteGuidePinGroup	1072
dbRouteGuideVerLayer	1073
dbRouteNext	1073
dbRowBox	1075
dbRowClusterBox	1076
dbRowClusterName	1076
dbRowClusterNext	1077
dbRowClusterNrRow	1078
dbRowClusterRowList	1078
dbRowId	1079
dbRowNext	1080
dbRowNextInReg	1080
dbRowOrient	1081
dbRowSite	1082
dbRowTechSite	1082
dbRulerElbow	1083

dbRulerGrab	1084
dbRulerGrabPosition	1085
dbRulerLoc	1086
dbRulerLoc0	1086
5	1088
Database Commands S - Z	1088
S	1088
dbScreenBox	1088
dbScreenCapacity	1089
dbScreenName	1089
dbScreenNext	1090
dbSelectInstanceByFlag	1090
dbSelNext	1091
dbSelPtr	1092
dbSetAllNetFrequency	1092
dbSetAreaECOWindow	1093
dbSetByPassDeleteNetWireTerm	1093
dbSetCellCongestionView	1094
dbSetCellDoNotFlatten	1095
dbSetCellLeakagePower	1095
dbSetCellRouteAlgUsed	1096
dbSetCellSpareGate	1097
dbSetCellType	1097
dbSetCloneListOrient	1098
dbSetConstraintType	1099
dbSetCurrentNet	1100
dbSetCurrentZ	1100
dbSetFPinLoc	1101
dbSetFPlanBox	1102
dbSetFPlanIloBox	1103
dbSetFPlanLayerHalo	1103
dbSetFPlanRegBox	1104
dbSetFTermDepth	1105
dbSetFTermLoc	1106
dbSetFTermPlacementStatus	1106
dbSetFTermType	1107

dbSetFTermWidth	1108
dbSetGeomBoxBox	1109
dbSetGeomLineEndPt	1110
dbSetGeomLineStartPt	1111
dbSetGeomTextHeight	1111
dbSetGeomTextLoc	1112
dbSetGeomTextText	1113
dbSetGroupBox	1113
dbSetGroupConType	1114
dbSetHeadDBUPerIGU	1115
dbSetHeadInHierMode	1116
dbSetHeadStdCellHgt	1116
dbSetHInstColorId	1117
dbSetHInstDensity	1118
dbSetHInstHorOrder	1118
dbSetHInstPin	1119
dbSetHInstRootX	1120
dbSetHInstUserDensity	1121
dbSetHInstVerOrder	1121
dbSetInstFlag	1122
dbSetInstHorOrder	1123
dbSetInstObstruct	1123
dbSetInstPlacementStatus	1124
dbSetInstPriority	1125
dbSetInstRoutingHaloBottomLayer	1126
dbSetInstRoutingHaloSideSize	1126
dbSetInstRoutingHaloTopLayer	1127
dbSetInstSpareGate	1128
dbSetInstVerOrder	1128
dbSetLoBox	1129
dbSetLoBump	1130
dbSetLoCell	1130
dbSetLoFTerm	1131
dbSetLoIndent	1131
dbSetLoInstance	1132
dbSetLoLayerId	1133
dbSetLoLoc	1133

dbSetIoName	1134
dbSetIoNext	1134
dbSetIoOffset	1135
dbSetIoOrder	1136
dbSetIoOrient	1136
dbSetIoRow	1137
dbSetIoRowMargin	1138
dbSetIoSide	1138
dbSetIoSpacing	1139
dbSetIoTdfLibName	1139
dbSetIsCellDontTouch	1140
dbSetIsCellDontUse	1141
dbSetIsCellDummy	1141
dbSetIsCellIo	1142
dbSetIsCellSpareGate	1143
dbSetIsCellITGDelayUpdated	1143
dbSetIsFTermAssigned	1144
dbSetIsFTermPreassigned	1145
dbSetIsFTermScanClk	1145
dbSetIsFTermScanIn	1146
dbSetIsFTermScanInv	1147
dbSetIsFTermScanOut	1147
dbSetIsGroupPhyHier	1148
dbSetIsGroupUngroup	1149
dbSetIsHeadCustomLayerChanged	1149
dbSetIsHeadDesignInMemory	1150
dbSetIsHeadFPlanChanged	1150
dbSetIsHeadHiliteSticky	1151
dbSetIsHInstHidden	1152
dbSetIsHInstInFPlan	1152
dbSetIsHInstPinInitiated	1153
dbSetIsHInstUngroup	1154
dbSetIsInstDefCovered	1155
dbSetIsInstDontTouch	1155
dbSetIsInstHilite	1156
dbSetIsInstIPOed	1157
dbSetIsInstMarked	1158

dbSetIsInstMarked2	1158
dbSetIsInstMarked3	1159
dbSetIsInstMarked4	1160
dbSetIsInstMoved	1160
dbSetIsInstPlaced	1161
dbSetIsInstPreplaced	1162
dbSetIsInstSpareGate	1163
dbSetIsInstTAlgnored	1163
dbSetIsInstUnused	1164
dbSetIsloAssigned	1165
dbSetIsloBump	1165
dbSetIsloClearance	1166
dbSetIsloCorner	1167
dbSetIsloCovered	1168
dbSetIsloDefCovered	1168
dbSetIsloDummyPad	1169
dbSetIsloFixed	1170
dbSetIsloGapFixed	1170
dbSetIsloGroundPad	1171
dbSetIsloHilite	1172
dbSetIsloMarked	1173
dbSetIsloPowerPad	1173
dbSetIsloPreplaced	1174
dbSetIsloSel	1175
dbSetIsNetAnalog	1175
dbSetIsNetAvoidDetour	1176
dbSetIsNetClock	1177
dbSetIsNetCritical	1178
dbSetIsNetDefInClock	1178
dbSetIsNetDontTouch	1179
dbSetIsNetGnd	1180
dbSetIsNetHilite	1180
dbSetIsNetIgnoreInRoute	1181
dbSetIsNetIPOed	1182
dbSetIsNetIPOIgnored	1182
dbSetIsNetListChangedForClink	1183
dbSetIsNetMarked	1184

dbSetIsNetMarked2	1184
dbSetIsNetMarked3	1185
dbSetIsNetMarked4	1186
dbSetIsNetPostRouteSiFix	1186
dbSetIsNetPwr	1187
dbSetIsNetScanNet	1188
dbSetIsNetSkipRouting	1188
dbSetIsNetTrialRouted	1189
dbSetIsRouteGuideOptimizePin	1190
dbSetIsRulerHilite	1191
dbSetIsRulerSelHTick	1191
dbSetIsRulerSelVTick	1192
dbSetIsStripBoxRail	1193
dbSetIsTermIgnored	1194
dbSetIsTermIPOIgnored	1194
dbSetIsTermMarked	1195
dbSetIsTermMarked2	1196
dbSetIsTermMarked3	1196
dbSetIsTermPlaceIgnored	1197
dbSetIsTermTAlgnored	1198
dbSetIsVNetGnd	1198
dbSetIsVNetPwr	1199
dbSetIsXNetGnd	1200
dbSetIsXNetPwr	1200
dbSetLayerBlkCutLayer	1201
dbSetLayerBlkLayer	1202
dbSetLayerBlkName	1202
dbSetLayerCapPerSQ	1203
dbSetLayerDrawName	1204
dbSetLayerMaxCap	1204
dbSetLayerMinCap	1205
dbSetLayerName	1206
dbSetLayerPrefDir	1207
dbSetLayerShapeType	1207
dbSetLayerSpace	1208
dbSetLayerThickness	1209
dbSetLayerWirePitch	1209

dbSetLayerWireWidth	1210
dbSetMacroDensity	1211
dbSetNetDblWidth	1211
dbSetNetExtRule	1212
dbSetNetFlag	1213
dbSetNetFrequency	1214
dbSetNetHorWt	1214
dbSetNetPCell	1215
dbSetNetPrefExtraSpace	1215
dbSetNetUserWt	1216
dbSetNetVerWt	1217
dbSetNetWidthX	1218
dbSetNetWireStatus	1218
dbSetNumOfloRow	1219
dbSetObjFPlanBox	1220
dbSetObstructName	1221
dbSetPinGuideAsFeedthrough	1222
dbSetPrerouteAsObs	1222
dbSetPropTypeUsage	1223
dbSetPropValue	1224
dbSetPtnCellRailWidth	1224
dbSetPtnFeedLayer	1225
dbSetPtnFeedName	1226
dbSetPtnFreePinLayerOnSide	1227
dbSetPtnLayerBlocked	1227
dbSetPtnLayerBlockedOnLayer	1228
dbSetPtnLayerHalo	1229
dbSetPtnLayerHaloOnLayer	1229
dbSetPtnMinPinSpace	1230
dbSetPtnMinPinSpaceOnSide	1231
dbSetPtnPinBlkLayer	1232
dbSetPtnPinBlkName	1232
dbSetPtnPinLayerFreeOnSideOnLayer	1233
dbSetPtnPlacementHaloBottomSideSize	1234
dbSetPtnPlacementHaloLeftSideSize	1235
dbSetPtnPlacementHaloRightSideSize	1235
dbSetPtnPlacementHaloTopSideSize	1236

dbSetPtnRoutingHaloBottomLayer	1236
dbSetPtnRoutingHaloSideSize	1237
dbSetPtnRoutingHaloTopLayer	1238
dbSetPtnStdCellHgt	1238
dbSetRouteBlkLayer	1239
dbSetRouteBlkName	1239
dbSetRulerGrab	1240
dbSetScreenCapacity	1241
dbSetScreenName	1241
dbSetStdPlaceHInstRoot	1242
dbSetStdRowOrient	1243
dbSetStripBoxShape	1243
dbSetStripBoxShieldNet	1244
dbSetStripBoxState	1245
dbSetStripBoxZ	1246
dbSetTechSite	1246
dbSetUserURrid	1247
dbSetViaCellContactId	1248
dbSetWireTerm	1248
dbSetWireViaDnId	1249
dbSetWireViaDnStatus	1250
dbSetWireWidthXId	1250
dbSetWireWireStatus	1251
dbSetWireXXMode	1252
dbShapePoly	1252
dbShapeRect	1253
dbSiteName	1254
dbSiteSizeX	1255
dbSiteSizeY	1255
dbSnapCoordToTrack	1256
dbSNetBoxList	1257
dbSNetCell	1257
dbSNetName	1258
dbSNetNext	1259
dbSNetNrTerm	1259
dbSNetTermList	1260
dbSNetViaList	1260

dbSortHInstByHorOrder	1261
dbSortHInstByVerOrder	1262
dbSpacingLayer1	1262
dbSpacingLayer2	1263
dbSpacingMinSpacing	1264
dbSpacingNext	1264
dbStdRowBox	1265
dbStdRowId	1266
dbStdRowNext	1267
dbStdRowNextInReg	1267
dbStdRowOrient	1268
dbStdRowSite	1269
dbStdRowTechSite	1270
dbStripBoxBox	1271
dbStripBoxList	1272
dbStripBoxNet	1272
dbStripBoxNext	1273
dbStripBoxShape	1274
dbStripBoxState	1275
dbStripBoxStrip	1276
dbStripBoxZ	1276
dbStripCell	1277
dbStripName	1277
dbStripNext	1278
dbStripSNet	1279
dbSwitchTermNet	1279
dbSymArrHInst	1280
dbSymArrNextPrefixIdInHier	1281
dbSymArrPrefix	1281
dbSymHInst	1282
dbSymId	1282
dbSymNextInArr	1283
dbSymNextPrefixIdInHier	1284
dbSymPrefix	1284
T	1285
dbTechSiteName	1285
dbTechSiteNext	1286

dbTechSiteSizeX	1286
dbTechSiteSizeY	1287
dbTermFTerm	1288
dbTermFTermWPG	1289
dbTermlIdx	1289
dbTermlIdxWPG	1290
dbTermlInst	1291
dbTermlInstName	1291
dbTermLayer	1292
dbTermLibraryContext	1293
dbTermLoc	1293
dbTermLocWithZ	1294
dbTermLogicValue	1294
dbTermMaxCap	1295
dbTermMaxTran	1296
dbTermMaxTranSlack	1296
dbTermName	1297
dbTermNet	1298
dbTermNextlnInst	1298
dbTermNextlnInstArr	1299
dbTermNextlnNet	1300
dbTermTranTime	1300
dbTermZ	1301
dbTimeArcRelFTerm	1302
dbTimeArcTgtFTerm	1302
dbTimeArcType	1303
dbTimeLibDefOpCond	1305
dbTimeLibInName	1305
dbTimeLibLibraryContext	1306
dbTimeLibName	1307
dbTimeLibNext	1307
dbTimeLibNomOpCond	1308
dbTimeLibTimeLibGrp	1309
dbTimeLibTimeLibGrpName	1309
dbTIsCellCell	1310
dbTIsCellLibraryContext	1311
dbTIsCellTimeLib	1311

dbTrimCellUnplacedYetConnectedInsts	1312
U	1312
dbUnassignBump	1312
dbUnassignTilePin	1313
dbUnflattenHInst	1314
dbUnfreezeAllInst	1314
dbUnfreezeWindowArea	1315
dbUpdateAllWireTerm	1315
dbUpdateIoSidesFromLocs	1316
dbUpdateNetTiming	1317
dbUpdatePinGuideRouteBox	1317
dbUpdateTermZFromFTerm	1318
V	1318
dbVerifyAllFTermLeaf	1318
dbViaCellContactId	1319
dbViaCellCutBox	1319
dbViaCellCutLayer	1320
dbViaCellDx	1321
dbViaCellDy	1321
dbViaCellHiBox	1322
dbViaCellHiLayer	1323
dbViaCellListViaCell	1323
dbViaCellLoBox	1324
dbViaCellLoLayer	1325
dbViaCellLx	1325
dbViaCellLy	1326
dbViaCellName	1327
dbViaCellNext	1327
dbViaCellPatternName	1328
dbViaCellRectanglesNext	1328
dbViaCellRes	1329
dbViaCellRuleId	1330
dbViaCellViaCode	1331
dbViaCellZ	1331
dbViaLoc	1332
dbViaViaCell	1333
dbVInstCell	1333

dbVInstName	1334
dbVInstNrTerm	1334
dbVNetBus	1335
dbVNetName	1336
dbVTermFTerm	1336
dbVTermInst	1337
dbVTermNet	1338
W	1338
dbWasHeadDesignChanged	1339
dbWasSelSetChanged	1339
dbWireAdjD	1340
dbWireAdjDir	1341
dbWireAdjE	1341
dbWireAdjN	1342
dbWireAdjS	1343
dbWireAdjU	1343
dbWireAdjW	1344
dbWireBox	1345
dbWireCleanup	1345
dbWireConnect	1346
dbWireCreate	1346
dbWireDegree	1347
dbWireDir	1348
dbWireDnViaCell	1348
dbWireDnViaRuleId	1349
dbWireDup	1350
dbWireDupSafe	1350
dbWireExt1	1351
dbWireExt2	1352
dbWireGetOrCreateInWireDir	1352
dbWireHead	1353
dbWireHeadWidth	1354
dbWireLen	1354
dbWireLoc	1355
dbWireLocWithZ	1356
dbWireNet	1356
dbWireNextInNet	1357

dbWireRawTerm	1358
dbWireRuleId	1358
dbWireRuleIdx	1359
dbWireTerm	1359
dbWireViaDnId	1360
dbWireViaDnRuleIdx	1361
dbWireViaDnStatus	1362
dbWireWidthXId	1363
dbWireWidthXIdx	1363
dbWireWidthYIdx	1364
dbWireWireStatus	1365
dbWireZ	1366
6	1367
dbg Commands	1367
dbgCellList	1367
dbgConvertSetupHoldOnAsyncToRecRem	1368
dbgCurRuler	1368
dbgDBUPerlGU	1369
dbgDBUPerlGU	1369
dbgDBUPerMGrid	1369
dbgDesignName	1370
dbgHead	1370
dbgHierChar	1371
dbgIsHInstRootX	1371
dbgLayerArr	1372
dbgMicronPerDBU	1372
dbgNrLayer	1373
dbgPicoFPerDBU	1373
dbgPicoSecPerDBU	1374
dbgTopCell	1374

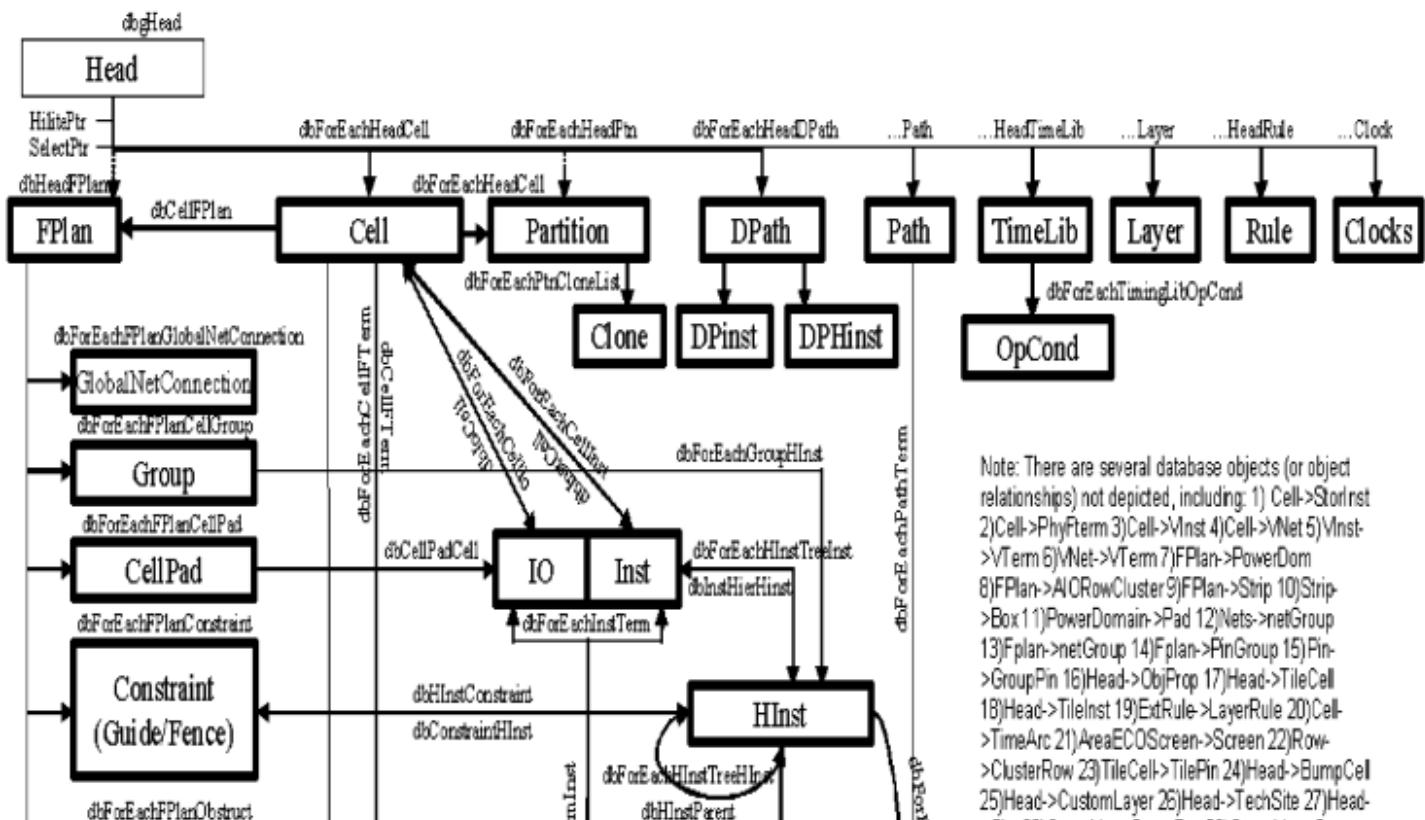
About This Manual

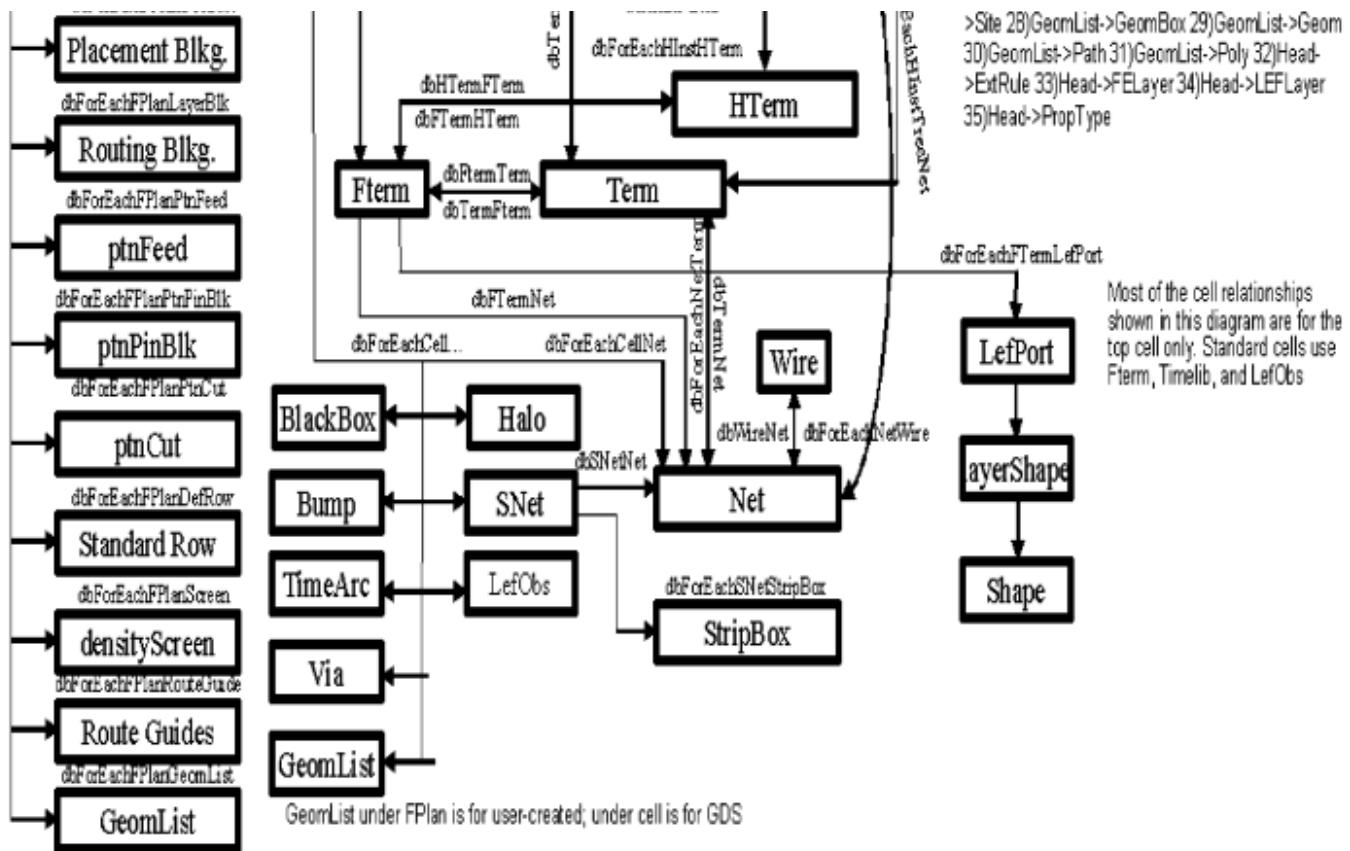
This document describes the database access commands for the Innovus Implementation System (Innovus) hierarchical RTL-to-GDSII physical implementation solution. This document is written for highly experienced designers who are proficient in the use of the Innovus software. This document assumes that the reader has read and understood the related Innovus documentation and is familiar with Innovus terminology. The reader is also expected to have a solid understanding of UNIX and Tcl/Tk programming.

Database Access Command Diagrams

The following diagram shows the database access command relationships between design objects. Most of the cell relationships shown in this diagram are for the top cell only. Standard cells use `Fterm`, `TimeLib`, and `LefObs`.

Head acts as the root for all design objects. Therefore, to retrieve design objects, you typically use `dbHeadobject` or `dbForEachHeadobject` commands. For example, to retrieve all of the cells of a design, use the `dbForEachHeadCell` command.



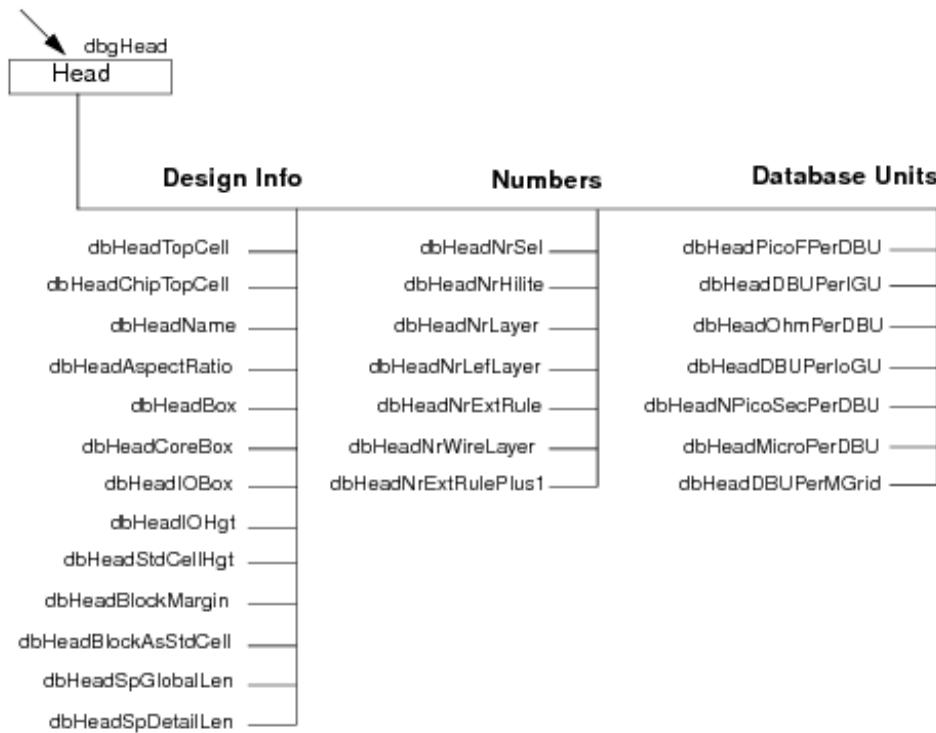


Note: Several database objects and object relationships are not shown in the above diagram, including:

- AreaECOScreen->Screen
- Cell->PhyFTerm
- Cell ->StorInst
- Cell->TimeArc
- Cell->VInst
- Cell->VNet
- ExtRule->LayerRule
- FPlan->AIORowCluster
- FPlan->NetGroup
- FPlan->PinGroup
- FPlan->PowerDom

- FPlan->Strip
- GeomList->Geom
- GeomList->GeomBox
- GeomList->Path
- GeomList->Poly
- Head->BumpCell
- Head->CustomLayer
- Head->ExtRule
- Head->FELayer
- Head->LEFLayer
- Head->ObjProp
- Head->PropType
- Head->Site
- Head->TechSite
- Head->TileCell
- Head->TileInst
- Nets->NetGroup
- Pin->GroupPin
- PowerDomain->Pad
- Row->ClusterRow
- Strip->Box
- TileCell->TilePin
- VInst->VTerm
- VNet->VTerm

Design information is stored additionally as single objects. There are database access commands to return design information, the number of objects in the design, and to convert between database units.



Note: This document applies to the Innovus user interface. See the [Innovus Stylus Common UI Database Object Information \(Limited Access\)](#) document for the Innovus Stylus user interface.

Related Documents

For more information about the Innovus family of products, see the following documents. You can access these and other Cadence documents with the Cadence Help documentation system.

Innovus Product Documentation

- [What's New in Innovus](#)
Provides information about new and changed features in this release of the Innovus family of products.
- [Innovus User Guide](#)
Describes how to install and configure the Innovus software, and provides strategies for implementing digital integrated circuits.
- [Innovus Menu Reference](#)
Provides information specific to the forms and commands available from the

Innovus graphical user interface.

- *Innovus Text Command Reference*

Describes the Innovus text commands, including syntax and examples.

- *Innovus Foundation Flows Guide*

Describes how to use the scripts that represent the recommended implementation flows for digital timing closure with the Innovus software.

- *Mixed Signal Interoperability Guide*

Describes the digital mixed-signal flow.

- README file

Contains installation, compatibility, and other prerequisite information, including a list of Cadence Change Requests (CCRs) that were resolved in this release. You can read this file online at downloads.cadence.com.

For a complete list of documents provided with this release, see the Cadence Help documentation system.

Innovus Stylus Common UI Product Documentation

- *Innovus Stylus Common UI Quick Start Guide*
Provides information on starting with the Stylus Common User Interface.
- *What's New in Innovus Stylus Common UI*
Provides information about new and changed features in this release of the Innovus family of products.
- *Innovus Stylus Common UI User Guide*
Describes how to install and configure the Innovus Stylus Common UI software, and provides strategies for implementing digital integrated circuits.
- *Innovus Stylus Common UI Text Reference Manual*
Describes the Innovus Stylus Common UI text commands, including syntax and examples.
- *Innovus Stylus Common UI Menu Reference*
Provides information specific to the forms and commands available from the Innovus Stylus Common UI graphical user interface.
- *Stylus Common UI Database Object Information*
Provides information about Stylus Common UI database objects.
- *Innovus Stylus Common UI Mixed Signal (MS) Interoperability Guide*
Describes the digital mixed-signal flow using Innovus Stylus Common UI.

For a complete list of documents provided with this release, see the Cadence Help online documentation system.

Database Commands A - E

- [A](#)
- [B](#)
- [C](#)
- [D](#)
- [E](#)

A

- [dbAddCellObs](#)
- [dbAddCoverInst](#)
- [dbAddDeCapAtLoc](#)
- [dbAddToHInst](#)
- [dbAddToSelSet](#)
- [dbAddWireSeg](#)
- [dbAreAnyNetWiresDangling](#)
- [dbAreCellsPhyReplaceable](#)
- [dbAreNetWiresOK](#)
- [dbAssignFTermToBump](#)
- [dbAttachBuffer](#)

dbAddCellObs

`dbAddCellObs`

```
-cell cell_ptr
-layer layername
-box box
```

Supports selection of macro and standard cell pins for Jenga via enhancement.

Parameters

-cell	Name of cell/block partition of the object for which the obstruction is added.
-layer	Name of the obstruction layer. It will be a LEF layer name
-box	Name of the co-ordinates of the obstruction in the following format with reference to the cell origin in db units: <i>llx1 lly1 urx2 ury2</i>

Command Order

Use this command after importing the design.

Example

```
dbAddCellObs <cell_ptr> <layer_ptr> { <llx> <llx> <urx> \
<ury> | <box> }
```

dbAddCoverInst

```
dbAddCoverInst nameOfInst
```

Makes the specified instance a cover macro.

Parameters

<i>nameOfInst</i>	Name of the instance.
-------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbAddCoverInst TOP_CHIP_COVER
```

Makes instance TOP_CHIP_COVER a cover macro.

dbAddDeCapAtLoc

```
dbAddDeCapAtLoc cellPtr dbLocX dbLocY orient instName
```

Adds a decap cell at the specified coordinate with the specified orientation.

Parameters

<i>cellPtr</i>	Address of instance or hierarchical instance.
<i>dbLocX</i>	X ordinate in DBU units.
<i>dbLocY</i>	Y ordinate in DBU units.
<i>orient</i>	Orientation.
<i>instName</i>	Instance name.

Command Order

Use this command after importing the design.

Example

```
dbAddDeCapAtLoc OR4X2 [dbMicronsToDBU 602] [dbMicronsToDBU 814] R180 DECAPINST
```

Creates an instance called `DECAPINST` at `602.0 814.0` with an orientation of `R180`.

dbAddToHInst

```
dbAddToHInst inst_or_hinst_ptr hinstPtr
```

Adds an instance to another module, but does not change the netlist.

Parameters

<code>inst_or_hinst_ptr</code>	Address of instance, or hierarchical instance
<code>hinstPtr</code>	Address of hierarchical instance your adding to.

Command Order

Use this command after importing the design.

dbAddToSelSet

```
dbAddToSelSet headPtr ptr
```

Adds the specified object to the currently selected set.

Parameters

<i>headPtr</i>	Address of head.
<i>ptr</i>	Address of object.

Command Order

Use this command after importing the design.

Example

After you select a set of objects, type the following command:

```
dbAddToSelSet [dbgHead] $objPtr
```

Perform a redraw to see that the object has been added to the selected set.

dbAddWireSeg

```
dbAddWireSeg x y z x y z
```

Adds a wire segment.

Parameters

<i>x</i>	The x ordinate.
<i>y</i>	The y ordinate.
<i>z</i>	The metal layer.

Command Order

Use this command during partition floorplanning.

Example

```
dbAddWireSeg 2353.8000 7123.000 2 2353.8000 8298.2000 2
```

Creates a vertical *metal2* wire at the coordinates (2353.8, 7123.0) and (2353.8, 8298.2).

dbAreAnyNetWiresDangling

```
dbAreAnyNetWiresDangling netPtr
```

Reports whether a net is dangling. The command returns 1 if the net is dangling and 0 if it is not.

Parameters

netPtr	Address of the net.
--------	---------------------

Command Order

Use this command after importing the design.

Example

```
set netPtr [dbGetNetByName SH17/I444/I20/GE_9692/A21_Z]
```

Returns:

0xedc2994

```
dbAreAnyNetWiresDangling $netPtr
```

Returns:

0

The net SH17/I444/I20/GE_9692/A21_Z is not dangling.

dbAreCellsPhyReplaceable

dbAreCellsPhyReplaceable *cellPtr* *cellPtr*

Determines whether the specified cell can be physically replaced by another specified cell. The command returns 1 if the cell is physically replaceable and 0 if it is not.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
set cellPtr1 [dbGetCellByName BUFX1]
```

Returns:

0xf177760

```
set cellPtr2 [dbGetCellByName CLKBUFX2]
```

Returns:

0xf1710d0

```
dbAreCellsPhyReplaceable $cellPtr1 $cellPtr2
```

Returns:

0

The BUFX1 cell is not physically replaceable by the CLKBUFX2 cell.

dbAreNetWiresOK

dbAreNetWiresOK *netPtr*

Determines if a wire is okay.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
set netPtr [dbGetNetByName SH15/I465_Z]
```

Returns:

0x104b1658

```
dbAreNetWiresOK $netPtr
```

Returns:

1

The net SH15/I465_Z is okay.

dbAssignFTermToBump

`dbAssignFTermToBump ftermPtr bumpPtr`

Assigns a bump to the specified f-term.

Parameters

<i>ftermPtr</i>	Address of a f-term
<i>bumpPtr</i>	Address of bump

Command Order

Use this command after design import.

dbAttachBuffer

`dbAttachBuffer {<termPtr>|<fterm>} <cellPtr> <instName> <netName>`

Adds a buffer to the specified terminal.

Parameters

<i>termPtr/fterm</i>	Address of terminal of f-term.
<i>cellPtr</i>	Address of cell.
<i>instName</i>	Instance name.
<i>netName</i>	Net name.

Command Order

Use this command after importing the design.

Example

```
innovus 5> set termPtr [dbGetFTermByName BLU]  
0xfb08e48  
innovus 6> set cellPtr [dbgTopCell]  
0xf930aa8  
innovus 7> dbAttachBuffer $termPtr $cellPtr ECO_BUFF1 BLU
```

Adds a buffer to terminal BLU.

B

dbBlackBoxArea

dbBlackBoxArea *blackBoxPtr*

Returns the area of the specified blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]  
0x83afc70  
innovus 18> dbBlackBoxArea $blackBoxPtr  
400.0
```

The blackbox area is 400.0.

dbBlackBoxAreaPerGate

`dbBlackBoxAreaPerGate blackBoxPtr`

Returns the area per gate of the specified blackbox.

Parameters

<code>blackBoxPtr</code>	Address of blackbox.
--------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
encounter 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]  
0x83afc70  
encounter 18> dbBlackBoxAreaPerGate $blackBoxPtr  
3.80
```

The area per gate is 3.80.

dbBlackBoxAspectRatio

dbBlackBoxAspectRatio *blackBoxPtr*

Returns the aspect ratio of the specified blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]  
0x83afc70  
innovus 18> dbBlackBoxAspectRatio $blackBoxPtr  
1.2
```

The aspect ratio is 1.2 for the specified blackbox.

dbBlackBoxCell

dbBlackBoxCell *blackBoxPtr*

Returns the address of the master cell of the specified blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]
```

```
0x83afc70
```

```
innovus 18> dbBlackBoxCell $blackBoxPtr
```

```
0x93f9352
```

The address of the blackbox is 0x93f9532.

dbBlackBoxGateCount

```
dbBlackBoxGateCount blackBoxPtr
```

Returns the number of gates of the specified blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]
```

```
0x83afc70
```

```
innovus 18> dbBlackBoxGetCount $blackBoxPtr
```

```
53.0
```

The number of gates for the blackbox is 53.

dbBlackBoxHeight

```
dbBlackBoxHeight blackBoxPtr
```

Returns the height of the blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]
```

```
0x83afc70
```

```
innovus 18> dbBlackBoxHeight $blackBoxPtr
```

```
229.68
```

The blackbox height is 229.68.

dbBlackBoxName

```
dbBlackBoxName blackBoxPtr
```

Returns the name of the specified blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]  
0x83afc70  
innovus 18> dbBlackBoxName $blackBoxPtr  
results_conv
```

The name of the blackbox is `results_conv`.

dbBlackBoxNext

```
dbBlackBoxNext blackBoxPtr
```

Returns the address of the next blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]  
0x83afc70  
innovus 18> dbBlackBoxNext $blackBoxPtr  
0x83afd74
```

The address of the next blackbox is 0x83afd74.

dbBlackBoxSpecifyMethod

`dbBlackBoxSpecifyMethod blackBoxPtr`

Determines whether a blackbox was specified. The command returns 1 if the blackbox was specified and 0 if it was not.

Parameters

<code>blackBoxPtr</code>	Address of black box.
--------------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]  
0x83afc70  
innovus 18> dbBlackBoxSpecifyMethod $blackBoxPtr  
0
```

The blackbox was not specified.

dbBlackBoxSpecifyValue

dbBlackBoxSpecifyValue *blackBoxPtr*

Returns the value of the specified blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]
```

```
0x83afc70
```

```
innovus 18> dbBlackBoxSpecifyValue $blackBoxPtr
```

```
0
```

The value is 0.

dbBlackBoxUtilization

dbBlackBoxUtilization *blackBoxPtr*

Specifies the address of the utilization of the specified blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
set blackBoxPtr [dbGetBlackBoxByName results_conv]
```

Returns:

0x83afc70

Then:

```
dbBlackBoxUtilization $blackBoxPtr
```

Returns:

0.7

The utilization is 70 percent.

dbBlackBoxWidth

```
dbBlackBoxWidth blackBoxPtr
```

Returns the width of the specified blackbox.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> set blackBoxPtr [dbGetBlackBoxByName results_conv]  
0x83afc70  
innovus 18> dbBlackBoxWidth $blackBoxPtr  
457.38
```

The width is 457.38.

dbBoxDimX

`dbBoxDimX box_coords`

Returns the x dimension of the specified box.

Parameters

<code><i>box_coords</i></code>	List of box coordinates.
--------------------------------	--------------------------

Command Order

Use this command after starting the Innovus Implementation System (Innovus) software.

Example

```
innovus> dbBoxDimX [list 0 0 100 100]  
100
```

The x dimension of the box is 100.

dbBoxDimY

dbBoxDimY *box_coords*

Returns the y dimension of the specified box.

Parameters

<i>box_coords</i>	List of box coordinates.
-------------------	--------------------------

Command Order

Use this command after starting the Innovus software.

Example

```
innovus 35> dbBoxDimY [list 0 10 100 100]  
90
```

The y dimension of the box is 90.

dbBoxLL

dbBoxLL *box_coords*

Returns the lower left coordinate of the specified box.

Parameters

<i>box_coords</i>	List of box coordinates.
-------------------	--------------------------

Command Order

Use this command after starting the Innovus software.

Example

```
dbBoxLL [list 0 10 100 100]
```

Returns;

```
0 10
```

The lower left coordinate is (0, 10).

dbBoxLLX

```
dbBoxLLX box_coords
```

Returns the lower left x ordinate of the specified box.

Parameters

<i>box_coords</i>	List of box coordinates.
-------------------	--------------------------

Command Order

Use this command after starting the Innovus software.

Example

```
innovus 37> dbBoxLLX [list 0 10 100 100]
```

```
0
```

The lower left x ordinate is 0.

dbBoxLLY

`dbBoxLLY box_coords`

Returns the lower left y ordinate of the specified box.

Parameters

<code><i>box_coords</i></code>	List of box coordinates.
--------------------------------	--------------------------

Command Order

Use this command after starting the Innovus software.

Example

```
innovus 37> dbBoxLLY [list 0 10 100 100]
```

```
10
```

The lower left y ordinate is 10.

dbBoxPtrBox

`dbBoxPtrBox boxPtr`

Returns the address of the specified box.

Parameters

boxPtr

Address of box

Command Order

Use this command after starting the Innovus software.

Example

```
innovus 37> dbBoxPtrBox $boxPtr
0x83afc70
```

The address of the box is 0x83afc70.

dbBoxUR

`dbBoxUR box_coords`

Returns the upper right coordinate of the specified box.

Parameters

box_coords

List of box coordinates.

Command Order

Use this command after starting the Innovus software.

Example

```
innovus 37> dbBoxUR [list 0 10 100 100]
100 100
```

The upper right coordinate is (100,100).

dbBoxURX

dbBoxURX *box_coords*

Returns the upper right x ordinate of the specified box.

Parameters

<i>box_coords</i>	List of box coordinates.
-------------------	--------------------------

Command Order

Use this command after starting the Innovus software.

Example

```
innovus 37> dbBoxURX [list 0 10 100 100]  
100
```

The upper right x ordinate is 100.

dbBoxURY

dbBoxURY *box_coords*

Returns the upper right y ordinate of the specified box.

Parameters

<i>box_coords</i>	List of box coordinates.
-------------------	--------------------------

Command Order

Use this command after starting the Innovus software.

Example

```
innovus 37> dbBoxURY [list 0 10 100 100]  
100
```

The y ordinate is 100.

dbBumpBumpCell

dbBumpBumpCell *bumpPtr*

Returns the address of the master for the specified bump cell.

Parameters

<i>bumpPtr</i>	Address of the bump.
----------------	----------------------

Command Order

Use this command after starting the Innovus software.

Example

```
innovus 37> dbBumpBumpCell $bumpPtr  
0x83afc70
```

The address of the master bump cell is 0x83afc70.

dbBumpCellBox

dbBumpCellBox *bumpcellPtr*

Returns the box coordinates for the specified bump cell.

Parameters

<i>bumpcellPtr</i>	Address of bump cell
--------------------	----------------------

Command Order

Use this command after importing the design and loading an I/O file.

Example

```
innovus 62> dbBumpCellBox $objPtr  
0 0 10000 10000
```

Returns the box of the specified bump cell.

dbBumpCellName

dbBumpCellName *bumpcellPtr*

Returns the name of the specified bump cell.

Parameters

<i>bumpcellPtr</i>	Address of bump cell.
--------------------	-----------------------

Command Order

Use this command after importing the design and loading an I/O file.

Example

```
innovus 59> dbBumpCellName $objPtr
```

```
A1
```

The name of the bump cell is `A1`.

dbBumpCellNext

```
dbBumpCellNext bumpcellPtr
```

Returns the address of the next bump cell.

Parameters

<i>bumpcellPtr</i>	Address of bump cell.
--------------------	-----------------------

Command Order

Use this command after importing the design and loading an I/O file.

Example

```
dbBumpCellNext $objPtr
```

Returns:

0x9a448ca0

The address of the next bump cell is 0x9a448ca0.

dbBumpCellNrItem

```
dbBumpCellNrItem bumpcellPtr
```

Returns the number of items for the specified bump cell. The command returns 0x0 if there are none.

Parameters

<i>bumpcellptr</i>	Address of bump cell.
--------------------	-----------------------

Command Order

Use this command after importing the design and loading an I/O file.

Example

```
dbBumpCellNrItem $objPtr
```

Returns:

43

The number of items is 43.

dbBumpCellPtArr

dbBumpCellPtArr *bumpcellPtr*

Returns the address of the list of bump cells.

Parameters

bumpcellptr	Address of bump cell.
-------------	-----------------------

Command Order

Use this command after importing the design and loading an I/O file.

Example

dbBumpCellPtArr \$objPtr

Returns:

0xd786218

The address of the bump cell array is 0xd786218.

dbBumpFTerm

dbBumpFTerm *bumpPtr*

Returns the f-term of the specified bump.

Parameters

bumpPtr	Address of bump.
---------	------------------

Command Order

Use this command after importing the design and loading an I/O file.

Example

```
innovus 60> dbBumpFTerm $objPtr
```

BLU

The name of the f-term is BLU.

dbBumpLoc

```
dbBumpLoc bumpPtr
```

Returns the location of the specified bump.

Parameters

<i>bumpPtr</i>	Address of bump cell.
----------------	-----------------------

Command Order

Use this command after importing the design and loading an I/O file.

Example

```
innovus 61> dbBumpLoc $objPtr
```

435800 257200

The location of the bump is 435.8 257.2.

dbBumpOrient

`dbBumpOrient bumpPtr`

Returns the orientation of the specified bump.

Parameters

<code>bumpPtr</code>	Address of bump.
----------------------	------------------

Command Order

Use this command after importing the design and loading an I/O file.

Example

```
innovus 61> dbBumpOrient $objPtr  
dbcR0
```

The orientation of the specified bump is R0.

dbBumpType

`dbBumpType bumpPtr`

Returns the type of bump. Possible values are:

Marked

Generic

FromLef

Parameters

<i>bumpPtr</i>	Address of bump.
----------------	------------------

Command Order

Use this command after importing the design and loading an I/O file.

dbBusBitArr

dbBusBitArr *busPtr*

Returns the address of the specified bus bit array.

Parameters

<i>busPtr</i>	Address of bus.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 4> dbBusBitArr [dbNetBus [dbGetNetByName ADDR\[0\]]]  
0x5707c30
```

The address of the bus bit array is 0x5707c30.

dbBusBitBus

dbBusBitBus *busPtr*

Returns address of the master. The command returns 0x0 if there is no master.

Parameters

<i>busPtr</i>	Address of bus.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 8> dbBusBitBus [dbNetBus [dbGetNetByName MIF_ACS_NR\[0\]]]  
0x0
```

The specified bus bit does not have a master.

dbBusBitParent

dbBusBitParent *busPtr*

Returns the address of the parent for the specified bus bit.

Parameters

busPtr

Address of bus.

Command Order

Use this command after importing the design.

Example

```
innovus 2> dbBusBitParent [dbNetBus [dbGetNetByName MIF_ACS_NR\[0\]]]  
0x5656c664
```

The address of the bus bit parent is 0x5656c664.

dbBusParent

dbBusParent *busPtr*

Returns the address of the parent for the specified bus.

Parameters

busPtr

Address of bus.

Command Order

Use this command after importing the design.

Example

```
innovus 3> dbBusParent [dbNetBus [dbGetNetByName MIF_ACS_NR\[0\]]]  
0x4f85d68
```

The address of the bus parent is 0x4f5d68.

C

dbCellBNetList

dbCellBNetList *cellPtr*

Returns the address of a bus netlist for the specified cell. The command returns 0x0 if there is no bus netlist.

Parameters

cellPtr	Address of the cell.
---------	----------------------

Command Order

Use this command after the design is routed.

Example

```
innovus 1> dbCellBNetList $cellPtr
```

```
0x0
```

There is no bus netlist for this cell.

dbCellChangeWireGlobalStatus

dbCellChangeWireGlobalStatus *cellPtr oldStatus newStatus*

Changes the wire global status for all signal nets for the specified cell.

Parameters

cellPtr	Address of the cell.
oldStatus	0 or 1
newStatus	0 or 1

Command Order

Use this command after the design has been global routed.

dbCellChangeWireStatus

dbCellChangeWireStatus cellPtr oldStatus newStatus

Changes the wire status for all signal nets for the specified cell.

Parameters

cellPtr	Address of the cell.
oldStatus	0 or 1.
newStatus	0 or 1.

Command Order

Use this command after routing the design.

dbCellClass

dbCellClass *cellPtr*

Returns the address of the CLASS of the specified cell. For information on CLASS, see the *LEF/DEF Language Reference*.

Parameters

cellPtr	Address of the cell.
---------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 27>dbCellClass $cellPtr  
0x104b5c48
```

The address of the class for this cell is 0x104b5c48.

dbCellCongestionView

dbCellCongestionView *cellPtr*

Returns the congestion view (Trial Route, NanoRoute, or WRoute) of the specified cell.

Parameters

cellPtr	Address of the cell.
---------	----------------------

Command Order

Use this command after routing the design.

Example

```
innovus 18> set cellPtr [dbgTopCell]  
0xf930aa8  
  
innovus 19> dbCellCongestionView $cellPtr  
dbcTrialRouteCongestionView
```

The congestion view is the result of using Trial Route.

dbCellDim

```
dbCellDim cellPtr
```

Returns the x and y dimensions of the specified cell.

Parameters

cellPtr	Address of the cell.
---------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 27> set instPtr [dbGetInstByName SH15/I493/XU2]  
0x104b5c48
```

```
innovus 29> set cellPtr [dbInstCell $instPtr]
```

```
0xc8aa078
```

```
dbCellDim $cellPtr
```

```
54000 27000
```

This cell is 54 µm wide and 27 µm tall.

dbCellDoNotFlatten

```
dbCellDoNotFlatten cellPtr
```

Reports whether the specified cell is flattened. The command returns 1 if the cell is flattened and 0 if it is not. This command works in conjunction with `dbCellHierCell` to reduce the number of trees expanded for multiple instantiations.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus> dbCellDoNotFlatten $cellPtr
```

```
0
```

dbCellDontKnowIsFlat

```
dbCellDontKnowIsFlat cellPtr
```

Returns 0 if it is not known whether the specified cell has been flattened.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

dbCellFixWireTermLoops

```
dbCellFixWireTermLoops cellPtr
```

Deletes all multiple wire terminal connections from the specified cell. A multiple wire terminal connection occurs when more than one wire of one net is connected to a single pin on a terminal.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after routing the design.

Example

```
dbCellFixWireTermLoops $cellPtr
```

dbCellFlattenedNrBlock

dbCellFlattenedNrBlock *cellPtr*

Returns the number of blocks flattened during design import for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 2> dbCellFlattenedNrBlock [dbgTopCell]
```

43

This cell has 43 blocks that have been flattened.

dbCellFlattenedNrGate

dbCellFlattenedNrGate *cellPtr*

Returns the number of gates flattened during design import for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 2> dbCellFlattenedNrGate [dbgTopCell]  
237
```

This cell has 237 gates that have been flattened.

dbCellFlattenedNrInst

`dbCellFlattenedNrInst cellPtr`

Returns the number of instances flattened during design import for the specified cell.

Parameters

<code><i>cellPtr</i></code>	Address of the cell.
-----------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 2> dbCellFlattenedNrInst [dbgTopCell]  
59129
```

This cell has 59,129 instances that have been flattened.

dbCellFlattenedNrIo

dbCellFlattenedNrIo *cellPtr*

Returns the number of I/Os flattened during design import for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 2> dbCellFlattenedNrIo [dbgTopCell]
```

```
302
```

This cell has 302 I/Os that have been flattened.

dbCellFlattenedNrStdCell

dbCellFlattenedNrStdCell *cellPtr*

Returns the number of standard cells flattened during design import for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 2> dbCellFlattenedNrStdCell [dbgTopCell]  
58941
```

This cell has 58,941 standard cells that have been flattened.

dbCellFlattenedNrStor

dbCellFlattenedNrStor *cellPtr*

Returns the number of storage cells flattened during design import for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 2> dbCellFlattenedNrStor [dbgTopCell]  
1824
```

This cell has 1,824 storage cells that have been flattened.

dbCellFootPrintName

dbCellFootPrintName *cellPtr*

Returns the footprint for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 4> set instPtr [dbGetInstByName SH15/I535]  
0x104b5330  
encounter 6> set cellPtr [dbInstCell $instPtr]  
0xc872448  
innovus 7> dbCellFootPrintName $cellPtr  
MUX21H
```

The footprint for instance SH15/I535 is MUX21H.

dbCellFPlan

dbCellFPlan *cellPtr*

Returns the address of the floorplan for the specified cell. The command returns `0x0` if the floorplan does not exist.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus >dbCellFPlan $cellPtr  
0xd4da4c0
```

The address of floorplan for the specified cell is 0xd4da4c0.

dbCellFTermArr

```
dbCellFTermArr cellPtr
```

Returns the address of the array of terminals for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> dbCellFTermArr [dbgTopCell]  
0xfb09145
```

The address of the f-term array is 0xfb09145.

dbCellFTermList

`dbCellFTermList cellPtr`

Returns the address of the list of terminals for the specified cell.

Parameters

<code>cellPtr</code>	Address of the cell.
----------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 176> dbCellFTermList [dbgTopCell]  
0xfb091e0
```

The address of list of f-terms is 0xfb091e0.

dbCellGlobalWireAddVias

dbCellGlobalWireAddVias *cellPtr*

Changes all global wires, from NanoRoute, on all signal nets, that span multiple layers into wires and vias. For example, wires that go directly from *METAL1* to *METAL5* are split into *METAL1* to *METAL2*, wires and vias, *METAL2* to *METAL3* wires and vias, and so forth.

Parameters

<i>cellPtr</i>	Address of the cell
----------------	---------------------

Command Order

Use this command after globally routing the design using NanoRoute.

dbCellGlobalWireToUsableWire

dbCellGlobalWireToUsableWire *cellPtr*

Changes all global routes, from NanoRoute, for all signal nets for the specified cell so they can be usable in Innovus for extraction, save, restore, and so forth. However, they cannot be read back into NanoRoute for detail routing.

Parameters

<i>cellPtr</i>	Address of the cell
----------------	---------------------

Command Order

Use this command after globally routing the design using NanoRoute.

dbCellHasInternalPower

```
dbCellHasInternalPower cellPtr
```

Reports if internal power data is present on cell. The command returns `1` if power data is present and `0` if not.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 10> dbCellHasInternalPower [cellPtr]
```

dbCellHasIoRing

```
dbCellHasIoRing cellPtr
```

Reports whether the specified cell has an I/O ring. The command returns `1` if there is an I/O ring and `0` if there is not.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 10> dbCellHasIoRing [dbgTopCell]  
0
```

This design does not have an I/O ring.

dbCellHasLeakagePower

dbCellHasLeakagePower *cellPtr*

Reports if leakage power group or `cell_leakage_power` attribute is present on the cell. The command returns 1 if leakage power group or `cell_leakage_power` attribute is present and 0 if not.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 10> dbCellHasLeakagePower [cellPtr]
```

dbCellHierCell

dbCellHierCell *cellPtr*

Returns the address of the hierarchical cell for the specified cell. Use this command when there are multiple instantiations, to eliminate the need to expand all instantiations.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 11> dbCellHierCell [dbgTopCell]  
0xecfb7c0
```

The address of the hierarchical address is 0xecfb7c0.

dbCellHInst

dbCellHInst *cellPtr*

Returns the address of the specified hierarchical instance. The command returns 0x0 if the cell is not a hierarchical instance.

Parameters

<i>cellPtr</i>	Address of the cell
----------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 4> set instPtr [dbGetInstByName SH15/I535]  
0x104b5330  
innovus 6> set cellPtr [dbInstCell $instPtr]  
0xc872448  
innovus 8> dbCellHInst $cellPtr  
0x0
```

The instance SH15/I535 is not a hierarchical instance.

dbCellIGLen

dbCellIGLen *cellPtr*

Returns the length of the instance grid for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 13> dbCellIGLen $cellPtr  
54
```

The instance grid length is 54.

dbCellInstList

```
dbCellInstList cellPtr
```

Returns the address of the instance list for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 19> dbCellInstList [dbgTopCell]  
0xe9d7abc
```

The address of the instance list is 0xe9d7abc.

dbCellIoList

```
dbCellIoList cellPtr
```

Returns the address of the I/O list for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 19> dbCellIoList [dbgTopCell]  
0xe9d7abc
```

The address of the I/O list is 0xe9d7abc.

dbCellName

```
dbCellName cellPtr
```

Returns the cell (master) name for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 23> set cellPtr [dbInstCell $instPtr]
```

```
0xc872448
```

```
innovus 24> dbCellName $cellPtr
```

```
MUX21HP
```

The cell name is MUX21HP.

dbCellNetList

```
dbCellNetList cellPtr
```

Returns the address of the netlist for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 25> dbCellNetList [dbgTopCell]
```

```
0xdbac2b8
```

The address of the netlist is 0xdbac2b8.

dbCellNext

dbCellNext *cellPtr*

Returns the address of the next cell. The command returns 0x0 if there is none.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
encounter 23> set cellPtr [dbInstCell $instPtr]  
0xc872448  
encounter 26> dbCellNext $cellPtr  
0xc8727c8
```

dbCellNrBidi

dbCellNrBidi *cellPtr*

Returns the number of bidirectional ports in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
encounter 6> dbCellNrBidi $cellPtr
```

```
145
```

There are 145 bidirectional signals in this design.

dbCellNrBlock

```
dbCellNrBlock cellPtr
```

Returns the number of blocks in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
innovus 5> dbCellNrBlock $cellPtr
```

```
0
```

There are no blocks in this design.

dbCellNrFlatInst

dbCellNrFlatInst *cellPtr*

Returns the number of instances in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]  
0xf936468  
innovus 5> dbCellNrFlatInst $cellPtr  
59138
```

There are 59,138 instances in this design.

dbCellNrFTerm

dbCellNrFTerm *cellPtr*

Returns the number of f-terms in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]  
0xf936468  
innovus 7> dbCellNrFTerm $cellPtr  
189
```

There are 189 f-terms in this design.

dbCellNrGate

```
dbCellNrGate cellPtr
```

Returns the number of gates in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]  
0xf936468  
innovus 5> dbCellNrGate $cellPtr  
2394
```

There are 2,394 gates in this design.

dbCellNrHInsts

`dbCellNrHInsts cellPtr`

Returns the number of modules in the specified cell.

Parameters

<code>cellPtr</code>	Address of the cell.
----------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]  
0xf936468  
innovus 5> dbCellNrHInsts $cellPtr  
428
```

There are 428 modules in this design.

dbCellNrInput

dbCellNrInput *cellPtr*

Returns the number of inputs to the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
innovus 4> dbCellNrInput $cellPtr
```

```
15
```

There are 15 inputs in this cell.

dbCellNrInst

dbCellNrInst *cellPtr*

Returns the number of instances in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
innovus 8> dbCellNrInst $cellPtr
```

```
59129
```

There are 59,129 instances in this design.

dbCellNrIo

```
dbCellNrIo cellPtr
```

Returns the number of I/Os in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
innovus 5> dbCellNrIo $cellPtr
```

```
302
```

There are 302 I/Os in this design.

dbCellNrNet

```
dbCellNrNet cellPtr
```

Returns the number of nets in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
innovus 9> dbCellNrNet $cellPtr
```

```
71868
```

There are 71,868 nets in this design.

dbCellNrPGFTerm

```
dbCellNrPGFTerm cellPtr
```

Returns the number of power and ground f-terms in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]  
0xf936468  
innovus 11> dbCellNrPGFTerm $cellPtr  
0
```

There are no power and ground f-terms in this cell.

dbCellNrPhysicalInst

```
dbCellNrPhysicalInst cellPtr
```

Returns the number of instances not defined in the netlist, such as filler cells, I/O pads, and I/O filler cells.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]  
0xf936468  
innovus 12> dbCellNrPhysicalInst $cellPtr  
0
```

There are 0 glue cells in this design.

dbCellNrRef

```
dbCellNrRef cellPtr
```

Returns the number of instantiations of the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 13> dbCellNrRef $cellPtr  
1
```

There is one instantiation of the specified cell.

dbCellNrRow

dbCellNrRow *cellPtr*

Returns the number of rows required by the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> dbCellNrRow $cellPtr
```

```
1
```

This cell requires one row.

dbCellNrStdCell

dbCellNrStdCell *cellPtr*

Returns the number of standard cells in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
innovus 5> dbCellNrStdCell $cellPtr
```

```
58194
```

There are 58,194 standard cells in this design.

dbCellNrStor

```
dbCellNrStor cellPtr
```

Returns the number of storage cells in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
innovus 5> dbCellNrStor $cellPtr
```

```
2951
```

There are 2,951 storage cells in this design.

dbCellNrSym

`dbCellNrSym cellPtr`

Returns the number of symbols in the specified cell's symbol table. The specified cell must be a top cell. The symbols are associated with hierarchical instances (modules); that is, `dbCellNrSym` returns the number of hierarchical instances based on Verilog hierarchy (not grouping).

Parameters

<code>cellPtr</code>	Address of the cell.
----------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> dbCellNrSym [dbgTopCell]  
167
```

There are 167 modules in this design.ct.

dbCellOrigCell

`dbCellOrigCell cellPtr`

Returns the address of the origin of the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 14> set cellPtr [dbgTopCell]  
0xf936468  
  
innovus 18> set cellOrigPtr [dbCellOrigCell $cellPtr]  
0xed00de8
```

The variable `cellOrigPtr` contains the address of the origin of the cell.

dbCellOrigin

`dbCellOrigin cellPtr`

Returns the origin of the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 14> set cellPtr [dbgTopCell]  
0xf936468
```

```
innovus 20> dbCellOrigin $cellPtr
```

```
0 0
```

The origin is (0, 0) for this design.

dbCellPGFTermArr

```
dbCellPGFTermArr cellPtr
```

Returns the address of the power and ground f-terms in the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 14> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
innovus 21> set PGTermArrPtr [dbCellPGFTermArr $cellPtr]
```

```
0xfb169f
```

dbCellPhysicalNetList

```
dbCellPhysicalNetList cellPtr
```

Returns the address of the list of the physical nets not in the netlist for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 56>dbCellPhysicalNetList [dbgTopCell]  
0x8720023
```

The address of the physical netlist is 0x8720023.

dbCellPrev

```
dbCellPrev cellPtr
```

Returns the address of the previous cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
set cellPtr [dbInstCell $instPtr]  
0xc872448  
innovus 28> dbCellPrev $cellPtr  
0xc872288
```

The address of the previous cell is 0xc872288.

dbCellPtn

`dbCellPtn cellPtr`

Returns the address of the partitions for the specified cell.

Parameters

<code>cellPtr</code>	Address of the cell.
----------------------	----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
innovus 29> dbCellPtn [dbgTopCell]  
0xcafaad8
```

The address is 0xcafaad8.

dbCellPtnList

dbCellPtnList *cellPtr*

Returns the address of the list of partitions for the specified cell. If no partitions have been specified, the command returns 0x0.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
innovus 34> dbCellPtnList [dbgTopCell]  
0xcafac68
```

The address of the partition list is 0xcafac68.

dbCellRepCell

dbCellRepCell *cellPtr*

Returns the address of the representative cell pointed to by the specified cell. The representative cell is the representative for all (timing library) "sibling" cells. Applications that need to see only one cell (for example, the LEF library cell), should see only the representative cell, because sibling cells are created for different timing libraries (min/max) that might not have all of the required (LEF) information.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 28> dbCellRepCell [dbGetCellByName OR4X2]  
0xc872288
```

The address of the representative cell is 0xc872288.

dbCellRouteAlgUsed

```
dbCellRouteAlgUsed cellPtr
```

Returns the algorithm used to route the specified cell. Possible algorithms are

dbcTrialRouter
dbcNanoRouter
dbcWarpRouter

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after routing the design.

Example

```
innovus 14> set cellPtr [dbgTopCell]  
0xf936468  
innovus 22> dbCellRouteAlgUsed $cellPtr  
dbcTrialRouter
```

dbCellSite

```
dbCellSite cellPtr
```

Returns the site address for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
encounter 450>set cellPtr1 [dbGetCellByName BUFX1]  
0xf177760  
encounter 451>dbCellSite $cellPtr1  
0x8923042
```

The site address for cell BUFX1 is 0x8923042.

dbCellSNetList

dbCellSNetList *cellPtr*

Returns the address of the special nets for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 451>dbCellSNetList [dbgTopCell]  
0x892304
```

The address of the special nets is 0x892304.

dbCellSubClass

dbCellSubClass *cellPtr*

Returns the subclass address for the specified cell. For information on subclasses, see the *LEF/DEF Language Reference*.

Parameters

<i>cellPtr</i>	Address of specified cell.
----------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 89>dbCellSubClass [dbgTopCell]  
0x8690270
```

dbCellSymArr

dbCellSymArr *cellPtr*

Returns the address of the symbols array, that is, hierarchical instances (modules). The specified cell must be the top cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

dbCellTechSite

dbCellTechSite *cellPtr*

Returns the tech site address for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
innovus 450>set cellPtr1 [dbGetCellByName BUFX1]
```

```
0xf177760
```

```
innovus 451>dbCellTechSite $cellPtr1
```

```
0x8923042
```

The tech site address for cell BUFX1 is 0x8923042.

dbCellTimeLib

dbCellTimeLib *cellPtr*

Returns the timing library address for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 450>set cellPtr1 [dbGetCellByName BUFX1]
```

```
0xf177760
```

```
innovus 451>dbCellTimeLib $cellPtr1
```

```
0x8923042
```

The timing library address for cell BUFX1 is 0x8923042.

dbCellType

```
dbCellType cellPtr
```

Returns the cell type for the specified cell. The possible cell types are

```
dbcGateCell 1
```

```
dbcBlockCell 4
```

```
dbcSuperCell 7
```

```
dbcSuperBlockCell 8
```

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 14> set cellPtr [dbgTopCell]
```

```
0xf936468
```

```
innovus 26> dbCellType $cellPtr
```

```
dbcSuperCell
```

The cell is a super cell.

dbCellVoltage

```
dbCellVoltage cellPtr
```

Returns the voltage for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after loading the design.

Example

```
innovus 14> set cellPtr [dbGetCellByName YFD1]
```

```
0xf936468
```

```
innovus 26> dbCellVoltage $cellPtr
```

```
2.2
```

The cell voltage is 2.2.

dbChangeInstCell

`dbChangeInstCell instPtr cellPtr`

Determines whether the specified change was successful. The command returns 1 if the change was successful and 0 if it was not.

Parameters

<code>instPtr</code>	Address of instance.
<code>cellPtr</code>	Address of cell.

Command Order

Use this command after importing the design.

Example

`dbChangeInstCell [dbGetInstByName DTMF/i8003] [dbGetCellByName BUF]`

Returns:

1

The change was successful.

dbCheckCell

```
dbCheckCell cellPtr
```

Reports the status of the design for the specified cell.

Parameters

cellPtr	Address of the cell.
---------	----------------------

Note: The f-terms can be IO Pins as viewed by the top cell and also pins on a LEF library cell, as in BUFX2.

Command Order

Use this command after importing the design.

Example

```
cellPtr [dbgTopCell]
```

Returns:

```
0xf936468
```

Then:

```
dbCheckCell $cellPtr
```

Returns:

```
**WARN: Dangling net SH1/I129/SH1/I585/I1/U1/PREADCYC_reg_QN (has no loads)
**WARN: Dangling net SH1/I129/SH1/I585/I1/U1/DPREAD_reg_Q (has no loads)
0 fterms of cell TOPCHIP_SP are unused
63 nets of cell TOPCHIP_SP are floating
0 nets of cell TOPCHIP_SP are degenerate
9860 nets of cell TOPCHIP_SP are dangling
```

There are 63 floating nets and 9,860 dangling nets.

dbCleanCellFlag

dbCleanCellFlag *flagName* {0 | 1}

Sets the specified flag to 0 or 1 for all cells with the specified flag.

Parameters

<i>flagName</i>	Available values: Marked, Marked2, Marked3, Marked4
{0 1}	

Command Order

Use this command after importing the design.

Example

dbCleanCellFlag marked2 0

Returns:

*** Total 515 cells with marked 2 set to 0

dbCleanInstFlag

dbCleanInstFlag *flagName* {0 | 1}

Sets the specified flag to 0 or 1 for all instances with the specified flag.

Parameters

<i>flagName</i>	Available values: IPOed, Marked, Marked2, Marked3, Marked4, Hilite
{0 1}	

Command Order

Use this command after importing the design.

Example

```
dbCleanInstFlag Hilite 1
```

Returns:

```
*** Total 5981 insts with flag Hilite set to 1
```

dbCleanNetFlag

```
dbCleanNetFlag flagName {0 | 1}
```

Sets the specified flag to 0 or 1 for all nets with the specified flag.

Parameters

<i>flagName</i>	Available values: Critical, IPOed, PostRouteSiFix, TrialRouted, RouteDirty, Marked, Marked2, Marked3, Marked4, Hilite
{0 1}	

Command Order

Use this command after importing the design.

Example

```
dbCleanNetFlag IPOed 1
```

Returns:

```
*** Total 6287 nets with flag IPOed set to 1
```

dbCleanTermFlag

```
dbCleanTermFlag flagName {0 | 1}
```

Sets the specified flag to 0 or 1 for all terms with the specified flag.

Parameters

<i>flagName</i>	Available values: IPOed, Marked, Marked2, Marked3, Marked4, Hilite
{0 1}	

Command Order

Use this command after importing the design.

Example

```
dbCleanTermFlag marked2 0
```

Returns:

```
*** Total 515 terms with marked 2 set to 0
```

dbClearCellFTermMarkers

`dbClearCellFTermMarkers cellPtr`

Clears all the markers for f-terms, instances, nets, terminals, and wires from the specified cell. Markers are temporary markers for applications.

Parameters

<code><i>cellPtr</i></code>	Address of the cell.
-----------------------------	----------------------

Command Order

Use this command after importing the design.

dbClearCellInstMarkers

`dbClearCellInstMarkers cellPtr`

Clears all hierarchical instance markers from the specified cell.

Parameters

<code><i>cellPtr</i></code>	Address of the cell.
-----------------------------	----------------------

Command Order

Use this command after importing the design.

dbClearCellNetMarkers

`dbClearCellNetMarkers cellPtr`

Clears all net markers from the specified cell.

Parameters

<code>cellPtr</code>	Address of the cell.
----------------------	----------------------

Command Order

Use this command after importing the design.

dbClearCellTermMarkers

`dbClearCellTermMarkers cellPtr`

Clears all terminal markers from the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

dbClearCellWireMarkers

`dbClearCellWireMarkers cellPtr`

Clears all wire markers from the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

dbClearHInstMarkers

`dbClearHInstMarkers cellPtr`

Clears all hierarchical instance markers from the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

dbClockName

`dbClockName clockPtr`

Returns the name of the clock.

Parameters

<i>clockPtr</i>	Address of clock.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachHeadClock [dbgHead] clockPtr {  
    set name [dbClockName $clkPtr]  
    puts "name = $name"  
}
```

Returns:

name = MCK

dbClockNext

dbClockNext *clockPtr*

Returns the address of the next clock and 0x0 if there is none.

Parameters

<i>clockPtr</i>	Address of clock.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

dbClockNext \$clkPtr

Returns:

0x0

There is no other clock.

dbClockPrev

dbClockPrev *clockPtr*

Returns the address of the previous clock and 0x0 if there is none.

Parameters

<i>clockPtr</i>	Address of clock.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbClockPrev $clkPtr
```

Returns:

0x0

dbClockRate

```
dbClockRate clockPtr
```

Returns the clock rate.

Parameters

<i>clockPtr</i>	Address of clock.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachHeadClock [dbgHead] clockPtr {
```

```
set rate [dbClockRate $clkPtr]  
puts "rate = $rate"  
}
```

dbCloneListInst

dbCloneListInst *cloneListPtr*

Returns the address of the list of instances for the clone.

Parameters

<i>cloneListPtr</i>	Address of clone list.
---------------------	------------------------

Command Order

Use this command after importing a design with partition clones.

Example

```
dbForEachPtnCloneList [dbGetPtnByName sheet1] cloneListPtr {  
    set ret [dbCloneListInst $cloneListPtr]  
    puts "return $ret"  
}
```

Returns:

```
return 0x916c3ac
```

dbCloneListOrient

`dbCloneListOrient cloneListPtr`

Returns the orientations of the specified clone.

Parameters

<code>cloneListPtr</code>	Address of clone list.
---------------------------	------------------------

Command Order

Use this command after importing a design with partition clones.

Example

```
dbForEachPtnCloneList [dbGetPtnByName sheet1] cloneListPtr {  
    set ret [dbCloneListOrient $cloneListPtr]  
    puts "orient = $ret"  
}
```

Returns:

```
orient = dbcR0
```

dbComputeHInstPDensity

`dbComputeHInstPDensity hinstPtr`

Returns the 'P' density of the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance
-----------------	----------------------------------

Command Order

Use this command after importing the design.

Example

```
dbComputeHInstPDensity $hinstPtr
```

Returns:

0.72

dbConstraintBox

```
dbConstraintBox constraintPtr
```

Returns the box of the constraint.

Parameters

<i>constraintPtr</i>	Address of the constraint.
----------------------	----------------------------

Command Order

Use this command after importing the design and applying a constraint.

Example

```
dbConstraintBox $constraintPtr
```

Returns:

431000 3644000 6693714 6749000

The box is (431.00 3644.00) (6693.714 6749.000).

dbConstraintHInst

dbConstraintHInst *constraintPtr*

Returns the address of the constraint of the hierarchical instance.

Parameters

<i>constraintPtr</i>	Address of the constraint.
----------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
set constraintPtr [dbHInstConstraint $objPtr]
```

Returns:

0xce51780

The address of the constraints is 0xce51780.

dbConstraintNext

dbConstraintNext *constraintPtr*

Returns the address of the next constraint. If there are no more constraints, the command returns

0x0.

Parameters

<i>constraintPtr</i>	Address of the constraint.
----------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
dbConstraintNext $constraintPtr
```

Returns:

0x0

There are no more constraints in this design.

dbConstraintNextInFPlan

```
dbConstraintNextInFPlan constraintPtr
```

Returns the address of the next constraint of the floorplan. If there are no more constraints, the command returns 0x0.

Parameters

<i>constraintPtr</i>	Address of the constraint.
----------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
dbConstraintNextInFPlan $constraintPtr
```

Returns:

0x0

There are no more constraints in this design.

dbConstraintType

```
dbConstraintType constraintPtr
```

Returns the constraint type. If there are no constraints, the command returns 0x0. Possible values are

dbcNoConstraint

dbcCluster

dbcGuide

dbcRegion

dbcFence

Parameters

<i>constraintPtr</i>	Address of the constraint.
----------------------	----------------------------

Command Order

Use this command after importing the design.

Example

(with a fence selected)

```
innovus 37> set s [dbHeadSelList]  
0x139e88b8  
innovus 38> set objPtr [dbSelPtr $s]  
0xcd6a578  
encounter 39> set obj_type [dbObjType $objPtr]  
dbcObjHInst  
innovus 40> set constraintPtr [dbHInstConstraint $objPtr]  
0xce51780  
innovus 41> set constraintType [dbConstraintType $constraintPtr]  
dbcFence
```

The constraint is a fence.

dbCopyNetWires

`dbCopyNetWires fromNetPtr toNetPtr`

Copies the specified net wires (but does not change the netlist).

Parameters

<code>fromNetPtr</code>	Address of the `from' net.
<code>toNetPtr</code>	Address of the `to' net.

Command Order

Use this command after routing the design.

dbCountDrc

dbCountDrc

Returns the total number of DRC markers in the database.

Command Order

Use this command after importing the design.

dbCountFTermFlag

dbCountFTermFlag *flagName*

Returns the number of f-terms with the specified flag. The f-terms can be IO Pins as viewed by the top cell and also pins on a LEF library cell, as in BUFX2.

Parameters

flagName

Available values: Sel, Hilite, Marked, Marked2, Marked3, Marked4

Command Order

Use this command after importing the design.

Example

dbCountFTermFlag Hilite

Returns:

*** Total 0/57 fterms with flag Hilite marked as 1.

0

There are 57 f-terms with 0 Hilite flags.

dbCountInstFlag

dbCountInstFlag *flagName*

Returns the number of instances with the specified flag.

Parameters

flagName

Flagname: Sel, Hilite, Marked, Marked2, Marked3, Marked4

Command Order

Use this command after importing the design.

dbCountNetFlag

dbCountNetFlag *flagName*

Returns the number of nets with the specified flag.

Parameters

flagName

FlagName: Sel, Hilite, Marked, Marked2, Marked3, Marked4

Command Order

Use this command after importing the design.

dbCountTermFlag

dbCountTermFlag <flagName> <1 | 0>

Returns the number of cell instance terminals and/or the DRC markers with the specified flag.

Parameters

flagName

Available values: Sel, Hilite, Marked, Marked2, Marked3, Marked4

Command Order

Use this command after importing the design.

dbCreateConstraint

dbCreateConstraint *hinstPtr* *llx lly urx ury*

Creates a constraint at the coordinates specified.

Parameters

<i>hinstPtr</i>	Address of HInst.
<i>llx</i>	Lower left x ordinate.
<i>lly</i>	Lower left y ordinate.
<i>urx</i>	Upper right x ordinate.
<i>ury</i>	Upper right y ordinate.

Command Order

Use this command after importing the design.

dbCreateMarker

```
dbCreateMarker
{-box llx lly urx ury | -poly {x1 y1} {x2 y2}... } 
[-layer layerName]
[-tool toolName]
[-type typeName]
[-subtype subTypeName]
```

Creates rectangle and polygon markers for violations in the database, and imports DRC markers generated by other software applications. The function returns an Id, which can be used later for adding or deleting to a set of highlighted markers.

Note: The software does not immediately display newly created markers. You must refresh the main window in order to see the markers.

The newly created markers are displayed using the color white. To change the color, you must add the marker to a highlighted set.

Parameters

```
-box llx lly urx ury
```

	Specifies the area of the bounding box of the rectangle marker.
<code>-poly {x1 y1} {x2 y2}...</code>	
	Specifies the area of the bounding box of the polygon marker.
<code>-layer layerName</code>	Specifies the layer on which the marker shape is to be created.
<code>-tool toolName</code>	Specifies the source software application. Note: This parameter is not optional for PVS; you must specify <code>PVS</code> .
<code>-type typeName</code>	Specifies the marker type.
<code>-subType subTypeName</code>	
	Specifies the marker subtype.

dbCreateObstruct

`dbCreateObstruct fplanPtr llx lly urx ury`

Creates an placement blockage at the coordinates specified and returns the address of the new obstruct.

Parameters

<code>fplanPtr</code>	Address of floor plan.
<code>llx</code>	Lower left x ordinate.
<code>lly</code>	Lower left y ordinate.
<code>urx</code>	Upper right x ordinate.
<code>ury</code>	Upper right y ordinate.

Command Order

Use this command after importing the design.

Example

```
dbCreateObstruct [dbHeadFPlan] 200 200 400 400
```

Returns:

```
0x7820b443
```

The obstruction is created at (200,200) (400,400).

dbCreateRestrictedVia

```
dbCreateRestrictedVia layer ulx uly uhx uhy llx lly lhx lhy cut_lx cut_ly  
cut_width cut_height cut_xtimes cut_ytimes cut_xpitch cut_ypitch
```

Creates a restricted via. A restricted via cell is a regular via cell that has one upper metal rectangle and one lower metal rectangle. If \$vc is {}, then the creation has failed.

Parameters

<i>layer</i>	Upper metal layer (Integer METAL1->1, METAL2->2,..., METAL15->15)
<i>ulx</i>	Upper layer metal lower left x ordinate.
<i>uly</i>	Upper layer metal lower left y ordinate.
<i>uhx</i>	Upper layer metal upper right x ordinate.
<i>uhy</i>	Upper layer metal upper right y ordinate.
<i>llx</i>	Lower layer metal lower left x ordinate.
<i>lly</i>	Lower layer metal lower left y ordinate.
<i>lhx</i>	Lower layer metal upper right x ordinate.

<i>lhy</i>	Lower layer metal upper right y ordinate.
<i>cut_1x</i>	Cut placement lower left x ordinate.
<i>cut_1y</i>	Cut placement lower left y ordinate.
<i>cut_width</i>	Cut width 'Y' dimension.
<i>cut_height</i>	Cut height 'X' dimension.
<i>cut_xtimes</i>	Number of cuts in the x dimension (array).
<i>cut_ytimes</i>	Number of cuts in the y dimension (array).
<i>cut_xpitch</i>	Pitch between cuts in the x dimension.
<i>cut_ypitch</i>	Pitch between cuts in the y dimension.

Command Order

Use this command after importing the design.

dbCreateScreen

```
dbCreateScreen fplanPtr l1x l1y urx ury cap
```

Creates a placement screen at the specified coordinates and returns the address.

Parameters

<i>fplanPtr</i>	Address of floor plan.
<i>l1x</i>	Lower left x ordinate.
<i>l1y</i>	Lower left y ordinate.

<i>urx</i>	Upper right x ordinate.
<i>ury</i>	Upper right y ordinate.
<i>cap</i>	Number

Command Order

Use this command after importing the design.

Example

```
dbCreateScreen [dbHeadFPlan] 200 200 400 400 0.1
```

Returns:

0xa209510

A new screen is created at (200,200) (400,400) with a capacitance of 0.1.

dbCreateSpareGateGroup

```
dbCreateSpareGateGroup
```

Creates a group of spare gates, if the spare gates are identified in the Innovus software.

Command Order

Use this command after importing the design.

dbCreateText

```
dbCreateText  
-message message
```

`-location x y`

Creates a text label. This function returns an Id that can be used later to delete the text label.

Note: The software does not automatically display newly created text labels. You must refresh the main window first, in order to see them.

Parameters

<code>-message message</code>	Specifies the text. You must enclose the text in quotation marks.
<code>-location x y</code>	Specifies the beginning coordinate of the text.

dbCreateVia

`dbCreateVia net viacell x y [shape]`

Creates a via at the specified x y of the specified shape.

Parameters

<code>net</code>	Address of the net.
<code>viaCell</code>	Address of the via cell.
<code>x</code>	x ordinate
<code>y</code>	y ordinate
<code>shape</code>	NOTYPE, RING, STRIPE, FOLLOWPIN, IOWIRE, COREWIRE, BLOCKWIRE, FILLWIRE, PADRING, BLOCKRING

Command Order

Use this command after importing the design and creating a group.

Example

```
set net [dbGetNetByName vdd]  
set via [dbCreateVia $net $vc $x $y]
```

dbCreateWire

`dbCreateWire net lx ly hx hy {layerPtr | layer} dir [shape]`

Creates a wire.

Parameters

<i>net</i>	Address of the net.
<i>lx</i>	Lower left x ordinate of wire rectangle
<i>ly</i>	Lower left y ordinate of wire rectangle
<i>hx</i>	Upper right x ordinate of wire rectangle
<i>hy</i>	Upper right y ordinate of wire rectangle
<i>layerPtr</i> <i>layer</i>	Specifies the metal layer on which to create the wire. You can specify a layer pointer, a LEF layer name, or the z number for the layer (for example, 0= M0, 1= METAL1).
<i>dir</i>	0 for horizontal, 1 for vertical
<i>shape</i>	NOTYPE, RING, STRIPE, FOLLOWPIN, IOWIRE, COREWIRE, BLOCKWIRE, FILLWIRE, PADRING, BLOCKRING

Command Order

Use this command after importing the design.

Example

```
set net [dbGetNetByName vdd]  
set wire [dbCreateWire $net 2000 2000 2000 4000 3 0]
```

dbCurrentDefaultLibraryContext

dbCurrentDefaultLibraryContext

Returns the current default library context being used by the Innovus software.

Command Order

Use this command after importing the design.

dbCustomLayerId

dbCustomLayerId *customerLayerPtr*

Returns the ID for the specified custom layer.

Parameters

<i>customLayerPtr</i>	Address of custom layer.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating a group.

Example

```
dbCustomLayerId $customLayerPtr
```

Returns:

1

The ID for the specified layer is 1.

dbCustomLayerName

```
dbCustomLayerName customLayerPtr
```

Returns the name for the specified custom layer.

Parameters

<i>customLayerPtr</i>	Address of custom layer.
-----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbCustomLayerName $customLayerPtr
```

Returns:

VDDMET3

The name of the specified layer is VDDMET3.

D

dbDBUToMicrons

dbDBUToMicrons *integer_in_dbu*

Converts integers in database units to microns.

Parameters

<i>integer_in_db</i>	Integer in database units.
----------------------	----------------------------

Command Order

Use this command after starting Innovus.

Example

```
set corellx [dbDBUToMicrons [lindex $box 0]]  
dbDBUToMicrons 1212121
```

Returns:

1212.121

The database number 12121211 is converted to 1212.121 microns.

dbDelCellObs

```
dbAddCellObs  
-cell cell_ptr  
-layer layername  
-box box
```

Supports selection of macro and standard cell pins for Jenga via enhancement.

Parameters

-cell	Name of cell/block partition of the object for which the obstruction is added.
-layer	Name of the obstruction layer. It will be a LEF layer name
-box	Name of the co-ordinates of the obstruction in the following format with reference to the cell origin in db units: <code>llx1 lly1 urx2 ury2</code>

Command Order

Use this command after importing the design.

Example

```
dbDelCellObs <cell_ptr> <layer_ptr> { <llx> <ll_y> <ur_x> \
<ur_y> | <box> }
```

dbDelCoverInst

```
dbDelCoverInst instPtr
```

Removes the cover macro from the specified instance.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after adding a cover macro.

Example

```
dbDelCoverInst $instPtr
```

Deletes the cover instance.

dbDeleteAreaIoRow

```
dbDeleteAreaIoRow stdRowPtr
```

Deletes the specified area I/O row.

Parameters

<i>stdRowPtr</i>	Address of stdRow.
------------------	--------------------

Command Order

Use this command after importing the design and creating at least one area I/O row.

Example

```
dbDeleteAreaIoRow $stdRowPtr
```

The specified row is deleted.

dbDeleteBuffer

```
dbDeleteBuffer instPtr
```

Deletes the specified buffer.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbDeleteBuffer $objPtr
```

The specified buffer is removed.

dbDeleteBumpGrid

```
dbDeleteBumpGrid
```

Deletes the bump grid.

Command Order

Use this command after importing the design and creating a bump grid.

Example

```
dbDeleteBumpGrid
```

The bump grid is deleted.

dbDeleteCustomObj

dbDeleteCustomObj *objPtr*

Deletes the specified object.

Parameters

<i>objPtr</i>	Address of object.
---------------	--------------------

Command Order

Use this command after importing the design and creating at least one custom object.

Example

```
dbDeleteCustomObj $objPtr
```

The custom object is deleted.

dbDeleteLayerBlk

dbDeleteLayerBlk *fplanPtr layerBlkPtr*

Deletes the specified routing blockage.

Parameters

<i>fplanPtr</i>	Address of floor plan.
<i>layerBlkPtr</i>	Address of routing blockage.

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
innovus 8> set fplanPtr [ dbCellFPlan [dbgTopCell] ]  
0x8113888  
(then select a routing layer blockage)  
innovus 11> set s [dbHeadSelList]  
0x82640c8  
innovus 12> set objPtr [dbSelPtr $s]  
0x259c87c  
innovus 14> dbDeleteLayerBlk $fplanPtr $objPtr  
innovus 15>
```

The specified layer blockage is deleted.

dbDeleteMarker

dbDeleteMarker
{-all | *markerList*}

Deletes markers in the design.

Note: The software does not immediately remove the markers from the main window. You must refresh the main window in order to see the change.

Parameters

- all	Deletes all markers in the design.
<i>markerList</i>	
	<p>Deletes the specified markers.</p> <p>When a marker is created with the dbCreateMarker command, the software returns an Id for the marker. Specify these Ids to identify which markers you want to delete.</p>

dbDeleteObj

`dbDeleteObj objPtrs`

Takes a list of database objects and deletes them from the database.

Parameters

<i>objPtrs</i>	Lists the DB object pointers and is a mandatory option.
----------------	---

Command Order

Use this command after importing the design.

dbDeleteObstruct

`dbDeleteObstruct fplanPtr obstructPtr`

Removes the specified placement obstruction from the floorplan.

Parameters

<code><i>fplanPtr</i></code>	Address of floor plan.
<code><i>obstructPtr</i></code>	Address of obstruction.

Command Order

Use this command after importing the design and creating a placement obstruction.

Example

```
innovus 8> set fplanPtr [ dbCellFPlan [dbgTopCell] ]  
0x8113888  
(then select an onstruction)  
innovus 11> set s [dbHeadSelList]  
0x82640c8  
innovus 12> set objPtr [dbSelPtr $s]  
0x259c87c  
innovus 14> dbDeleteObstruct $fplanPtr $objPtr
```

The specified obstruction is removed.

dbDeletePtnCut

`dbDeletePtnCut ptnCutPtr`

Removes the specified partition cut.

Parameters

<code><i>ptnCutPtr</i></code>	Address of the partition cut.
-------------------------------	-------------------------------

Command Order

Use this command after partitioning the design and creating a partition cut.

Example

```
dbDeletePtnCut $ptnCutPtr
```

The specified partition cut is removed.

dbDeletePtnFeed

`dbDeletePtnFeed ptnFeedPtr`

Removes the specified partition feedthrough.

Parameters

<code><i>ptnFeedPtr</i></code>	Address of the partition feedthrough.
--------------------------------	---------------------------------------

Command Order

Use this command after importing and partitioning the design and creating a partition feedthrough.

Example

```
dbDeletePtnFeed $ptnFeedPtr
```

The associated partition feeds are deleted.

dbDeletePtnPinBlk

```
dbDeletePtnPinBlk fplanPtr ptnCutPtr
```

Removes the specified partition pin blockage.

Parameters

<i>fplanPtr</i>	Address of the floorplan.
<i>ptnPinBlkPtr</i>	Address of the partition pin blockage.

Command Order

Use this command after importing and partitioning the design and creating a partition pin blockage.

Example

```
innovus 8> set fplanPtr [ dbCellFPlan [dbgTopCell] ]
```

```
0x8113888
```

```
(with a partition pin blockage selected)
```

```
innovus 11> set s [dbHeadSelList]
```

```
0x82640c8
```

```
innovus 12> set objPtr [dbSelPtr $s]
```

0x259c87c

```
innovus 14> dbDeletePtnPinBlk $fplanPtr $objPtr  
innovus 15>
```

The specified pin blockage is deleted.

dbDeletePtnPinShapes

`dbDeletePtnPinShapes ptnPtr`

Removes all pin shapes of the specified partition. The location of deleted pins is set to LLX (lower left X) and LLY (lower left Y). The layer of the deleted pins is set to 0. The status of the deleted pins is set to unplaced.

Parameters

<code><i>ptnPtr</i></code>	Address of the partition.
----------------------------	---------------------------

Command Order

Use this command after importing and partitioning the design.

Example

```
set ptnPtr [dbGetPtnByName sh1]
```

```
dbDeletePtnPinShapes $ptnPtr
```

All the pin shapes on the partition `sh1` are deleted.

dbDeletePtnPinShapesForNet

```
dbDeletePtnPinShapesForNet ptnPtr netPtr
```

Removes all pin shapes of the specified net in the specified partition. The net pin can be on any layer. The location of deleted pins is set to LLX (lower left X) and LLY (lower left Y). The layer of the deleted pins is set to 0. The status of the deleted pins is set to `unplaced`.

Parameters

<i>netPtr</i>	Address of the net.
<i>ptnPtr</i>	Address of the partition.

Command Order

Use this command after importing and partitioning the design.

Example

```
set ptnPtr [dbGetPtnByName sh1]  
set netPtr [dbGetNetByName N1]  
dbDeletePtnPinShapesForNet $ptnPtr $netPtr
```

All the pin shapes of the pins of the net `N1` of the partition `sh1` are deleted.

dbDeletePtnPinShapesForNetOnLayer

```
dbDeletePtnPinShapesForNetOnLayer ptnPtr netPtr layerPtr
```

Removes all pin shapes on the specified net pins on the specified layer in the specified partition. The location of deleted pins is set to LLX (lower left X) and LLY (lower left Y). The layer of the deleted pins is set to 0. The status of the deleted pins is set to `unplaced`.

Parameters

<i>layerPtr</i>	Address of the net.
<i>netPtr</i>	Address of the net.
<i>ptnPtr</i>	Address of the partition.

Command Order

Use this command after importing and partitioning the design.

Example

```
set ptnPtr [dbGetPtnByName sh1]  
set layerPtr [dbGetLayerByName METAL1]  
set netPtr [dbGetNetByName N1]  
dbDeletePtnPinShapesforNetOnLayer $ptnPtr $netPtr $layerPtr
```

All the pin shapes of the pins of the net `N1` in the partition `sh1` on layer `METAL1` are deleted. If the net pin is on any other layer, its shape is not deleted.

dbDeletePtnPinShapesOnLayer

```
dbDeletePtnPinShapesOnLayer ptnPtr layerPtr
```

Removes all pin shapes on the specified layer of the specified partition. The location of deleted pins is set to LLX (lower left X) and LLY (lower left Y). The layer of the deleted pins is set to 0. The status of the deleted pins is set to `unplaced`.

Parameters

<i>layerPtr</i>	Address of the partition layer.
<i>ptnPtr</i>	Address of the partition.

Command Order

Use this command after importing and partitioning the design.

Example

```
set ptnPtr [dbGetPtnByName sh1]  
set layerPtr [dbGetLayerByName METAL1]  
dbDeletePtnPinShapesOnLayer $ptnPtr $layerPtr
```

All the pin shapes on the layer 1 of the partition `sh1` are deleted.

dbDeleteRouteBlk

```
dbDeleteRouteBlk fplanPtr routeBlkPtr
```

Deletes the specified routing blockage.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>routeBlkPtr</i>	Address of routing blockage.

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
innovus 8> set fplanPtr [ dbCellFPlan [dbgTopCell] ]  
0x8113888  
(then select a routing layer blockage)  
innovus 11> set s [dbHeadSelList]  
0x82640c8  
innovus 12> set objPtr [dbSelPtr $s]  
0x259c87c  
innovus 14> dbDeleteRouteBlk $fplanPtr $objPtr  
innovus 15>
```

The specified layer blockage is deleted.

dbDeleteRouteBox

`dbDeleteRouteBox routeBoxPtr`

Deletes the specified route box.

Parameters

<code>routeBoxPtr</code>	Address of route box.
--------------------------	-----------------------

Command Order

Use this command after importing the design and creating at least one route box.

Example

```
dbDeleteRouteBox $objPtr
```

The specified route box is removed.

dbDeleteRowCluster

```
dbDeleteRowCluster rowClusterPtr
```

Deletes the specified row cluster.

Parameters

<i>rowClusterPtr</i>	Address of row cluster.
----------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one row cluster.

Example

```
dbDeleteRouteBox $rowClusterPtr
```

The specified route box is deleted.

dbDeleteScreen

```
dbDeleteScreen fplanPtr screenPtr
```

Removes the specified density screen.

Parameters

<i>fplanPtr</i>	Address of the floorplan.
<i>screenPtr</i>	Address of the density screen.

Command Order

Use this command after importing the design and creating a density screen.

Example

(with a density screen selected)

```
innovus 8> set fplanPtr [ dbCellFPlan [dbgTopCell] ]  
0x8113888  
innovus 11> set s [dbHeadSelList]  
0x82640c8  
innovus 12> set objPtr [dbSelPtr $s]  
0x259c87c  
innovus 14> dbDeleteScreen $fplanPtr $objPtr  
innovus 25>
```

The specified density screen is removed.

dbDeleteSpareGateGroup

dbDeleteSpareGateGroup

Removes a spare group.

Command Order

Use this command after importing the design and creating a spare group.

Example

```
dbDeleteSpareGateGroup
```

The spare gate group is removed.

dbDeleteStripBox

```
dbDeleteStripBox stripBoxPtr
```

Deletes a power and ground strip.

Parameters

<i>stripBoxPtr</i>	Address of the strip box.
--------------------	---------------------------

Command Order

Use this command after creating a power and ground strip.

Example

Select a power and ground strip, then issue the following command:

```
innovus 77> set s [dbHeadSelList]
```

```
0x139e88b8
```

```
innovus 78> set objPtr [dbSelPtr $s]
```

```
0x10fbe674
```

```
innovus 79> dbDeleteStripBox $objPtr
```

```
0
```

A redraw confirms that the strip is gone.

dbDeleteText

`dbDeleteText
{-all | textList}`

Deletes text labels in the design.

Note: The software does not immediately remove the text labels from the main window. You must refresh the main window in order to see the change.

Parameters

<code>-all</code>	Deletes all text labels in the design.
<code><i>textList</i></code>	Deletes specified text labels. When a text label is created with the dbCreateText command, the software returns an Id for the text label. Specify these Ids to identify which text labels you want to delete.

dbDeleteTrialRoute

`dbDeleteTrialRoute`

Deletes all routes created by Trial Route.

Command Order

Use this command after running Trial Route on the design.

Example

dbDeleteTrialRoute

All Trial Route routes are removed.

dbDelFromHInst

dbDelFromHInst *instPtr_or_hinstPtr hinstPtr*

Deletes the specified instance or hierarchical instance from the specified hierarchical instance (but does not change the netlist).

Parameters

<i>instPtr</i>	Address of the instance.
<i>hinstPtr</i>	Address of the hierarchical instance.

Command Order

Use this command after importing the design.

Example

dbDelFromHInst \$instPtr \$hinstPtr

The specified instance is removed from the specified module.

dbDelFromSelSet

dbDelFromSelSet *headPtr ptr*

Deletes the specified object from a selected set.

Parameters

<i>headPtr</i>	Address of head.
<i>ptr</i>	Address of the object.

Command Order

Use this command after importing the design and selecting a set of objects.

Example

After you select a set of objects, type the following commands:

```
innovus 2> set s [dbHeadSelList]
```

```
0x82640c8
```

```
innovus 3> set objPtr [dbSelPtr $s]
```

```
0x68164b8
```

Next, add other instances to the selected set:

```
innovus 4> dbDelFromSelSet [dbgHead] $objPtr
```

Then do a redraw.

dbDelHiliteSet

```
dbDelHiliteSet headPtr
```

Deletes the highlighted set.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after importing the design and highlighting at least one object.

Example

```
dbDelHiliteSet [dbgHead]
```

The highlighted set is deleted.

dbDelHInst

```
dbDelHInst hinstPtr
```

Note: This command currently does not work, but should remove the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of the hierarchical instance.
-----------------	---------------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 112> set obj_type [dbObjType $objPtr]  
dbcObjHInst  
innovus 111> dbDelHInst $objPtr  
Error in: dbDelHInst - cannot delete "hierarchical type" hinst
```

This command always returns an Error.

dbDelProp

`dbDelProp propPtr`

Deletes the property.

Parameters

<code>propPtr</code>	Address of property.
----------------------	----------------------

Command Order

Use this command after creating a property.

Example

`dbDelProp $objPtr`

The specified property is deleted.

dbDelSelSet

`dbDelSelSet headPtr`

Removes all currently selected instances from the selected set.

Parameters

headPtr	Address of the head.
---------	----------------------

Command Order

Use this command after selecting instances.

Example

```
dbDelSelSet [dbgHead]
```

The selected set is deleted.

dbDelWireSeg

```
dbDelWireSeg x y z x y z
```

Deletes a wire segment.

Parameters

x	The x ordinate.
y	The y ordinate.
z	The metal layer.

Command Order

Use this command after routing the design (or after creating wires).

Example

```
dbDelWireSeg 300.0 1200.0 2 300.0 1400.0 2
```

Removes the *metal2* wire segment.

dbDisplayUnplacedInst

dbDisplayUnplacedInst *instPtr dbLocX dbLocY orient*

Displays the specified unplaced instance.

Parameters

<i>instPtr</i>	Address of instance.
<i>dbLocX</i>	X ordinate in DBU
<i>dbLocY</i>	Y ordinate in DBU
<i>orient</i>	Orientation of instance.

Command Order

Use this command after importing the design.

Example

```
dbDisplayUnplacedInst [dbGetInstByName DTMF_INST/i_10048] 50 50 R0
```

The unplaced instance is displayed.

E

dbEndIterAllRoutes

dbEndIterAllRoutes *iter*

Denotes the end of iterating all routes.

Parameters

<i>iter</i>	Iteration.
-------------	------------

Command Order

Use this command after importing the design.

Note: The command should always be used (as a pair) with the [dbIterAllRoutes](#) command to ensure proper functioning with other applications.

Example

```
proc iter_routes_delete_vias {} {  
  
    set fout [open "IterRoutesRmVias.out" w]  
  
    puts $fout "Routes Info:  
    puts $fout ""  
    puts $fout ""  
  
    dbForAllCellNet [dbgTopCell] net {  
  
        set nName [dbNetName $net]  
  
        dbIterAllRoutes  
  
        set iter [dbIterRoutes $net]  
  
        while {[set route [dbRouteNext $iter]] != {0x0}} {  
  
            puts $fout "Processing net -> $nName\n"  
            if { [dbIsRouteWire $route] } {  
  
                puts $fout " wirePtr -> $route"  
  
                set wire_info [dbInfoWire $route]  
  
                puts $fout " wire -> $wire_info"  
                if { [ dbIsObjStripBox $route ] } {  
  
                    set rc [ dbDeleteStripBox $route ]  
                }  
            }  
        }  
    }  
}
```

```
puts $fout " Deleted the (strip box) wire, $route, rc = $rc"
} else {
set rc [ dbDeleteRouteBox $route ]
puts $fout " Deleted the (route box) wire, $route, rc = $rc"
}
} elseif { [dbIsRouteVia $route] } {
puts $fout " viaPtr -> $route"
set via_info [dbInfoVia $route]
puts $fout " via -> $via_info"
if { [ dbIsObjStripBox $route ] } {
set rc [ dbDeleteStripBox $route ]
puts $fout " Deleted the (strip box) via, $route, rc = $rc"
} else {
set rc [ dbDeleteRouteBox $route ]
puts $fout " Deleted the (route box) via, $route, rc = $rc"
}
} else {
puts $fout "ROUTE NOT A WIRE OR A VIA"
}
puts $fout "\n"
}
dbEndIterRoutes $iter
dbEndIterAllRoutes
}
close $fout
return
}
```

Deletes all power/ground structures.

dbEndIterRoutes

dbEndIterRoutes *iter*

Denotes the end of route iteration.

Parameters

<i>iter</i>	Iteration
-------------	-----------

Command Order

Use this command after importing the design.

Example

```
set iter [dbIterRoutes $net]
while {[set route [dbRouteNext $iter]] != "0x0"} {
    if [dbIsRouteWire $route] {
        set wire_info [dbInfoWire $route]
    } elseif [dbIsRouteVia $route] {
        set via_info [dbInfoVia $route]
    } else {
        puts "something's wrong\n"
    }
    ...
    if (want to break out of the iterator loop)
        dbEndIterRoutes $iter
```

```
...  
}
```

dbEndIterViaCellRectangles

```
dbEndIterViaCellRectangles iter
```

Denotes the end of iterating via cell rectangles.

Parameters

<i>iter</i>	Iteration
-------------	-----------

Command Order

Use this command after importing the design.

Example

```
set iter [dbIterViaCellRectangles LAYER $vc]  
while {[set rect [dbViaCellRectangleNext $iter]] != ""} {  
    # $rect is list of four coordinates: lx ly hx hy  
    ...  
    if (want to break out of the iterator loop)  
        dbEndIterViaCellRectangles $iter  
    ...  
}
```

dbEvalAddEcoBuffer

`dbEvalAddEcoBuffer netPtr cellPtr xcoord ycoor tclScript`

Evaluates the cell or cell footprint at the specified coordinate.

Parameters

<i>netPtr</i>	Address of net
<i>cellPtr</i>	Address of cell
<i>xcoord</i>	x ordinate
<i>ycoor</i>	y ordinate
<i>tclScript</i>	tcl script

Command Order

Use this command after performing timing analysis.

dbEvalChangeCell

`dbEvalChangeCell instPtr cellPyt tclScript`

Evaluates changing the specified instance with the specified cell or cell footprint.

Parameters

<i>instPtr</i>	Address of instance
<i>cellPtr</i>	Address of cell
<i>tclScript</i>	tcl script

Command Order

Use this command after performing timing analysis.

dbEvalRemoveBuffer

dbEvalRemoveBuffer *instPtr tclScript*

Evaluates removing the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
<i>tclScript</i>	tcl script.

Command Order

Use this command after performing timing analysis.

dbEvalSwitchTerm

`dbEvalSwitchTerm termPtr termPtr tclScript`

Evaluates switching the first term with the second term.

Parameters

<i>termPtr</i>	Address of term.
<i>termPtr</i>	Address of term.
<i>tclScript</i>	tcl script

Command Order

Use this command after performing timing analysis.

dbExtRuleArrNext

`dbExtRuleArrNext extRulePtr`

Returns the next address of the rule specified by the current address.

Parameters

<i>extRulePtr</i>	Address of the external rule.
-------------------	-------------------------------

Command Order

Use this command after importing the design.

Example

```
dbExtRuleArrNext [dbGetExtRuleByName RULE1]
```

Returns:

0x4a7f318

The address of the external rule array is 0x4a7f318.

dbExtRuleId

```
dbExtRuleId extRulePtr
```

Returns the ID of layer rules.

Parameters

<i>extRulePtr</i>	Address of the external rule.
-------------------	-------------------------------

Command Order

Use this command after importing the design.

Example

```
dbExtRuleId [dbGetExtRuleByName RULE1]
```

Returns:

0x4a7f318

The address of the external rule ID is 0x4a7f318.

dbExtRuleLayerRuleList

dbExtRuleLayerRuleList *extRulePtr*

Returns the address of the list of layer rules.

Parameters

<i>extRulePtr</i>	Address of the external rule.
-------------------	-------------------------------

Command Order

Use this command after importing the design.

Example

```
dbExtRuleLayerRuleList [dbGetExtRuleByName RULE1]
```

Returns:

0x4a7f318

The address of the external layer rule list is 0x4a7f318.

dbExtRuleName

dbExtRuleName *extRulePtr*

Returns the name.

Parameters

<i>extRulePtr</i>	Address of the external rule.
-------------------	-------------------------------

Command Order

Use this command after importing the design.

Example

```
dbExtRuleName [dbGetExtRuleByName RULE1]
```

Returns:

RULE1

The external rule name is RULE1.

dbExtRuleNext

```
dbExtRuleNext extRulePtr
```

Returns the address of the next rule, 0x0 if there is none.

Parameters

<i>extRulePtr</i>	Address of the external rule.
-------------------	-------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 9> dbExtRuleNext [dbGetExtRuleByName RULE1]
```

0x0

There is no next external rule.

dbExtRuleNrRoutingLayerRule

dbExtRuleNrRoutingLayerRule *headPtr*

Returns the number of routing layer rules.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
innovus 7> dbExtRuleNrRoutingLayerRule [dbgHead]  
7
```

The number of routing layers is 7.

dbExtRuleRoutingLayerRuleArr

dbExtRuleRoutingLayerRuleArr *extRulePtr*

Returns the address of the array of layer rules.

Parameters

<i>extRulePtr</i>	Address of the external rule.
-------------------	-------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 10> dbExtRuleRoutingLayerRuleArr [dbGetExtRuleByName RULE1]  
0x4989890
```

The address of the external rule routing layer rule array is 0x4989890.

dbExtRuleSpacingList

```
dbExtRuleSpacingList extRulePtr
```

Returns the address off the list of spacing rules.

Parameters

<i>extRulePtr</i>	Address of the external rule.
-------------------	-------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 11> dbExtRuleSpacingList [dbGetExtRuleByName RULE1]  
0x4a87c38
```

The address of the external rule spacing list is 0x4a87c38.

dbExtRuleViaCellList

dbExtRuleViaCellList *extRulePtr*

Returns the address of the external rule via cell list.

Parameters

<i>extRulePtr</i>	Address of the external rule.
-------------------	-------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 13> dbExtRuleViaCellList [dbGetExtRuleByName RULE1]  
0x4a86c60
```

The address of the external rule via cell list is 0x4a86c60.

Database Commands F - H

- [F](#)
- [G](#)
- [H](#)

F

dbFindInstsByCell

`dbFindInstsByCell pattern`

Returns the cells that match the specified pattern.

Parameters

<i>pattern</i>	Wildcard matching
----------------	-------------------

Command Order

Use this command after importing the design.

Example

`dbFindInstsByCell YFD1`

Returns:

IADR0 IADR1 IADR2 IADR3 IADR4 IADR5 IADR6 IADR7 INCLK

IADR0-7 and INCLK are the cells that match the master cell YFD1.

dbFindInstsByName

dbFindInstsByName *pattern*

Returns the instances that match the specified pattern.

Parameters

<i>pattern</i>	Wildcard matching
----------------	-------------------

Command Order

Use this command after importing the design.

Example

dbFindInstsByName I*

Returns:

IADR0 IADR1 IADR2 IADR3 IADR4 IADR5 IADR6 IADR7 INCLK

IADR0-7 and INCLK are the instances that match.

dbFindInstsOnNet

dbFindInstsOnNet *pattern*

Returns the instances that match the specified pattern for a net.

Parameters

<i>pattern</i>	Wildcard matching
----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbFindInstsOnNet DTMF_INST/n_*
```

Returns:

```
DTMF_INST/DIGIT_REG_INST/i_9464 DTMF_INST/TDSP_DS_CS_INST/i_6352
DTMF_INST/DMA_INST/i_433 DTMF_INST/ARB_INST/i_320 DTMF_INST/SPI_INST/i_279
DTMF_INST/SPI_INST/i_4463 DTMF_INST/SPI_INST/i_4458 DTMF_INST/i_10048
DTMF_INST/ARB_INST/present_state_reg_0 DTMF_INST/SPI_INST/i_10639
DTMF_INST/DMA_INST/a_reg_0 DTMF_INST/ARB_INST/tdsp_grant_reg DTMF_INST/ARB_INST/i_10029
DTMF_INST/DMA_INST/i_10633 DTMF_INST/TDSP_DS_CS_INST/t_sel_7_reg
DTMF_INST/TDSP_DS_CS_INST/t_bit_7_reg DTMF_INST/DIGIT_REG_INST/digit_out_reg_0
```

There are many instances on the nets that begin with `DTMF_INDT/n_`.

dbFindModulesByName

```
dbFindModulesByName pattern
```

Returns the modules that match the specified pattern.

Parameters

<i>pattern</i>	Wildcard matching
----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbFindModulesByName DTMF_INST*
```

Returns:

```
DTMF_INST
```

There is one module that matches the pattern `DTMF_INST*`.

dbFindNetsByName

```
dbFindNetsByName pattern
```

Returns the nets that match the specified pattern.

Parameters

<i>pattern</i>	Wildcard matching
----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbFindNetsByName I*
```

Returns:

```
IADR0 IADR1 IADR2 IADR3 IADR4 IADR5 IADR6 IADR7 INCLK
```

`IADR0-7` and `INCLK` are the nets that match.

dbFindTermsByName

`dbFindTermsByName pattern`

Returns the instances that match the specified pattern.

Parameters

<code><i>pattern</i></code>	Wildcard matching
-----------------------------	-------------------

Command Order

Use this command after importing the design.

Example

`dbFindTermsByName I*`

Returns:

`IADR0 IADR1 IADR2 IADR3 IADR4 IADR5 IADR6 IADR7 INCLK`

`IADR0-7` and `INCLK` are the instances that match.

dbFlattenInst

`dbFlattenInst instPtr`

Flattens the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbFlattenInst MEM_DSCAN
```

Module `MEM_DSCAN` is flattened.

dbFootPrintCellList

```
dbFootPrintCellList footprint_name
```

Returns the list of cell masters that match the given footprint name.

Parameters

<i>footprint_name</i>	Footprint name.
-----------------------	-----------------

Command Order

Use this command after importing the design.

Example

```
dbFootPrintCellList a16lvt16_syncdff0_a2
```

Returns:

```
0x2aac80aa140 0x2aac805f188 0x2aac805eb40 0x2aac805f7d0 0x2aac80aff78
```

All the footprints that match `a16lvt16_syncdff0_a2` are returned.

dbFootPrintCellNameList

dbFootPrintCellNameList *footprint_name*

Returns the list of cell masters that match the given footprint name.

Parameters

<i>footprint_name</i>	Footprint name.
-----------------------	-----------------

Command Order

Use this command after importing the design.

Example

dbFootPrintCellNameList IV

Returns:

IV4 IV IVP_SPG IVP IV6 IV7 IV8 IV10 IV12

All the names of the footprints that match IV are returned.

dbFootPrintTIsCellList

dbFootPrintTIsCellList *footprint_name*

Returns a list of all timing library cells in the specified footprint.

Parameters

<i>footprint_name</i>	Footprint name.
-----------------------	-----------------

Command Order

Use this command after importing the design.

dbForAllCellFTerm

dbForAllCellFTerm *cellPtr ftermPtrVar body*

Returns a list of f-term pointers (*ftermPtrVar*).

Parameters

<i>cellPtr</i>	Address of cell.
<i>ftermPtrVar</i>	Variable name.
<i>body</i>	.

Command Order

Use this command after importing the design.

Example

```
set cellPtr [dbInstCell $objPtr]
```

Returns:

0x49e17b8

```
dbForAllCellFTerm $cellPtr ftermPtr {
```

```
set dir [dbFTermInOutDir $ftermPtr]  
puts "dir = $dir"  
}
```

Returns:

```
dir = dbcOutput  
dir = dbcOutput  
dir = dbcOutput
```

All of the f-terms are outputs.

dbForAllCellFTermandInternalFTerm

dbForAllCellFTermandInternalFTerm cellPtr ftermPtrVar body

Returns a list of all the specified cells f-term pointers (*ftermPtrVar*).

Parameters

<i>cellPtr</i>	Address of cell.
<i>ftermPtrVar</i>	Variable name.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
set cellPtr [dbInstCell $objPtr]
```

Returns:

0x49e17b8

```
dbForAllCellFTermAndInternalFTerm $cellPtr ftermPtr {  
    set dir [dbFTermInOutDir $ftermPtr]  
    puts "dir = $dir"  
}
```

Returns:

```
dir = dbcOutput  
dir = dbcOutput  
dir = dbcInput  
dir = dbcInput
```

Two of the f-terms are outputs and two are inputs.

dbForAllCellNet

dbForAllCellNet cellPtr netPtrVar body

Iterates through all the nets for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>netPtrVar</i>	Variable containing address of nets.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForAllCellNet [dbgTopCell] netPtr {  
    set nName [dbNetName $netPtr]  
    puts "net name = $nName"  
}
```

Returns:

```
net name = BLU  
net name = CLA
```

The nets BLU and CLA are all of the nets in the top cell.

dbForAllCellPGFTerm

```
dbForAllCellPGFTerm cellPtr ftermPtrVar body
```

Returns all PG f-terms.

Parameters

<i>cellPtr</i>	Address of cell.
<i>ftermPtrVar</i>	f-term address variable.
<i>body</i>	

Command Order

Use this command after importing the design and P/G have been defined.

Example

```
dbForAllCellPGFTerm $cellPtr ftermPtr {
```

```
set name [dbFTermName $ftermPtr]  
puts "* pgfterm name: $name"  
printOneFTerm $ftermPtr  
}
```

Returns:

* pgfterm name: VDD
* pgfterm name: VSS

VDD and VSS are all of the PG f-terms for the specified cell.

dbForAllCellPGTerm

dbForAllCellPGTerm cellPtr termPtrVar body

Returns all P/G terminals.

Parameters

<i>cellPtr</i>	Address of cell.
<i>termPtrVar</i>	Address of terminals
<i>body</i>	

Command Order

Use this command after importing the design and P/G have been defined.

Example

```
dbForAllCellPGTerm $cellPtr termPtr {  
    set name [dbTermName $termPtr]
```

```
puts "* pgterm name: $name"  
}
```

Returns:

```
* pgfterm name: VDD  
* pgfterm name: VSS
```

VDD and VSS are all of the PG f-terms for all cells.

dbForAllInstTerm

```
dbForAllInstTerm cellPtr termPtrVar body
```

Returns the address of all terminals for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>termPtrVar</i>	terminal addresses.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x965eaf8
```

```
set cellPtr [dbSelPtr $s]
```

Returns:

```
0x902f63c
```

```
dbForAllInstTerm $cellPtr termPtr {  
  
    set name [dbTermName $termPtr]  
  
    puts "* term name: $name"  
}
```

Returns:

```
* term name: DBLU0  
* term name: DBLU1  
* term name: DBLU2  
* term name: DBLU3  
* term name: DBLU4
```

The terms for the selected cell are DBLU0-4.

dbForAllVInstVTerm

```
dbForAllVInstVTerm vinstPtr vtermPtrVar body
```

Returns the addresses for all Verilog terminals.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
<i>vtermPtrVar</i>	Address of terminals.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForEachCellVIInst $objPtr vinstPtr {  
    dbForAllVIInstVTerm $vinstPtr vtermPtr {  
        set fterm [dbVTermFTerm $vterPtr]  
        set name [dbFTermName $fterm]  
        puts "name = $name"  
    }  
}
```

Returns all of the f-terms for the specified Verilog instance.

dbForEachAreaECOScreen

```
dbForEachAreaECOScreen Ptr screenPtrVar body
```

Returns the addresses of all ECO screens.

Parameters

<i>Ptr</i>	Address of ECO screen.
<i>screenPtrVar</i>	Addresses of ECO screens.
<i>body</i>	

Command Order

Use this command after importing the design and creating at least one ECO screen.

Example

```
dbCreateAreaECOScreen 3000 3000 4000 4000 .5
```

Returns:

```
0xa209548
```

```
set Ptr [dbCreateAreaECOScreen 3000 3000 4000 4000 .5]
```

Returns:

```
0xa209564
```

```
dbForEachAreaECOScreen $Ptr screenPtr {  
    puts "Screen address = $screenPtr" }
```

Returns:

```
Screen address = 0xa209548
```

```
Screen address = 0xa209564
```

Returns the addresses of two area ECO screens.

dbForEachCellBlackBox

```
dbForEachCellBlackBox cellPtr blackBoxPtrVar body
```

Returns the addresses of blackboxes for the associated cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>blackBoxPtrVar</i>	blackbox variable.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForEachCellBlackBox $cellPtr blackBoxPtr {  
    set name [dbBlackBoxName $blackBoxPtr]  
    puts " * blackbox name: $name"  
}
```

The names of all of the blackboxes for the specified cell are returned.

dbForEachCellBump

```
dbForEachCellBump cellPtr bumpPtrVar body
```

Returns the addresses of the bumps for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>bumpPtrVar</i>	bump variable.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForEachCellBump $cellPtr bumpPtr {  
    set name [dbBumpName $bumpPtr]  
    puts " * bumpname: $name"  
}
```

The bump names are returned for the specified cell.

dbForEachCellFTerm

dbForEachCellFTerm cellPtr ftermPtrVar body

Returns the addresses of f-terms for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>ftermPtrVar</i>	f-term variable.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForEachCellFTerm $cellPtr ftermPtr {  
    set name [dbFTermName $fermPtr]  
    puts " * fterm name: $name"  
}
```

Returns all of the f-term names for the specified cell.

dbForEachCellGeomList

```
dbForEachCellGeomList cellPtr geomListPtrVar body
```

Returns the address of geometries associated with the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>geomListPtrVar</i>	Addresses of geometries.
<i>body</i>	

Command Order

Use this command after importing the design.

dbForEachCellHaloBox

```
dbForEachCellHaloBox cellPtr boxPtrVar body
```

Returns the box of the halo for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>boxPtrVar</i>	Addresses of halo boxes.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForEachCellHaloBox $cellPtr boxPtr {  
    puts $boxPtr  
}
```

Returns:

9849600 0 10456100 965000

The halo is a box with the lower left coordinate at (9849.6, 0) and the upper right coordinate at (10456.1, 965.0).

dbForEachCellInputFTerm

```
dbForEachCellInputFTerm cellPtr ftermPtrVar body
```

Returns all input f-terms for the specified cell.

Parameters

<i>cellPtr</i>	Address of specified cell.
<i>ftermPtrVar</i>	f-term variable.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
set cellPtr [dbgTopCell]
```

Returns:

```
0x4e378e0
```

```
dbForEachCellInputFTerm $cellPtr ftermPtr {  
    set name [dbFTermName $ftermPtr]  
    puts "* term name: $name"  
}
```

Returns:

```
* term name: A2  
* term name: A3  
* term name: A4
```

The input f-terms for the specified cell are A2, A3 and A4.

dbForEachCellInst

`dbForEachCellInst cellPtr instPtrVar body`

Returns the instance specified by cell.

Parameters

<code>cellPtr</code>	Address of cell.
<code>instPtrVar</code>	Instance variable.
<code>body</code>	.

Command Order

Use this command after importing the design.

Example

```
proc findInstsByCellName { cellName } {  
  
    set designName [dbgDesignName]  
  
    Puts "\nInfo: Searching for number of instances with $cellName cell master in  
    $designName...\n"  
  
    set cnt 0  
  
    set cellPtr [dbgTopCell]  
  
    dbForEachCellInst $cellPtr instPtr {  
  
        set instName [dbInstName $instPtr]  
  
        #puts " - Instance: $instName"  
  
        set leafCell [dbInstCellName $instPtr]  
  
        if { $leafCell == $cellName } {  
  
            incr cnt  
  
            Puts " $instName"  
  
        }  
    }  
}
```

```
}
```

```
Puts "\nInfo: Found $cnt "
```

Reports all instances with the specified cell master.

dbForEachCellIo

```
dbForEachCellIo cellPtr ioPtrVar body
```

Returns the address of IOs for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>ioPtrVar</i>	Io variable.
<i>body</i>	.

Command Order

Use this command after importing the design.

Example

```
proc findIosByCellName { cellName } {  
  
    set designName [dbgDesignName]  
  
    Puts "\nInfo: Searching for number of instances with $cellName cell master in  
    $designName...\n"  
  
    set cnt 0  
  
    set cellPtr [dbgTopCell]
```

```
dbForEachCellIo $cellPtr ioPtr {  
  
set instName [dbInstName $ioPtr]  
  
#puts " - Io: $instName"  
  
set leafCell [dbInstCellName $instPtr]  
  
if { $leafCell == $cellName } {  

```

Report all IOs with the specified cell master.

dbForEachCellNet

dbForEachCellNet *cellPtr netPtrVar body*

Returns all the special nets for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>netPtrVar</i>	Net variable.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
proc findNets { netName } {  
  
    dbForEachCellNet [dbgTopCell] netPtr {  
  
        set net [dbNetName $netPtr]  
  
        if { $net == $netName } { puts $net}  
  
    }  
  
}
```

Report the special net.

dbForEachCellOutputFTerm

dbForEachCellOutputFTerm cellPtr ftermPtrVar body

Returns all of the active output f-terms for the specified cell. The command does not return the names of unconnected output f-terms.

Parameters

<i>cellPtr</i>	Address of cell.
<i>ftermPtrVar</i>	f-term variable.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForEachCellOutputFTerm $cellPtr ftermPtr {  
    set name [dbFTermName $ftermPtr]  
    puts " * term name: $name"  
}
```

```
* term name: VLF  
* term name: WRPP  
* term name: XTL2
```

The output f-terms for the specified cell are VLF, VRPP and XTL2.

dbForEachCellPhyFTerm

```
dbForEachCellPhyFTerm cellPtr ftermPtrVar body
```

Returns all of the addresses of the physical f-terms for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>ftermPtrVar</i>	f-term variable.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachCellPhyFTerm $cellPtr ftermPtr {  
    set name [dbFTermName $ftermPtr]  
    puts " * term name: $name"  
}
```

dbForEachCellPtn

dbForEachCellPtn cellPtr ptnPtrVar body

Returns all of the addresses of the partitions for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>ptnPtrVar</i>	Partition variable.
<i>body</i>	

Command Order

Use this command after design import and partitioning.

Example

```
dbForEachCellPtn $cellPtr ptnPtr {  
    set name [dbPtnName $ptnPtr]  
    puts " * partition name: $name"  
}  
* partition name: sheet17
```

```
* partition name: sheet25
* partition name: sheet7
```

The partitions for the specified cell are sheet17, sheet25, and sheet7.

dbForEachCellSNet

`dbForEachCellSNet cellPtr netPtrVar body`

Returns all of the addresses of the special nets for the specified cell.

Parameters

<code>cellPtr</code>	Address of cell.
<code>netPtrVar</code>	Addresses of special nets.
<code>body</code>	

Command Order

Use this command after design import.

Example

```
dbForEachCellSNet [dbgTopCell] netPtr {
    set name [dbSNetName $netPtr]
    puts "name = $name"
}
```

Reports all special nets for the top cell.

dbForEachCellSortedInst

```
dbForEachCellSortedInst {-byName | -bySize} cellPtr instPtrVar body
```

Returns a sorted list of instances for the specified cell.

Parameters

-byName	Option to sort based on alphabet.
-bySize	Option to sort based on size.
cellPtr	Address of cell.
instPtrVar	Instance variable.
body	

Example

```
dbForEachCellSortedInst $cellPtr instPtr {  
    set name [dbInstName $instPtr]  
    puts " * inst name: $name"  
}
```

```
* inst name: SH28/I12/I36/PACK1/YUV_reg<0>  
* inst name: SH28/I12/I36/PACK1/YUV_reg<1>  
* inst name: SH28/I12/I36/PACK1/YUV_reg<2>  
* inst name: SH28/I12/I36/PACK1/YUV_reg<3>
```

Four instances are sorted

dbForEachCellSortedVInst

`dbForEachCellSortedVInst {-byName | -bySize} cellPtr instPtrVar body`

Returns a sorted list of instances for the specified cell.

Parameters

<code>-byName</code>	Specifies to sort by name.
<code>-bySize</code>	Specifies to sort by size.
<code>cellPtr</code>	Address of cell.
<code>instPtrVar</code>	Address of instances.
<code>body</code>	

Command Order

Use this command after design import.

dbForEachCellStorInst

`dbForEachCellStorInst cellPtr instPtrVar body`

Returns all of the "storage" instances in the specified cell.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

<i>instPtrVar</i>	Instance variable.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachCellStorInst $cellPtr instPtr {  
  
set name [dbInstName $instPtr]  
  
puts " * inst name: $name"  
}
```

Returns all storage instances for the specified cell.

dbForEachCellTileInst

```
dbForEachCellTileInst cellPtr instPtrVar body
```

Returns the address of tile instances for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>instPtrVar</i>	Addresses of tile instances.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachCellTileInst [dbgTopCell] tilePtr {  
    set objname [dbObjName $tilePtr]  
    puts "object name = $objname"  
}
```

Reports the address of the tile instances.

dbForEachCellTimeArc

dbForEachCellTimeArc cellPtr timeArcPtrVar body

Returns the type of timing arc in the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>timeArcPtrVar</i>	Addresses of time arcs.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachCellTimeArc [dbgTopCell] timeArcPtr {
```

```
set type [dbTimeArcType $timeArcPtr]  
puts "type = $type"  
}
```

Reports the type of timing arc for all arcs in the top cell.

dbForEachCellVInst

```
dbForEachCellVInst cellPtr vinstPtrVar body
```

Returns the name of Verilog instances for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>vinstPtrVar</i>	Addresses of Verilog instances.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachCellVInst [dbgTopCell] vinstPtr {  
    set name [dbInstName $vinstPtr]  
    puts "name = $name"  
    set sel [dbIsVInstHilite $vinstPtr]  
    puts "sel = $sel"  
}
```

Returns the name of all Verilog instances in the top cell and reports whether the instance is highlighted.

dbForEachClockDomainInst

dbForEachClockDomainInst

Outputs a report on the instances for the selected clock domain.

Command Order

Use this command after importing the design, specifying clock domains, and selecting a clock domain.

dbForEachClockDomainNet

dbForEachClockDomainNet

Outputs a report on the nets for the selected clock domain.

Command Order

Use this command after importing the design, specifying clock domains, and then selecting a clock domain.

dbForEachConstraintBox

dbForEachConstraintBox *conPtr boxPtrVar body*

Returns all of the box addresses for the specified constraint.

Parameters

<i>conPtr</i>	Address of constraint
<i>boxPtrVar</i>	Address of box.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanConstraint [dbHeadFPlan] constraint {  
    dbForEachConstraintBox $constraint box {  
        puts "[dbBoxLLX [dbBoxPtrBox $box]]"  
    }  
}
```

dbForEachCutLayer

dbForEachCutLayer layerPtrVar body

Returns the addresses of the specified cut layer blockage.

Parameters

<i>layerPtrVar</i>	Addresses of layers.
<i>body</i>	

Command Order

Use this command after importing the design.

dbForEachDomainInst

dbForEachDomainInst

Outputs a report on the instances for all domains.

Command Order

Use this command after importing the design.

dbForEachExtRuleLayerRuleList

dbForEachExtRuleLayerRuleList *extRulePtr* *layerRulePtrVar* *body*

Returns the layer rule name for the specified external rule.

Parameters

<i>extRulePtr</i>	Address of external rule.
<i>layerRulePtrVar</i>	Addresses of layer rules.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachExtRuleLayerRuleList $extRulePtr ptr {  
    set name [ set name [dbLayerRuleName $ptr]  
    puts " * layer rule name: $name"  
}
```

Returns the layer rule name for all external rules.

dbForEachFileLine

```
dbForEachFileLine fileName line body
```

Returns the line for the specified file.

Parameters

<i>filename</i>	Name of file.
<i>line</i>	Line in file.
<i>body</i>	

Command Order

Use this command after starting the Innovus Implementation System (Innovus) software.

Example

```
dbForEachFileLine runme line  
puts $line  
}
```

dbForEachFileUpdateLine

`dbForEachFileUpdateLine fileName line body`

Returns the updated line for the specified file.

Parameters

<i>filename</i>	Name of file.
<i>line</i>	Line in file
<i>body</i>	

Command Order

Use this command after starting the Innovus software.

Example

```
dbForEachFileUpdateLine runme line
puts $line
}
```

dbForEachFPlanAIORowCluster

`dbForEachFPlanAIORowCluster fplanPtr rowClusterPtrVar body`

Returns the addresses of row clusters for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floor plan.
<i>rowClusterPtrVar</i>	Addresses of row clusters.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachFPlanAIORowCluster [dbHeadFPlan] rowClusterPtr {  
    dbForEachRowClusterRow $rowClusterPtr stdRowPtr {  
        set box [dbRowClusterBox $stdRowPtr]  
        puts "box = $box"  
    }  
}
```

Returns the box for each area I/O row.

dbForEachFPlanCellPad

```
dbForEachFPlanCellPad fplanPtr cellPadPtrVar body
```

Returns the instance name for the specified I/O pads.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>cellPadPtrVar</i>	Addresses of I/Os.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachFPlanCellPad [dbHeadFPlan] ptr {
    set name [ set name [dbInstName $ptr]
    puts " * inst name: $name"
}
```

Reports the instance name for all the pads.

dbForEachFPlanConstraint

dbForEachFPlanConstraint fplanPtr constraintPtrVar body

Returns all of the floorplan constraints.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>constraintPtrVar</i>	Constraint variable.
<i>body</i>	

Command Order

Use this command after design import and after some constraints have been applied.

Example

```
dbForEachFPlanConstraint [dbHeadFPlan] constrPtr {  
    set constraintType [dbConstraintType $constrPtr]  
    puts " * constraint type: $constraintType"  
}  
  
* constraint type: dbcFence  
* constraint type: dbcFence  
* constraint type: dbcFence
```

All of the constraints in this design are fences.

dbForEachFPlanDefRow

dbForEachFPlanDefRow fplanPtr stdRowPtrVar body

Returns the coordinates of the specified standard rows.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>stdRowPtrVar</i>	Standard row variable.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachFPlanDefRow [ dbHeadFPlan] rowPtr {  
  
set box [dbStdRowBox $rowPtr]  
  
puts " * box : $box"  
}
```

Returns:

```
* box : 430300 6748300 6802300 6775300
```

The standard row has a lower left coordinate of (430.3, 6748.3) and an upper right coordinate of (6802.3, 6775.3).

dbForEachFPlanGlobalNetConnection

```
dbForEachFPlanGlobalNetConnection fplanPtr globalNetConnectionPtrVar body
```

Returns the type of global connections associated with the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>globalNetConnectionPtrVar</i>	
	Global connections variable.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachFPlanGlobalNetConnection [dbHeadFPlan] glbnetPtr {  
    set type [dbGlobalNetConnectionConnectType $glbnetPtr]  
    puts " * type: $type"  
}
```

Returns the type of global connection for all of the global nets in the design.

dbForEachFPlanGroup

dbForEachFPlanGroup fplanPtr groupPtrVar body

Returns the groups associated with the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>groupPtrVar</i>	Group variable.
<i>body</i>	

Command Order

Use this command after design import and after groups have been created.

Example

```
dbForEachFPlanGroup [dbHeadFPlan] groupPtr {  
    set groupName [dbGroupName $groupPtr]  
    puts " * group name: $groupName"  
}
```

Returns all of the group names for the design.

dbForEachFPlanLayerBlk

```
dbForEachFPlanLayerBlk fplanPtr layerBlkPtrVar body
```

Returns routing blockages.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>layerBlkPtrVar</i>	Routing blockage variable.
<i>body</i>	

Command Order

Use this command after design import and after routing blockages have been created.

Example

```
dbForEachFPlanLayerBlk [dbHeadFPlan] layerBlkPtr {  
    set lbbox [dbLayerBlkBox $layerBlkPtr]  
    puts " * layer blockage bbox: $lbbox"
```

```
}
```

* layer blockage bbox: 2789434 3343819 3275162 3770311

There is a layer blockage at (2789.434, 3343.819) (3275.162, 3770.311).

dbForEachFPlanNetGroup

dbForEachFPlanNetGroup *fplanPtr netGroupPtrVar body*

Returns the address of the net groups for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>netGroupPtrVar</i>	Net group variable.
<i>body</i>	

Command Order

Use this command after design import and after net groups have been created.

Example

```
dbForEachFPlanObstruct [dbHeadFPlan] netGroupPtr {  
  
set name [dbNetGroupName $netGroupPtr]  
  
puts " * name : $name"  
}
```

```
* name : netGroup1
* name : netGroup2
```

The names of the 2 net groups are: `netGroup1` and `netGroup2`

dbForEachFPlanObstruct

`dbForEachFPlanObstruct fplanPtr obstructPtrVar body`

Returns floorplan obstructions for the specified floorplan.

Parameters

<code>fplanPtr</code>	Address of floorplan.
<code>obstructPtrVar</code>	Obstruction variable.
<code>body</code>	

Command Order

Use this command after design import and after obstructions have been created.

Example

```
dbForEachFPlanObstruct [dbHeadFPlan] obsPtr {
    set name [dbObstructName $obsPtr]
    puts " * name : $name"
}
```

Returns:

```
* name : defObstructName1
* name : defObstructName2
```

The names of the obstructions are `defObstructName1` and `defObstructName2`.

dbForEachFPlanPinGroup

`dbForEachFPlanPinGroup fplanPtr pinGroupPtrVar body`

Returns names of the pin groups for the specified floorplan.

Parameters

<code>fplanPtr</code>	Address of floorplan.
<code>pinGroupPtrVar</code>	Pin Group variable.
<code>body</code>	

Command Order

Use this command after design import and after pin groups have been created.

Example

```
dbForEachFPlanObstruct [dbHeadFPlan] pinGroupPtr {
    set name [dbPinGroupName $obsPtr]
    puts " * name : $name"
}
```

Returns:

```
* name : pinGroup1
```

* name : pinGroup2

The names of the two pin groups are: pinGroup1 and pinGroup2.

dbForEachFPlanPtnCut

dbForEachFPlanPtnCut *fplanPtr ptnCutPtrVar body*

Returns partition cuts for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>ptnCutPtrVar</i>	Partition Cut variable.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachFPlanPtnCut [dbHeadFPlan] ptncutPtr {  
    set box [dbPtnCutBox $ptncutPtr]  
    puts * ptn bbox : $box$"  
}
```

Returns the boxes of all of the partition cuts for the design.

dbForEachFPlanPtnFeed

```
dbForEachFPlanPtnFeed fplanPtr ptnFeedPtrVar body
```

Returns partition feedthroughs for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>ptnFeedPtrVar</i>	Partition feedthrough variable.
<i>body</i>	

Command Order

Use this command after design import, partitions have been created and feedthroughs added to at least one of the partitions.

Example

```
dbForEachFPlanPtnFeed [dbHeadFPlan] feedPtr {  
    set box [dbPtnFeedBox $feedPtr]  
    puts " * ptn bbox : $box"  
}
```

Returns the partition feedthrough boxes for the design.

dbForEachFPlanPtnPinBlk

```
dbForEachFPlanPtnPinBlk fplanPtr ptnPinBlkPtrVar body
```

Returns the partition pin blockages for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>ptnPinBlkPtrVar</i>	Partition pin blockage variable.
<i>body</i>	

Command Order

Use this command after importing the design, creating partitions, and creating at least one partition pin blockage.

Example

```
dbForEachFPlanPtnPinBlk [dbHeadFPlan] ptnpinblkPtr {  
    set name [dbPtnPinBlkName $ptnpinblkPtr]  
    puts " * ptn blk name : $name"  
}
```

Returns the partition blockage names for the design.

dbForEachFPlanPtnPinBlks

```
dbForEachFPlanPtnPinBlks fplanPtr ptnPinBlkPtrVar body
```

Returns the partition pin blockages for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
-----------------	-----------------------

<i>ptnPinBlkPtrVar</i>	Partition pin blockage variable.
<i>body</i>	

Command Order

Use this command after importing the design, creating partitions, and creating at least one partition pin blockage.

Example

```
dbForEachFPlanPtnPinBlks [dbHeadFPlan] ptnpinblkPtr {
    set name [dbPtnPinBlkName $ptnpinblkPtr]
    puts " * ptn blk name : $name"
}
```

Returns the partition blockage names for the design.

dbForEachFPlanRouteBlk

```
dbForEachFPlanRouteBlk fplanPtr routeBlkPtrVar body
```

Returns routing blockages.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>routeBlkPtrVar</i>	Routing blockage variable.
<i>body</i>	

Command Order

Use this command after importing the design and after creating at least one routing blockage.

Example

```
dbForEachFPlanRouteBlk [dbHeadFPlan] layerBlkPtr {  
    set lbbox [dbLayerBlkBox $layerBlkPtr]  
    puts " * layer blockage bbox: $lbbox"  
}
```

Returns:

```
* layer blockage bbox: 2789434 3343819 3275162 3770311
```

There is a route blockage at (2789.434, 3343.819) (3275.162, 3770.311).

dbForEachFPlanRouteGuide

```
dbForEachFPlanRouteGuide fplanPtr routeGuidePtrVar body
```

Returns placement density screens for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>routeGuidePtrVar</i>	Addresses of route guides.
<i>body</i>	

Command Order

Use this command after design import and after at least one route guide has been created.

Example

```
dbForEachFPlanRouteGuide [dbHeadFPlan] routeGuidePtr {  
    set name [dbRouteGuideName $routeGuidePtr]  
    puts " * box : $box"  
}
```

Returns:

* name : dma_feeds

The name of the route guide is `dma_feeds`.

dbForEachFPlanScreen

`dbForEachFPlanScreen fplanPtr screenPtrVar body`

Returns placement density screens for the specified floorplan.

Parameters

<code>fplanPtr</code>	Address of floorplan.
<code>screenPtrVar</code>	Density screen variable.
<code>body</code>	

Command Order

Use this command after design import and at least one density screen has been created.

Example

```
dbForEachFPlanScreen [dbHeadFPlan] screenPtr {  
    set box [dbScreenBox $screenPtr]
```

```
puts " * box : $box"  
}
```

Returns:

* box : 5419300 3400300 6083300 3832300

The only screen for the design is at (5419.3,3400.3) (6083.3,3832.3).

dbForEachFPlanStrip

dbForEachFPlanStrip fplanPtr stripPtrVar body

Returns names of the strips for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>stripPtrVar</i>	Address of strips.
<i>body</i>	

Command Order

Use this command after importing the design and after creating at least one power/ground.

Example

```
dbForEachFPlanStrip [dbHeadFPlan] stripPtr {  
    set name [dbStripName $stripPtr]  
    puts "name = $name"  
}
```

Returns:

```
name = gnd
name = vdd
```

There are two floorplan strips in the design: `gnd` and `vdd`.

dbForEachFTermFPin

```
dbForEachFTermFPin ftermPtr fpinPtrVar body
```

Returns the address of FPins for the specified f-term.

Parameters

<i>ftermPtr</i>	Address of f-term.
<i>fpinPtrVar</i>	Address of FPins.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHInstHTerm $hinstptr ptr {
    set netPtr [dbHTermNet $ptr]
    dbForEachNetTerm $netPtr termPtr {
        set ftermPtr [dbTermFTerm $termPtr]
        dbForEachFTermFPin $ftermPtr fpinPtr {
            puts "Address of FPin = $fpinPtr"
        } }}}
```

Returns:

Address of FPin = 0x4968fe8

Address of FPin = 0x4a0cbe0

Address of FPin = 0x495cd08

The address of the FPins are returned.

dbForEachFTermLefPort

`dbForEachFTermLefPort ftermPtr lefPortPtrVar body`

Returns the address of the LEF ports for the specified f-term.

Parameters

<code>ftermPtr</code>	Address of f-term.
<code>lefPortPtrVar</code>	Address of LEF port.
<code>body</code>	

Command Order

Use this command after importing the design.

Example

```
proc printOneLefPort {p} {
    if {[dbIsLefPortClassNone $p]} {
        puts "lefPort class <none>"
    } elseif {[dbIsLefPortClassCore $p]} {
        puts "lefPort class <core>"
    } elseif {[dbIsLefPortClassUndefined $p]} {
```

```
puts "lefPort class <undefined>"  
} else {  
puts "*** error *** lefPort class"  
}  
}  
}
```

```
proc printOneFTerm {ft} {  
puts "fterm [dbFTermName $ft]"  
dbForEachFTermLefPort $ft p {  
printOneLefPort $p  
}  
}
```

dbForEachGeomListBox

dbForEachGeomListBox *gdbsGeomPtr* *geomBoxPtrVar* *body*

Returns the address of geometry boxes associated with the specified *gdbsGeom*.

Parameters

<i>gdbsGeomPtr</i>	Address of
<i>geomBoxPtrVar</i>	Addresses of geometry boxes
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachFPlanGeomList [dbHeadFPlan] g {  
    dbForEachGeomListBox $g geomBoxPtr {  
        puts "Address of geometry box = $geomBoxPtr"  
    } }  
}
```

Returns:

Address of geometry box = 0x11f73fe0

Address of geometry box = 0x11f73fc4

The addresses of the geometry boxes are returned.

dbForEachGeomListGeom

dbForEachGeomListGeom *gdbsGeomPtr* *geomGeomPtrVar* *body*

Returns the address of geometry boxes associated with the specified *gdbsGeom*.

Parameters

<i>gdbsGeomPtr</i>	Address of
<i>geomGeomPtrVar</i>	Addresses of geometry boxes
<i>body</i>	

Command Order

Use this command after

Example

```
dbForEachFPlanGeomList [dbHeadFPlan] g {  
    dbForEachGeomListGeom $g geomBoxPtr {  
        puts "Address of geometry = $geomBoxPtr"  
    } }  
}
```

Returns:

Address of geometry = 0x11b73fe0

Address of geometry = 0x11b73fc4

The addresses for all of the geometry boxes are returned.

dbForEachGeomListPath

dbForEachGeomListPath *gdbsGeomList* *geomPathPtr* *body*

Returns the address of geometry paths.

Parameters

<i>gdbsGeomList</i>	Address of list of geometries
<i>geomPathPtr</i>	Addresses of geometry path
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachFPlanGeomList [dbHeadFPlan] g {
```

```
dbForEachGeomListPath $g geomPathPtr {  
    puts "Address of geometry path = $geomPathPtr"  
}
```

Returns:

Address of geometry path = 0x11f73be0

Address of geometry path = 0x11f73bc4

Returns the address of the geometry paths.

dbForEachGeomListPoly

```
dbForEachGeomListPoly gdbsGeomPtr geomPolyPtrVar body
```

Returns the address of geometry boxes associated with the specified.

Parameters

<i>gdbsGeomPtr</i>	Address of <i>gdbsGeom</i> .
<i>geomPolyPtrVar</i>	Addresses of geometry boxes
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachFPlanGeomList [dbHeadFPlan] g {  
    dbForEachGeomListPoly $g geomBoxPtr {  
        puts "Address of geometry poly = $geomBoxPtr"
```

} }

Returns:

Address of geometry poly = 0x11f73be0

Address of geometry poly = 0x11f73bc4

Returns the address of all of the poly geometries.

dbForEachGroupBoxListBox

dbForEachGroupBoxListBox groupPtr geomListPtrVar body

Returns the address of geometries associated with the specified cell.

Parameters

<i>groupPtr</i>	Address of group.
<i>geomListPtrVar</i>	Addresses of geometries.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
createInstGroup junk
addInstToInstGroup junk DTMF_INST/RAM_128x16_TEST_INST/RAM_128x16_INST
dbForEachGroupBoxListBox [dbGetGroupByName junk] p {puts $p}
```

dbForEachGroupHInst

`dbForEachGroupHInst groupPtr hinstPtrVar body`

Returns the names of all hierarchical instances for the specified group.

Parameters

<i>groupPtr</i>	Address of
<i>hinstPtrVar</i>	Specifies
<i>body</i>	Specifies

Command Order

Use this command after

Example

```
dbForEachGroupHInst [dbGetGroupByName my_group1] hinstPtr {  
    set name [dbHInstName $hinstPtr]  
    puts "name = $name"}
```

Returns:

```
name = SH15/I384  
name = SH15/I383
```

There are two instances for the group `my_group1`.

dbForEachHeadBumpCell

dbForEachHeadBumpCell *headPtr* *bumpCellPtrVar* *body*

Returns names of all of the bump cells for the design.

Parameters

<i>headPtr</i>	Address of Head.
<i>bumpCellPtrVar</i>	Address of bump cells.
<i>body</i>	

Command Order

Use this command after importing the design (including bump cells).

Example

```
dbForEachHeadBumpCell [dbgHead] bumpCellPtr {  
    set name [dbBumpCellName $bumpCellPtr]  
    puts "name = $name"  
}
```

Reports the name of each bump cell in the design.

dbForEachHeadCell

dbForEachHeadCell *headPtr* *cellPtrVar* *body*

Returns the object type for each of the specified cells in the design.

Parameters

<i>headPtr</i>	Address of Head.
<i>cellPtrVar</i>	Address of cells.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadCell [dbgHead] dpinstPtr {  
    set obj_type [dbObjType $dpinstPtr]  
    puts "type = $obj_type"  
}
```

Returns:

```
type = dbcObjCell  
type = dbcObjCell  
type = dbcObjCell
```

There are 3 cells of type `dbcObjCell` in the design.

dbForEachHeadClock

```
dbForEachHeadClock headPtr clockPtrVar body
```

Returns the type for each specified clock in the design.

Parameters

<i>headPtr</i>	Addresses of Head.
<i>clockPtrVar</i>	Addresses of clock.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadClock [dbgHead] dpinstPtr {  
    set obj_type [dbObjType $dpinstPtr]  
    puts "type = $obj_type"  
}
```

Returns the type of clock for each clock in the design.

dbForEachHeadCustomLayer

```
dbForEachHeadCustomLayer headPtr customLayerPtrVar body
```

Returns the names of the custom layers in the design.

Parameters

<i>headPtr</i>	Address of Head.
----------------	------------------

<i>customLayerPtrVar</i>	Addresses of custom layers.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadCustomLayer [dbgHead] clPtr {  
    set cl_name [dbCustomLayerName $clPtr]  
    puts "name = $cl_name"  
}
```

Returns:

```
name = VDDMet3
```

There is one custom layer, `VDDMet3`, in the design.

dbForEachHeadHilitePtr

```
dbForEachHeadHilitePtr headPtr ptrVar body
```

Returns the type of the highlighted objects in the design.

Parameters

<i>headPtr</i>	Address of Head.
<i>ptrVar</i>	Addresses of highlights.
<i>body</i>	

Command Order

Use this command after design import and after objects have been highlighted.

Example

```
dbForEachHeadHilitePtr [dbgHead] dpinstPtr {  
    set obj_type [dbObjType $dpinstPtr]  
    puts "type = $obj_type"  
}
```

Returns the type of object highlighted.

dbForEachHeadLayer

```
dbForEachHeadLayer headPtr layerPtrVar body
```

Returns names of layers in the design.

Parameters

<i>headPtr</i>	Address of Head.
<i>layerPtrVar</i>	Addresses of layers.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadLayer [dbgHead] layerPtr {
```

```
set name [dbLayerName $layerPtr]  
puts "name = $name"  
}
```

Returns:

```
name = METAL1  
name = METAL2  
name = METAL3  
name = VS1  
name = V12  
name = V23  
name = CUT02  
name = CUT03
```

The names of all of the layers are reported.

dbForEachHeadPath

dbForEachHeadPath headPtr pathPtrVar body

Returns the delay for specified paths in the design.

Parameters

<i>headPtr</i>	Address of Head.
<i>pathPtrVar</i>	Addresses of paths
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadPath [dbgHead] pathPtr {  
    set delay [dbPathFallDelay $pathPtr]  
    puts "delay = $delay"  
}
```

Reports the delay for every path in the design.

dbForEachHeadPtn

dbForEachHeadPtn headPtr ptnPtrVar body

Returns the names of partitions in design.

Parameters

<i>headPtr</i>	Address of Head.
<i>ptnPtrVar</i>	Address of partitions.
<i>body</i>	

Command Order

Use this command after partitioning.

Example

```
dbForEachHeadPtn [dbgHead] ptnPtr {  
  
set name [dbPtnName $ptnPtr]  
  
puts "name = $name"  
}
```

Reports the names of all of the partitions in the design.

dbForEachHeadRepCell

dbForEachHeadRepCell

```
dbForEachHeadRepCell headPtr cellPtrVar body
```

Returns names of the representative cells for the specified cell. The representative cell is the representative for all (timing library) "sibling" cells. Applications that need to see only one cell (for example the LEF library cell), should see the representative cell because sibling cells are created for different timing libraries (min/max) that might not have all of the required (LEF) information.

Parameters

<i>headPtr</i>	Address of Head.
<i>cellPtrVar</i>	Address of cell.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForEachHeadRepCell [dbgHead] cellPtr {
```

```
puts "[dbCellName $cellPtr]"  
}
```

Returns:

XOR2X4

XOR2XL

RF1R1WX2

RF2R1WX2

FILL1

FILL16

FILL2

FILL32

FILL4

FILL64

FILL8

Reports the cell names that are representative.

dbForEachHeadRuleList

```
dbForEachHeadRuleList headPtr ruleListPtrVar body
```

Returns the type of rule for selected rules for the design.

Parameters

<i>headPtr</i>	Address of Head.
<i>ruleListPtrVar</i>	Variable of addresses of rule lists.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadRuleList [dbgHead] ruleListPtr {  
    set obj_type [dbObjType $ruleListPtr]  
    puts "obj_type = $obj_type"  
}
```

Reports the type of rule for all rules in the design.

dbForEachHeadSel

```
dbForEachHeadSel headPtr selPtrVar body
```

Returns the type of selected objects for the design.

Parameters

<i>headPtr</i>	Address of Head.
<i>selPtrVar</i>	Variable of addresses of selected objects.
<i>body</i>	

Command Order

Use this command after design import and after at least one object has been selected.

Example

```
dbForEachHeadSel [dbgHead] selPtr {
```

```
set obj_type [dbObjType $selPtr]  
puts "obj_type = $obj_type"  
}
```

Reports the type of all objects selected in the design.

dbForEachHeadSelPtr

dbForEachHeadSelPtr headPtr ptrVar body

Returns the addresses of selected objects.

Parameters

<i>headPtr</i>	Address of Head.
<i>ptrVar</i>	Addresses of selected objects.
<i>body</i>	

Command Order

Use this command after design import and after at least one object has been selected.

Example

```
dbForEachHeadSelPtr [dbgHead] selPtr {  
  
set obj_type [dbObjType $selPtr]  
puts "obj_type = $obj_type"  
}  
  
obj_type = dbcObjHInst  
obj_type = dbcObjInst
```

There is a module and an instance selected.

dbForEachHeadSite

`dbForEachHeadSite headPtr sitePtrVar body`

Returns the names of the specified sites in the design.

Parameters

<code>headPtr</code>	Address of Head.
<code>sitePtrVar</code>	Addresses of sites
<code>body</code>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadSite [dbgHead] sitePtr {  
    set name [dbTechSiteName $sitePtr]  
    puts "name = $name"  
}
```

Returns the names of all the sites in the design.

dbForEachHeadTechSite

`dbForEachHeadTechSite headPtr techSitePtrVar body`

Returns the names of the specified tech sites in the design.

Parameters

<code>headPtr</code>	Address of Head.
<code>techSitePtrVar</code>	Addresses of tech sites.
<code>body</code>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadTechSite [dbgHead] techSitePtr {  
    set name [dbTechSiteName $techSitePtr]  
    puts "name = $name"  
}
```

Returns the names of all the tech sites in the design.

dbForEachHeadTileCell

`dbForEachHeadTileCell headPtr tileCellPtrVar body`

Returns the addresses of the tile cell pins in the design.

Parameters

<i>headPtr</i>	Address of Head.
<i>timeCellPtrVar</i>	Addresses of tile cell.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```
dbForEachHeadTileCell [dbgHead] tileCellPtr {  
    dbForEachTileCellTilePin $tileCellPtr tileCellPin {  
        puts $tileCellPin"  
    }  
}
```

Returns all of the addresses of the tile cell pins.

dbForEachHeadTimeLib

```
dbForEachHeadTimeLib headPtr timeLibPtrVar body
```

Returns the names of the timing libraries in the design.

Parameters

<i>headPtr</i>	Address of Head.
----------------	------------------

<i>timeLibPtrVar</i>	Addresses of timing libraries.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadTimeLib [dbgHead] selPtr {  
    set name [dbTimeLibName $selPtr]  
    puts "name = $name"  
}
```

Returns:

`name = t25lib_t`

There is only one timing library in the design: `t25lib_t`.

dbForEachHeadTlsCell

```
dbForEachHeadTlsCell headPtr cellPtrVar body
```

Returns the addresses of the timing library cells in the design.

Parameters

<i>headPtr</i>	Address of Head.
<i>cellPtrVar</i>	Addresses of timing library cells.
<i>body</i>	

Command Order

Use this command after importing the design.

dbForEachHInstHInst

dbForEachHInstHInst hinstPtr dpinstPtrVar body

Returns object name and type of the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of HInst.
<i>dpinstPtrVar</i>	Address of instances and hierarchical instances.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHInstHInst [dbGetHInstByName SH17] ptr {  
    set obj_type [dbObjType $ptr]  
    puts "type = $objtype"  
    if { $obj_type == "dbcObjInst" } {  
        set name [dbInstName $ptr]  
        puts "name = $name"  
    } else {  
        set name [dbHInstName $ptr]
```

```
    puts "name = $name"  
}  
}
```

Returns:

```
type = dbcObjInst  
name = SH17/I341  
type = dbcObjInst  
name = SH17/I340
```

Returns the object type and name for the instances of `SH17`.

dbForEachHInstHTerm

`dbForEachHInstHTerm hinstPtr dpinstPtrVar body`

Returns types for the specified hierarchical instances.

Parameters

<code>hinstPtr</code>	Address of hierarchical instance.
<code>dpinstPtrVar</code>	Address of h-term
<code>body</code>	

Command Order

Use this command after design import.

Example

```
dbForEachHInstHTerm $objPtr dpinstPtr {  
    set obj_type [dbcObjType $dpinstPtr]
```

```
puts "type = $obj_type"  
}
```

Returns:

```
type = dbcObjHTerm  
type = dbcObjHTerm  
type = dbcObjHTerm
```

The type is h-term.

dbForEachHInstTreeHInst

dbForEachHInstTreeHInst hinstPtr dpinstPtrVar body

Returns the type of all the hierarchical instances in the tree for the specified hierarchical instance

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
<i>dpinstPtrVar</i>	Addresses of hierarchical instances.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHInstTreeHInst $objPtr dpinstPtr {  
set obj_type [dbObjType $dpinstPtr]  
puts "type = $obj_type"
```

}

Returns:

```
type = dbcObjHInst  
type = dbcObjHInst  
type = dbcObjHInst
```

The type is hierarchical instance.

dbForEachHInstTreeInst

dbForEachHInstTreeInst hinstPtr dpinstPtrVar body

Returns the type for the specified hierarchical instances.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
<i>dpinstPtrVar</i>	Addresses of instances.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHInstTreeInst $objPtr dpinstPtr {  
    set obj_type [dbObjType $dpinstPtr]  
    puts "type = $obj_type"  
}
```

Returns:

type = dbcObjInst

type = dbcObjInst

type = dbcObjInst

The type is instance.

dbForEachHInstTreeNet

dbForEachHInstTreeNet *hinstPtr dpinstPtrVar body*

Returns types for the specified hierarchical instances.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
<i>dpinstPtrVar</i>	Addresses of instances.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHInstTreeNet $objPtr dpinstPtr {  
    set obj_type [dbObjType $dpinstPtr]  
    puts "type = $obj_type"  
}
```

Returns:

```
type = dbcObjNet
type = dbcObjNet
type = dbcObjNet
```

The type is net.

dbForEachInstInputTerm

```
dbForEachInstInputTerm instPtr termPtrVar body
```

Returns names of input terminals for the specified instance.

Parameters

<i>headPtr</i>	Address of instance.
<i>termPtrVar</i>	Addresses of terminals.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachInstInputTerm $objPtr termPtr {
    set name [dbTermName $termPtr]
    puts "name = $name"
}
```

Returns;

```
name = RP1
```

```
name = WBUFEN
name = WP0
name = WP1
```

The input terminals are RP1, WBUFEN, WP0, and WP1.

dbForEachInstOutputTerm

dbForEachInstOutputTerm *instPtr termPtrVar body*

Returns names of active output terminals for the specified instance. The command does not return the names of unconnected terminals.

Parameters

<i>instPtr</i>	Address of instance
<i>termPtrVar</i>	Addresses of terminals.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachInstOutputTerm $objPtr termPtr {
    set name [dbTermName $termPtr]
    puts "name = $name"
}
```

Returns:

```
name = DATA_29
```

```
name = DATA_30
name = DATA_31
```

The output terminals are DATA29, DATA30, and DATA31.

dbForEachInstTerm

```
dbForEachInstTerm instPtr termPtrVar body
```

Returns all of the active terminals for the specified instance. The command does not return the names of unconnected terminals.

Parameters

<i>instPtr</i>	Address of instance.
<i>termPtrVar</i>	Addresses of terminals
<i>body</i>	

dbForEachLayerShapeShape

```
dbForEachLayerShapeShape layerShapePtr shapePtrVar body
```

Returns all of the shapes for the specified layer shape list.

Parameters

<i>layerShapePtr</i>	Address of list of layer shapes.
----------------------	----------------------------------

<i>shapePtrVar</i>	Addresses of shape
<i>body</i>	

Command Order

Use this command after

Example

```
proc printOneLayerShape {ls} {
    if {[dbIsLayerShapeShapeList $ls]} {
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
        dbForEachLayerShapeShape $ls s {
            printOneShape $s
        }
    } elseif {[dbIsLayerShapeVia $ls]} {
        puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
        [dbLayerShapeViaViaLoc $ls]"
    } else {
        puts "*** error *** layerShape"
    }
}
```

dbForEachLefFPortLayerShape

dbForEachLefFPortLayerShape leffPortPtr layerShapePtrVar body

Returns the addresses of layer shapes for the specified LEF FPort.

Parameters

<i>lefFPortPtr</i>	Address of LEF FPort.
<i>layerShapePtrVar</i>	Addresses of layer shapes.
<i>body</i>	

Command Order

Use this command after importing the design.

Example

```

proc printOneLayerShape {ls} {
    if {[dbIsLayerShapeShapeList $ls]} {
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
        dbForEachLayerShapeShape $ls s {
            printOneShape $s
        }
    } elseif {[dbIsLayerShapeVia $ls]} {
        puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
        [dbLayerShapeViaViaLoc $ls]"
    } else {
        puts "*** error *** layerShape"
    }
}

proc printOneLefPort {p} {
    if {[dbIsLefPortClassNone $p]} {
        puts "lefPort class <none>"
    } elseif {[dbIsLefPortClassCore $p]} {
        puts "lefPort class <core>"
    } elseif {[dbIsLefPortClassUndefined $p]} {

```

```
puts "lefPort class <undefined>"  
} dbForEachLefPortLayerShape $p ls {  
printOneLayerShape $ls  
}  
}
```

dbForEachListElem

dbForEachListElem listPtr ptrVar body

Returns the address for each element in the list.

Parameters

<i>listPtr</i>	Address of list.
<i>ptrVar</i>	Addresses of elements.
<i>body</i>	

Command Order

Use this command after design import.

dbForEachListPtr

dbForEachListPtr listPtr ptrVar body

Returns the addresses of elements in list.

Parameters

<i>listPtr</i>	Address of list.
<i>ptrVar</i>	Addresses of elements.
<i>body</i>	

Command Order

Use this command after design import.

dbForEachMetalLayer

`dbForEachMetalLayer listPtr ptrVar body`

Returns the addresses of the metal layer.

Parameters

<i>layerPtrVar</i>	Address of layers.
<i>body</i>	

Command Order

Use this command after design import.

dbForEachNetGroupNet

dbForEachNetGroupNet *netGroupPtr netPtrVar body*

Returns names of nets for the specified group.

Parameters

<i>netGroupPtr</i>	Address of net group
<i>netPtrVar</i>	Addresses of net.
<i>body</i>	

Command Order

Use this command after design import and creation of at least one net group.

Example

```
createNetGroup my_net_group  
addNetToNetGroup my_net_group BLU  
addNetToNetGroup my_net_group XTL1  
set netGroupPtr [dbGetNetGroupByName my_net_group]
```

Returns:

0x9660650

Then:

```
dbForEachNetGroupNet $netGroupPtr netPtr {  
    set name [dbNetName $netPtr]  
    puts "name: $name"  
}
```

Returns:

name: BLU

name: XTL1

The nets `BLU` and `XTL1` are the nets in the net group `my_net_group`.

dbForEachNetHNet

dbForEachNetHNet

`dbForEachNetHNet netPtr hNetPtrVar body`

Iterates through all hierarchical nets associated with the specified net, and executes the Tcl commands in the specified script.

Parameters

<code>netPtr</code>	Address of net.
<code>hNetPtrVar</code>	Address of hierarchical net.
<code>body</code>	Tcl commands to run for each <code>hNetPtrVar</code> .

Command Order

Use this command after importing the design.

Example

The following script returns the names of the hierarchical nets for net `clk1`:

```
dbForEachNetHNet [dbGet -p top.nets.name clk1] hNetPtr { Puts [dbGet  
$hNetPtr.name] }
```

dbForEachNetMsNetGroup

`dbForEachNetMsNetGroup netPtr netGroupPtrVar body`

Returns the names of the mixed signal net groups for the specified net.

Parameters

<code>netPtr</code>	Address of net.
<code>netGroupPtrVar</code>	Addresses of net groups.
<code>body</code>	

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbForEachNetMsNetGroup [dbGetNetByName BLU] netGroupPtr {  
    set name [dbGroupName $netGroupPtr]  
    puts "name = $name"  
}
```

Prints the name of all mixed signal net groups for the specified net.

dbForEachNetNetGroup

`dbForEachNetNetGroup netPtr netGroupPtrVar body`

Returns the names of the net groups for the specified net.

Parameters

<i>netPtr</i>	Address of net.
<i>netGroupPtrVar</i>	Addresses of net groups.
<i>body</i>	

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbForEachNetNetGroup [dbGetNetByName BLU] netGroupPtr {  
    set name [dbGroupName $netGroupPtr]  
    puts "name = $name"  
}
```

Prints the names of all net groups for the specified net.

dbForEachNetOutputTerm

```
dbForEachNetOutputTerm netPtr termPtrVar body
```

Returns names of output terminals for the specified net.

Parameters

<i>netPtr</i>	Address of net.
<i>termPtrVar</i>	Addresses output terminals.

<i>body</i>	
-------------	--

Command Order

Use this command after design import.

Example

```
dbForEachNetOutputTerm [dbGetNetByName MAA3] term {  
    set name [dbTermName $term]  
    puts "name: $name"  
}
```

Returns:

name: MAA3

The only output terminal for the net MAA3 is MAA3.

dbForEachNetTerm

dbForEachNetTerm *netPtr* *termPtrVar* *body*

Returns names of terminals on the specified net.

Parameters

<i>netPtr</i>	Address of net.
<i>termPtrVar</i>	Addresses of terminals.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachNetTerm [dbGetNetByName BLU] termPtr {  
    set name [dbTermName $termPtr]  
    puts "name = $name"  
}
```

Prints the names of all terminals for the net `BLU`.

dbForEachNetTermS

`dbForEachNetTermS netPtr termPtrVar body`

Returns names of terminals on the specified special net.

Parameters

<code>netPtr</code>	Address of special net.
<code>termPtrVar</code>	Addresses of terminals.
<code>body</code>	

Command Order

Use this command after design import.

Example

```
dbForEachNetTermS [dbGetNetByName VDD] termPtr {
```

```
set name [dbTermName $termPtr]  
puts "name = $name"  
}
```

Prints the name of all terminals on VDD.

dbForEachNetWire

dbForEachNetWire *netPtr* *wirePtrVar* *body*

Returns lengths of wires on the specified net.

Parameters

<i>netPtr</i>	Address of net.
<i>wirePtrVar</i>	Addresses of wires.
<i>body</i>	

Command Order

Use this command after route.

Example

```
dbForEachNetWire [dbGetNetByName BLU] wirePtr {  
set len [dbWireLen $wirePtr]  
puts "len = $len"  
}
```

Returns:

len = 1000

```
len = 600
len = 1000
len = 500
len = 350
len = 0
len = 0
len = 0
len = 961000
len = 0
len = 10920
len = 0
```

The various wire lengths are reported for the segments of the net `BLU`.

dbForEachObjProp

`dbForEachObjProp` *Ptr propPtrVar*

Returns the name and value of properties at the specified address.

Parameters

<i>Ptr</i>	Address of object.
<i>propPtrVar</i>	Addresses of properties.

Command Order

Use this command after applying properties.

Example

```
dbForEachObjProp $route prop {  
    set propName [ dbPropertyName $prop ]  
    set propVal [ dbPropertyValue $prop ]  
    puts $fout " PropName = $propName"  
    puts $fout " PropValue = $propVal\n"  
}
```

Writes to a file the property name and value.

dbForEachPathTerm

dbForEachPathTerm pathPtr termPtrVar body

Returns the addresses for each terminal for the specified path.

Parameters

<i>pathPtr</i>	Address of path.
<i>termPtrVar</i>	Addresses of terminals.
<i>body</i>	

Command Order

Use this command after timing analysis.

dbForEachPinGroupPin

`dbForEachPinGroupPin pinGroupPtr ftermPtr body`

Returns names for each terminal in the specified pin group.

Parameters

<i>pinGroupPtr</i>	Address of pin group.
<i>ftermPtr</i>	Addresses of f-terminals.
<i>body</i>	

Command Order

Use this command after timing analysis.

Example

```
set pinGroupPtr [dbGetPinGroupByName my_group]
0x9660650
dbForEachPinGroupPin $pinGroupPtr ftermPtr {
    set name [dbFTermName $ftermPtr]
    puts "name: $name"
}
```

Returns:

```
name: BLU
name: XTL1
```

The f-terms `BLU` and `XTL1` are in the pin group `my_group`.

dbForEachPowerDomain

`dbForEachPowerDomain [headPtr] powerDomainPtr body`

Returns ground nets for the specified power domains; returns nothing if there are none.

Parameters

<code>headPtr</code>	Address of head.
<code>powerDomainPtr</code>	Addresses of power domains.
<code>body</code>	

Command Order

Use this command after design import.

Example

```
dbForEachPowerDomain [dbgHead] powerDomainPtr {  
    set g [dbPowerDomainGNet $powerDomainPtr]  
    puts "ground = $g"  
}
```

Reports the ground nets.

dbForEachPowerDomainPad

`dbForEachPowerDomainPad powerDomainPtr instPtrVar body`

Returns the addresses of instances for the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
<i>instPtrVar</i>	Addresses of instances.
<i>body</i>	

Command Order

Use this command after power analysis.

dbForEachPtnCloneList

`dbForEachPtnCloneList ptnPtr cloneListPtrVar body`

Returns the rotation of clones for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>cloneListPtrVar</i>	Address of clones.
<i>body</i>	

Command Order

Use this command after importing a design with partitioned clones.

Example

```
dbForEachPtnCloneList [dbGetPtnByName sheet1] cloneListPtr {
```

```
set ret [dbCloneListOrient $cloneListPtr]
puts "orient = $ret"
}
```

Returns:

```
orient = dbcR0
```

The clone has no rotation.

dbForEachRowClusterRow

```
dbForEachRowClusterRow rowClusterPtr stdRowPtrVar body
```

Returns the boxes of row areas.

Parameters

<i>rowClusterPtr</i>	Address of row cluster.
<i>stdRowPtrVar</i>	Addresses of standard rows.
<i>body</i>	

Command Order

Use this command after importing the design and creating row clusters.

Example

```
dbForEachFPlanAIORowCluster [dbHeadFPlan $rowClusterPtr {
    dbForEachRowClusterRow $rowClusterPtr stdRowPtr {
        set box [dbRowClusterBox $stdRowPtr]
        puts "box = $box"
```

```
}
```

```
}
```

Reports all of the boxes for the area I/O rows.

dbForEachSNetBox

```
dbForEachSNetBox snetPtr stripBoxPtrVar body
```

Returns the addresses of snet boxes.

Parameters

<i>snetPtr</i>	Addresses of snet.
<i>stripBoxPtrVar</i>	Addresses of stripboxes.
<i>body</i>	

Command Order

Use this command after importing the design and creating at least one power or ground strap.

Example

```
dbForEachSNetBox [dbGetSNetByName vdd] stripBoxPtr
set box [dbStripBoxBox $stripBoxPtr]
puts "box: $box"
}
}
```

The names of the strip boxes are output for the snet vdd.

dbForEachSNetStripBox

`dbForEachSNetStripBox snetPtr stripBoxPtrVar body`

Iterates through all special wires (special vias excluded) associated with the specified special net, and executes the Tcl commands in the specified script.

Parameters

<i>snetPtr</i>	Address of the special net.
<i>stripBoxPtrVar</i>	Variable that is set to a different special wire during each iteration.
<i>body</i>	Tcl commands to run for each <i>stripBoxPtrVar</i>

Command Order

Use this command after importing the design and creating at least one power or ground strap.

Example

The following script returns the shape for each vdd special wire in the design:

```
dbForEachSNetStripBox [dbGetSNetByName vdd] stripBoxPtr {  
set shape [dbStripBoxShape $stripBoxPtr]puts "shape: $shape"  
}
```

dbForEachTileCellTilePin

`dbForEachTileCellTilePin tileCellPtrVar tilePinPtrVar body`

Returns the addresses of tile pins for the specified tile cell.

Parameters

<code>tileCellPtrVar</code>	Address of tile cell.
<code>tilePinPtrVar</code>	Addresses of tile pins.
<code>body</code>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadTileCell [dbgHead] tileCellPtr {  
    dbForEachTileCellTilePin $tileCellPtr tilePinPtr{  
        connectTilePinToGlobalNet -all -net $netPtr -pin $tilePinPtr -power  
    }  
}
```

Connects the tile pins to power.

dbForEachTimeLibCell

`dbForEachTimeLibCell timeLibPtrVar cellPtrVar body`

Iterates through all cells associated with the specified timing library and executes the Tcl commands in the specified script.

Parameters

<i>timeLibPtrVar</i>	Address of timing library.
<i>cellPtrVar</i>	Address of cell.
<i>body</i>	Tcl commands to run for each <i>cellPtrVar</i> .

Command Order

Use this command after importing the design.

Example

The following script returns the names of the timing library cells in the design:

```
dbForEachHeadTimeLib [dbgHead] timeLibPtr {  
    dbForEachTimeLibCell $timeLibPtr cellPtr {  
        set name [dbcName $cellPtr]  
        puts "name: $name"  
    }  
}
```

Returns:

```
name: BUFX1  
name: BUFX2  
name: BUFX4  
name: BUFX8
```

There are four cells in the libraries for the design.

dbForEachTimeLibOpCond

dbForEachTimeLibOpCond *timeLibPtrVar opCondPtrVar body*

Returns names of timing libraries.

Parameters

<i>timeLibPtrVar</i>	Address of timing library.
<i>opCondPtrVar</i>	Addresses of operating conditions.
<i>body</i>	

Command Order

Use this command after design import.

Example

```
dbForEachHeadTimeLib [dbgHead] timeLibPtr {  
    dbForEachTimeLibOpCond $timeLibPtr opCondPtr {  
        set name [dbOpCondName $opCondPtr]  
        puts "name: $name"  
    }  
}
```

Returns:

```
name: BEST  
name: TYPICAL  
name: TYP-WORST  
name: WORST
```

There are 4 timing libraries for the design.

dbForEachVNetVTerm

`dbForEachVNetVTerm vnetPtr vtermPtrVar body`

Returns the addresses of Verilog terminals for the specified Verilog net.

Parameters

<code><i>vnetPtr</i></code>	Address of Verilog net.
<code><i>vtermPtrVar</i></code>	Addresses of Verilog terms.
<code><i>body</i></code>	

Command Order

Use this command after design import.

dbFPinAccessDir

`dbFPinAccessDir fpinPtr`

Returns the access direction of the pin.

Parameters

<code><i>fpinPtr</i></code>	Address of fpin.
-----------------------------	------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachHeadCell [dbgHead] cellPtr {  
    if { [dbIsCellCdumpDefined $cellPtr] } {  
        set name [dbCellName $cellPtr]  
        puts "cell name: $name"  
        dbForAllCellFTerm $cellPtr ftermPtr {  
            set name [dbFTermName $ftermPtr]  
            puts " + fterm name: $name"  
            dbForEachFTermFPin $ftermPtr FPinPtr {  
                set ADir [dbFPinAccessDir $FPinPtr]  
                puts "adir = $ADir"  
            } } } }
```

Returns:

```
.cell name: BUF_FT  
+ fterm name: Z  
adir = NWSEUD  
+ fterm name: A  
adir = NWSEUD
```

All access directions are available for the terminals `A` and `Z`.

dbFPinADir

```
dbFPinADir fpinPtr
```

Returns the direction of the pin.

Parameters

<i>fpinPtr</i>	Address of fpin
----------------	-----------------

Command Order

Use this command after the design is loaded.

Example

```
dbForEachHeadCell [dbgHead] cellPtr {  
    if { [dbIsCellCdumpDefined $cellPtr] } {  
        set name [dbCellName $cellPtr]  
        puts "cell name: $name"  
        dbForAllCellFTerm $cellPtr ftermPtr {  
            set name [dbFTermName $ftermPtr]  
            puts " + fterm name: $name"  
            dbForEachFTermFPin $ftermPtr FPinPtr {  
                set ADir [dbFPinADir $FPinPtr]  
                puts "adir = $ADir"  
            }}}}}
```

Returns:

```
cell name: BUF_FT  
+ fterm name: Z  
adir = 63  
+ fterm name: A  
adir = 63
```

The terminals are `A` and `Z` for the cell `BUF_FT`.

dbFPinLayer

dbFPinLayer *fpinPtr*

Returns the address of the metal layer associated with the specified fpin.

Parameters

<i>fpinPtr</i>	Address of fpin
----------------	-----------------

Command Order

Use this command after design import.

Example

```
dbForEachHeadCell [dbgHead] cellPtr {  
    if { [dbIsCellCdumpDefined $cellPtr] } {  
        set name [dbCellName $cellPtr]  
        puts "cell name: $name"  
        dbForAllCellFTerm $cellPtr ftermPtr {  
            set name [dbFTermName $ftermPtr]  
            puts " + fterm name: $name"  
            dbForEachFTermFPin $ftermPtr FPinPtr {  
                set ADir [dbFPinLayer $FPinPtr]  
                puts "layer = $ADir"  
            }}}}}
```

Returns:

cell name: BUF_FT

```
+ fterm name: Z
layer = 0xc7f7750
+ fterm name: A
layer = 0xc7f7750
```

The layer addresses are returned for the f-terms A and Z.

dbFPinLoc

```
dbFPinLoc fpinPtr
```

Returns the location of the fpin (relative to the origin of the cell master).

Parameters

<i>fpinPtr</i>	Address of fpin.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachHeadCell [dbgHead] cellPtr {
    if { [dbIsCellCdumpDefined $cellPtr] } {
        set name [dbCellName $cellPtr]
        puts "cell name: $name"
        dbForAllCellFTerm $cellPtr ftermPtr {
            set name [dbFTermName $ftermPtr]
            puts " + fterm name: $name"
        }
    }
}
```

```
set ADir [dbFPinLoc $FPinPtr]
puts "loc = $ADir"
} } }
```

Returns:

```
.cell name: BUF_FT
+ fterm name: Z
loc = 10450 22500
```

The location for f-term Z is (10.45,22.5).

dbFPinNext

dbFPinNext *fpinPtr*

Returns the address of the next fpin. The command returns 0x0 if there is no next fpin.

Parameters

<i>fpinPtr</i>	Address of fpin.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
set next [dbFPinNext $FPinPtr]
```

Returns:

0x0

There is no next fpin.

dbFPinNrAPin

dbFPinNrAPin *fpinPtr*

Returns the number of access pins associated with fpin.

Parameters

<i>fpinPtr</i>	Address of fpin
----------------	-----------------

Command Order

Use this command after importing the design.

Example

```
dbForEachHeadCell [dbgHead] cellPtr {  
    if { [dbIsCellCdumpDefined $cellPtr] } {  
        set name [dbCellName $cellPtr]  
        puts "cell name: $name"  
        dbForAllCellFTerm $cellPtr ftermPtr {  
            set name [dbFTermName $ftermPtr]  
            puts " + fterm name: $name"  
            dbForEachFTermFPin $ftermPtr FPinPtr {  
                set ADir [dbFPinNrAPin $FPinPtr]  
                puts "nr = $ADir"  
            }}}}}
```

Returns:

cell name: BUF_FT

```
+ fterm name: Z
```

```
nr = 0
```

```
+ fterm name: A
```

```
nr = 0
```

The number of access pins for f-term's Z and A are 0.

dbFPinSide

```
dbFPinSide fpinPtr
```

Returns the side the fpin is on. Possible values are:

```
dbcN
```

```
dbcW
```

```
dbcS
```

```
dbcE
```

```
dbcU
```

```
dbcD
```

```
dbcUnknownDir
```

Parameters

<i>fpinPtr</i>	Address of fpin.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachHeadCell [dbgHead] cellPtr {
```

```
if { [dbIsCellCdumpDefined $cellPtr] } {  
  
set name [dbCellName $cellPtr]  
  
puts "cell name: $name"  
  
dbForAllCellFTerm $cellPtr ftermPtr {  
  
set name [dbFTermName $ftermPtr]  
  
puts " + fterm name: $name"  
  
dbForEachFTermFPin $ftermPtr FPinPtr {  
  
set ADir [dbFPinSide $FPinPtr]  
  
puts "side = $ADir"  
}}}}
```

Returns:

```
cell name: BUF_FT  
+ fterm name: Z  
side = dbcN  
+ fterm name: A  
side = dbcE
```

f-term Z is on the north side, while f-term A is on the east side.

dbFPinWidth

```
dbFPinWidth fpinPtr
```

Returns the width of the fpin. The default value is -1.

Parameters

<i>fpinPtr</i>	Address of fpin.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachHeadCell [dbgHead] cellPtr {  
    if { [dbIsCellCdumpDefined $cellPtr] } {  
        set name [dbCellName $cellPtr]  
        puts "cell name: $name"  
        dbForAllCellFTerm $cellPtr ftermPtr {  
            set name [dbFTermName $ftermPtr]  
            puts " + fterm name: $name"  
            dbForEachFTermFPin $ftermPtr FPinPtr {  
                set ADir [dbFPinWidth $FPinPtr]  
                puts "width = $ADir"  
            }}}}}
```

Returns:

```
cell name: BUF_FT  
+ fterm name: Z  
width = -1  
+ fterm name: A  
width = -1
```

Both f-terms are at the default width.

dbFPinZ

```
dbFPinZ fpinPtr
```

Returns the metal layer (for example, 1 denotes *meta/1*, 2 denotes *meta/2*).

Parameters

<i>fpinPtr</i>	Address of fpin
----------------	-----------------

Command Order

This command can be used after importing the design.

Example

```
dbForEachHeadCell [dbgHead] cellPtr {  
    if { [dbIsCellCdumpDefined $cellPtr] } {  
        set name [dbCellName $cellPtr]  
        puts "cell name: $name"  
        dbForAllCellFTerm $cellPtr ftermPtr {  
            set name [dbFTermName $ftermPtr]  
            puts " + fterm name: $name"  
            dbForEachFTermFPin $ftermPtr FPinPtr {  
                set ADir [dbFPinZ $FPinPtr]  
                puts "Z = $ADir"  
            }}}}}
```

Returns:

```
cell name: BUF_FT  
+ fterm name: Z  
Z = 1  
+ fterm name: A  
Z = 1
```

Pins A and Z are both on *metal1*.

dbFPlanBox

dbFPlanBox *fplanPtr*

Returns the coordinates of the floorplan (die/block size)

Parameters

<i>fplanPtr</i>	Address of fplan.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
set fp [dbHeadFPlan]
```

Returns:

```
0xd4da4c0
```

```
dbFPlanBox $fp
```

Returns:

```
0 0 7237450 7232600
```

The coordinates of the die area are (0, 0) (7237.45, 7232.6).

dbFPlanCellPadList

dbFPlanCellPadList *fplanPtr*

Returns the address of the cell pad list. The command returns 0x0 if there is no cell pad list.

Parameters

<i>fplanPtr</i>	Address of FPlan.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbFPlanCellPadList [dbHeadFPlan]
```

Returns:

0x0

There is no cell pad list for this design.

dbFPlanConstraintList

```
dbFPlanConstraintList fplanPtr
```

Returns the address of the list of constraints for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of fplan.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
set fp [dbHeadFPlan]
```

Returns:

0xd4da4c0

```
dbFPlanConstraintList $fp
```

Returns:

0xce51780

The address of the constraint list is returned.

dbFPlanCoreBox

```
dbFPlanCoreBox fplanPtr
```

Returns the lower left and upper right coordinates for the core area (rows).

Parameters

<i>fplanPtr</i>	Address of fplan.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
set fp [dbHeadFPlan]
```

Returns:

0xd4da4c0

```
dbFPlanoreBox $fp
```

Returns:

```
430300 430300 6802300 6802300
```

The core area coordinates are (430.3 430.3) (6802.3 6802.3)

dbFPlanDefaultTechSite

dbFPlanDefaultTechSite

```
dbFPlanDefaultTechSite fplanPtr
```

Returns the address of the default tech site for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of fplan.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
set fp [dbHeadFPlan]
```

Returns:

```
0xd4da4c0
```

```
dbTEchSiteName [dbFPlanDefaultTechSite $fp]
```

Returns:

tsm3site

The default tech site name is tsm3site.

dbFPlanEqualizeAllPtnCellHInst

dbFPlanEqualizeAllPtnCellHInst

Synchronizes all partition clone floorplan to the master partition.

Command Order

Use this command after importing the design.

Example

dbFPlanEqualizeAllPtnCellHInst MEM_DSCAN

Creates a cover for instance MEM_DSCAN.

dbFPlanFlip

dbFPlanFlip *fplanPtr*

Returns the flip status of the rows for the specified floorplan. Possible values are:

dbcFlipOdd

dbcFlipEven

dbcNoFlip

Parameters

<i>fplanPtr</i>	Address of fplan.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
set fp [dbHeadFPlan]
```

Returns:

```
0xd4da4c0
```

```
dbFPlanFlip $fp
```

Returns:

```
dbcFlipOdd
```

The flip status is FlipOdd.

dbFPlanGroupList

```
dbFPlanGroupList fplanPtr
```

Returns the address of the list of groups (0x0 implies no groups).

Parameters

<i>fplanPtr</i>	Address of fplan.
-----------------	-------------------

Command Order

Use this command after importing the design (more useful after groups have been defined).

Example

```
set fp [dbHeadFPlan]
```

Returns:

```
0xd4da4c0
```

```
dbFPlanGroupList $fp
```

Returns:

```
0x0
```

There are no groups defined for this design.

dbFPlanInstIGLen

```
dbFPlanInstIGLen fplanPtr
```

Returns the instance grid length used by the instances for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of fplan.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
set fp [dbHeadFPlan]
```

Returns:

```
0xd4da4c0
```

```
dbFPlanIGLen $fp
```

Returns:

```
1478412.0
```

```
dbFPlanInstIGLen $fp
```

Returns:

```
1303052.0
```

Creates a cover for instance `MEM_DSCAN`.

dbFPlanUpdateGmapByPrerouteAsObs

```
dbFPlanUpdateGmapByPrerouteAsObs fplanPtr
```

Updates the floorplan to the current layers marked as obstructions for preroutes.

Parameters

<i>fplanPtr</i>	Address of fplan.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbFPlanUpdateGmapByPrerouteAsObs [dbHeadFPlan]
```

Updates the floorplan.

dbFreezeAllInst

dbFreezeAllInst

Marks all standard cells in the core as fixed.

Command Order

Use this command after importing the design place.

Example

dbFreezeAllInst

Marks all the instances as fixed.

dbFTermBox

dbFTermBox *ftermPtr*

Returns the box occupied by the f-term (2147483647 2147483647 -2147483648 -2147483648 indicates a non-valid box).

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

```
dbFTermBox $objPtr
```

Returns:

```
2147483647 2147483647 -2147483648 -2147483648
```

There is no box.

dbFTermBTerm

```
dbFTermBTerm ftermPtr
```

Returns the address of bus term (0x0 indicates there is not one).

Parameters

<i>ftermPtr</i>	Address of f-term
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

(with a pin selected)

```
set s [dbHeadSelList]
```

Returns:

```
0x139e88b8
```

```
set objPtr [dbSelPtr $s]
```

Returns:

0xfb11320

```
set obj_type [dbObjType $objPtr]
```

Returns:

dbcObjFTerm

```
dbFTermBTerm $objPtr
```

Returns:

0x0

There are no bus terminals associated with this f-term.

dbFTermBus

```
dbFTermBus ftermPtr
```

Returns the address of the bus for the specified f-term (0x0 indicates there is not one).

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

```
dbFTermBus $objPtr
```

Returns:

0x0

There is no bus associated with the f-term.

dbFTermCell

dbFTermCell *ftermPtr*

Returns the address of the cell associated with the specified f-term.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

(with a pin selected)

```
set s [dbHeadSelList]
```

Returns:

0x139e88b8

```
set objPtr [dbSelPtr $s]
```

Returns:

0xfb11320

```
set obj_type [dbObjType $objPtr]
```

Returns:

dbcObjFTerm

```
dbFTermCell $objPtr
```

Returns:

0xf930e40

The address of the f-term cell is 0xf930e40

dbFTermDepth

```
dbFTermDepth ftermPtr
```

Returns the depth of the associated f-term, -1 if using default depth as specified in the technology file.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellFTerm $cellPtr ftermPtr {  
    set depth [dbFTermDepth $ftermPtr]  
    puts "depth = $depth"  
}
```

Returns the depth of all f-terms for the specified cell.

G

dbGeomBoxBox

`dbGeomBoxBox geomBoxPtr`

Returns the address of the geometry box specified by the geometry box.

Parameters

<code><i>geomBoxPtr</i></code>	Address of geometry box.
--------------------------------	--------------------------

Command Order

Use this command after invoking Innovus.

Example

`dbGeomBoxBox $geomBoxPtr`

Returns:

`0xb0874242`

The address of the geometry box is `0xb0874242`.

dbGeomLineEndPt

`dbGeomLineEndPt geomLinePtr`

Returns the end point for the specified geometry line.

Parameters

geomLinePtr

Address of geometry line.

Command Order

Use this command after invoking Innovus and have created at least one geometry line.

Example

```
set line [addCustomLine line 1000 1000 4000 4000]  
dbGeomLineEndPt $line
```

Returns:

4000000 4000000

The endpoint is 4000.0 4000.0.

dbGeomLineStartPt

`dbGeomLineStartPt geomLinePtr`

Returns the start point for the specified geometry line.

Parameters

geomLinePtr

Address of geometry box.

Command Order

Use this command after invoking Innovus and creating at least one geometry line.

Example

```
set line [addCustomLine line 1000 1000 4000 4000]  
dbGeomLineStartPt $line
```

Returns:

1000000 100000

The start point is 1000.0 1000.0.

dbGeomListCustLayer

`dbGeomListCustLayer geomGeomListPtr`

Returns the address of the custom layers for the specified *geomGeomList*.

Parameters

<i>geomGeomListPtr</i>	Address of geometry list.
------------------------	---------------------------

Command Order

Use this command after invoking Innovus.

Example

```
dbForEachFPlanGeomList [dbHeadFPlan] g {  
    dbGeomListCustLayer $g  
}
```

Returns:

0x11f74b0

The address of the custom layer list is 0x11f74b0.

dbGeomPathLocArr

`dbGeomPathLocArr geomPathPtr`

Returns the address of the path location array specified by the geomPath

Parameters

<i>geomPathPtr</i>	Address of geometry path.
--------------------	---------------------------

Command Order

Use this command after invoking Innovus.

Example

```
dbGeomPathLocArr $geomBoxPtr
```

Returns:

0x836c9902

The address of the path location array is 0x836c9902.

dbGeomPathNrLoc

```
dbGeomPathNrLoc geomPathPtr
```

Returns the number of locations for the specified geomPath

Parameters

<i>geomPathPtr</i>	Address of geometry path.
--------------------	---------------------------

Command Order

Use this command after invoking Innovus.

Example

```
dbGeomPathNrLoc $geomBoxPtr
```

Returns:

8

There are 8 locations for the specified path.

dbGeomPathWidth

```
dbGeomPathWidth geomPathPtr
```

Returns the width of the specified geometry path.

Parameters

<i>geomPathPtr</i>	Address of geometry path.
--------------------	---------------------------

Command Order

Use this command after invoking Innovus.

dbGeomPolyLocArr

```
dbGeomPolyLocArr geomPolyPtr
```

Returns the address of the location array for the specified poly geometry.

Parameters

<i>geomPathPtr</i>	Address of geometry box.
--------------------	--------------------------

Command Order

Use this command after invoking Innovus.

Example

```
dbGeomPolyLocArr $geomBoxPtr
```

Returns:

0x83690cb4

The address for the poly location array is 0x83690cb4.

dbGeomPolyNrLoc

```
dbGeomPolyNrLoc geomPolyPtr
```

Returns the number of locations for the specified poly geometry.

Parameters

<i>geomPolyPtr</i>	Address of geometry box.
--------------------	--------------------------

Command Order

Use this command after invoking Innovus.

Example

```
dbGeomPolyNrLoc $geomBoxPtr
```

Returns:

5

The number of poly geometry locations is 5.

dbGeomTextHeight

`dbGeomTextHeight geomTextPtr`

Returns the height of the specified geometry text.

Parameters

<code><i>geomTextPtr</i></code>	Address of geometry text.
---------------------------------	---------------------------

Command Order

Use this command after invoking Innovus.

Example

`dbGeomTextHeight $geomBoxPtr`

Returns:

3.6

The text height is 3.6.

dbGeomTextLoc

`dbGeomTextLoc geomTextPtr`

Returns the location of the geometry text.

Parameters

<code><i>geomTextPtr</i></code>	Address of geometry text.
---------------------------------	---------------------------

Command Order

Use this command after invoking Innovus.

Example

```
dbGeomTextLoc $geomBoxPtr
```

Returns:

```
3000000 4000000
```

The location of the text is 3000.0 4000.0.

dbGeomTextText

```
dbGeomTextText geomTextPtr
```

Returns the address of the geometry text.

Parameters

<i>geomTextPtr</i>	Address of geometry text.
--------------------	---------------------------

Command Order

Use this command after invoking Innovus.

Example

```
dbGeomTextText $geomTextPtr
```

Returns:

```
0xb0396220
```

The address of the geometry text is 0xb0396220.

dbGetBlackBoxByName

```
dbGetBlackBoxByName {name | cellPtr name}
```

Returns the address of the blackbox.

Parameters

<i>name</i> <i>cellPtr name</i>	
	Name of blackbox or address and name of cell.

Command Order

Use this command after invoking Innovus.

Example

```
dbGetBlackBoxByName DMA
```

Returns:

```
0xb56a95d4
```

The address of the blackbox is 0xb56a95d4.

dbGetBottomIoPadOrient

```
dbGetBottomIoPadOrient
```

Returns the bottom I/O pad orientation.

Command Order

Use this command after invoking Innovus.

Example

dbGetBottomIoPadOrient

Returns:

R90

The bottom pad orientation is R90.

dbGetBumpByName

dbGetBumpByName {name | cellptr name}

Returns the address of the bump.

Parameters

<i>name</i> <i>cellptr name</i>	
	Name of bump or address and name of cell.

Command Order

Use this command after invoking Innovus.

Example

dbGetBumpByName VDDBMP45

Returns:

0x8d836c52

The address of the bump VDDBMP45 is 0x8d836c52.

dbGetBumpCellByName

dbGetBumpCellByName *name*

Returns the address of the bump cell.

Parameters

<i>name</i>	Name of bump.
-------------	---------------

Command Order

Use this command after invoking Innovus.

Example

dbGetBumpCellByName BUMP 34

Returns:

0xb9060320

The address of the bump BUMP 34 is 0xb9060320.

dbGetBumpCount

dbGetBumpCount

Returns the number of bumps, power nets, ground nets, and signals in the design, and whether the bump has been assigned.

Command Order

Use this command after starting the Innovus software.

Example

dbGetBumpCount

Returns:

0 0 0 0 0

The are no bumps, power nets, ground nets or signals.

dbGetBumpDetailCount

dbGetBumpDetailCount

Returns the number of bumps, assigned signal bumps, unassigned bumps, and the detailed power and ground bump information.

Command Order

Use this command after starting the Innovus software.

Example

dbGetBumpDetailCount

Returns:

0 0 0 {} {}

The are no bumps in the design.

dbGetCapUnitStr

dbGetCapUnitStr *floatingPtNumber*

Converts number to cap unit.

Parameters

floatingPtNumber

Floating point number.

Command Order

Use this command after invoking Innovus.

Example

```
dbGetCapUnitStr 3.0
```

Returns:

0.0003333

The cap unit is converted to 0.0003333.

dbGetCellByName

```
dbGetCellByName name
```

Returns the address of the specified cell.

Parameters

name

Cell master name.

Command Order

Use this command after importing the design.

Example

```
dbGetCellByName GRACC_DR
```

Returns:

0x83b568b0

The address of the cell GRACC_DR is 0x83b568b0.

dbGetCellByNameAndLib

dbGetCellByNameAndLib *name timeLibPtr*

Returns the address of the specified cell for the specified timing library.

Parameters

<i>name</i>	Cell master name.
<i>timeLibPtr</i>	Address of timing library.

Command Order

Use this command after importing the design.

Example

dbGetCellByNameAndLib GRACC_DR \$timeLibPtr

Returns:

0x83b568b

The address of the cell GRACC_DR is 0x83b568b.

dbGetCellByNameAndLibType

dbGetCellByNameAndLibType *name cond*

Returns the address of the specified cell for the specified timing condition.

Parameters

<i>name</i>	Cell master name.
<i>condition</i>	Can be either <code>min</code> or <code>max</code> .

Command Order

Use this command after importing the design.

Example

```
dbGetCellByNameAndLibType GRACC_DR min
```

Returns:

0x83b568b

The address of the cell `GRACC_DR` for the `min` operating condition is `0x83b568b`.

dbGetCellFreeLegalLoc

```
dbGetCellFreeLegalLoc cellPtr x_loc y_loc radius
```

Finds a legal location (X, Y and orientation) within the radius specified.

Parameters

<i>cell</i>	Address of cell.
<i>x_loc</i>	x location.
<i>y_loc</i>	y location.
<i>radius</i>	Radius.

Command Order

Use this command after design import.

Example

```
dbGetCellFreeLegalLoc $cellPtr 6300 57500 10000
```

Returns a location around 6.3 57.5 with a radius of 10.

dbGetCellTimeArc

```
dbGetCellTimeArc cellPtr relTermName tgtFTermName type:0~24 sense:0~4
```

Returns the address of the time arc.

Parameters

<i>cell</i>	Address of cell.
<i>relTermName</i>	From term
<i>tgtFTermName</i>	To term
<i>type:0~24</i>	Type: -1-undefined, 0-combinational, 1-output-to-output, 2-tristate enable, 3-tristate disable, 4-rising edge, 5-falling edge, 6-rising setup, 7-falling setup, 8-rising hold, 9-falling hold, 10-rising recovery, 11-falling recovery, 12-rising removal, 13-falling removal, 14-preset, 15-clear, 16-clock insertion, 17-generated clock, 18-edge up tri enable, 19-edge up tri disable, 20-edge down tri enable, 21-dege down tri disable, 22-retain, 23-min period, 24-MPW, 25-setup (inTran High, refClk rising, hold - inTran Low, refClk), 26-setup (inTran High, refClk falling, hold - inTran Low, refClk), 27 -(inTran Low, refClk rising, hold - inTran High, refClk), 28-(inTran Low, refClk falling, hold - inTran High refClk)
<i>sense:0~4</i>	Sense: -1-undefined, 0-nonunate, 1-posunate, 2-negunate, 3-Tlfposunate, 4-Tlfnegunate

Command Order

Use this command after design import.

Example

```
dbGetCellTimeArc GRACC_DR
```

Returns:

0x49e20a8

The address of the time arc is 0x49e20a8.

dbGetChildHInst

```
dbGetChildHInst hinstPtr
```

Returns the address of the list of child hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
set s [dbHeadSelList]
```

Returns:

0x96dc188

```
set objPtr [dbSelPtr $s]
```

Returns:

0x7f75340

```
dbGetChildHInst $objPtr
```

Returns:

0x7f753c8

The address of the list of child hierarchical instances is 0x7f753c8.

dbGetExtRuleByName

```
dbGetExtRuleByName name
```

Returns the address of the rule with the specified name.

Parameters

<i>name</i>	Name of rule.
-------------	---------------

Command Order

Use this command after design import.

Example

```
dbGetExtRuleByName RULE1
```

Returns:

0x38b4002c

The address of the rule named RULE1 is 0x38b4002c.

dbGetFPlanLayerHalo

dbGetFPlanLayerHalo *fplanPtr*

Returns the layer for halos for the specified floorplan.

Parameters

<i>fplanPtr</i>	Floorplan address.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

dbGetFPlanLayerHalo [dbHeadFPlan]

Returns:

0

The layer for halos is 0 for this design.

dbGetFPlanNrRow

dbGetFPlanNrRow *fplanPtr*

Returns the number of rows for the specified floorplan.

Parameters

<i>fplanPtr</i>	Name of rule.
-----------------	---------------

Command Order

Use this command after importing the design.

Example

```
dbGetFPlanNrRow [dbHeadFPlan]
```

Returns:

145

There are 145 rows for this design.

dbGetFTermByName

```
dbGetFTermByName name | cellPtr name
```

Returns the address.

Parameters

<i>name</i>	f-term name.
<i>cellPtr name</i>	Address of cell and f-term name.

Command Order

Use this command after design import.

Example

```
dbGetFTermByName BLU
```

Returns:

```
0x7c8f158
```

The address for the f-term BLU is 0x7c8f158.

dbGetGroupAncestor

```
dbGetGroupAncestor groupPtr
```

Returns the address of the ancestor for the specified group.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import and creating at least one group.

dbGetGroupByName

```
dbGetGroupByName name | fplanPtr name
```

Returns the address of group

Parameters

<i>name</i>	User specified name.
<i>fplanPtr name</i>	Floorplan address and name.

Command Order

Use this command after design import and creating at least one group.

dbGetHaloValue

`dbGetHaloValue instPtr`

Returns the halo value for the specified instance (one value for each side), 0 0 0 if there is none.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

`dbGetHaloValue $objPtr`

Returns:

0 0 0 0

The instance has no halo.

dbGetHInstBoxArea

dbGetHInstBoxArea *hinstPtr*

Returns the area of the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbGetHInstBoxArea $objPtr
```

Returns:

8367000000.0

The area is 836700000.0.

dbGetHInstByName

dbGetHInstByName *name*

Returns the address of the hierarchical instance by the specified name.

Parameters

<i>name</i>	Name of hierarchical instance.
-------------	--------------------------------

Command Order

Use this command after design import.

Example

```
dbGetHInstByName SH17
```

Returns:

```
0x7f75340
```

The address of the module SH17 is 0x7f75240.

dbGetHTermByInstTermName

```
dbGetHTermByInstTermName name | cellPtr name
```

Returns the address of the h-term specified by an instance terminal name.

Parameters

<i>name</i>	Name of terminal
<i>cellPtr name</i>	Address of cell and name.

Command Order

Use this command after design import.

dbGetInstByName

```
dbGetInstByName name | cellPtr name
```

Returns the address of an instance with the specified name.

Parameters

<i>name</i>	Instance name.
<i>cellPtr name</i>	Address of cell and name.

Command Order

Use this command after design import.

Example

```
dbGetInstByName SH19/I0/SH7/I119
```

Returns:

```
0x8b64ef0
```

The address of SH19/I0/SH7/I119 is 0x8b64ef0.

dbGetIoByName

```
dbGetIoByName name | cellPtr name
```

Returns the address of the I/O with the specified name.

Parameters

<i>name</i>	Name of cell.
<i>cellPtr name</i>	Address of cell and name.

Command Order

Use this command after design import.

Example

```
dbGetIoByName SH15/I194
```

Returns:

```
0x965488c4
```

The address of the I/O SH15/I194 is 0x965488c4.

dbGetIoDriverSiteCount

```
dbGetIoDriverSiteCount
```

Returns the number of I/O driver sites.

Command Order

Use this command after importing the design.

Example

```
dbGetIoDriverSiteCount
```

Returns:

```
3
```

There are three I/O driver sites.

dbGetIsNetlistChangedForClink

```
dbGetIsNetlistChangedForClink
```

Reports whether the netlist has been changed for clink. The command returns `1` if the netlist has been changed and `0` if has not.

Command Order

Use this command after importing the design.

Examples

```
dbGetNetlistChangedForClink
```

Returns:

`0`

The netlist has not been changed for clink.

dbGetLayerBlkByName

```
dbGetLayerBlkByName name | fplanPtr name
```

Returns the address of the layer blockage with the specified name.

Parameters

<i>name</i>	Name of routing blockage.
<i>fplanPtr name</i>	Address of floorplan and name.

Command Order

Use this command after design import and creation of at least one routing blockage.

Example

```
dbGetLayerBlkByName defLayerBlkName
```

Returns:

0x4a6ecbc

The address of the layer blockage with the name `defLayerBlkName` is 0x4a6ecbc.

dbGetLayerByExtId

```
dbGetLayerByExtId extID
```

Returns the address of the associated layer.

Parameters

<i>extID</i>	External ID.
--------------	--------------

Command Order

Use this command after design import.

Example

```
dbGetLayerByExtId 1
```

Returns:

0x48c6210

The address of the layer specified by the ID of 1 is 0x48c6210.

dbGetLayerByName

```
dbGetLayerByName name
```

Returns the address of the layer with the specified name.

Parameters

<i>name</i>	Name of layer.
-------------	----------------

Command Order

Use this command after design import.

Example

```
dbGetLayerByName METAL2
```

Returns:

0x48c75c0

The address of the layer METAL2 is 0x48c75c0.

dbGetLayerShapeLayer

```
dbGetLayerShapeLayer shapePtr
```

Returns the address of the layer for the specified shape.

Parameters

<i>shapePtr</i>	Name of shape
-----------------	---------------

Command Order

Use this command after design import and creation of at least one shape.

Example

```
dbGetLayerShapeLayer $shape
```

Returns:

0x48c75c0

The address of the layer is 0x48c75c0.

dbGetLDFLibs

```
dbGetLDFLibs
```

Gets all LEF files that are defined in the `ldf` file. The `ldf` file is used for running CIOP (chip planner).

Command Order

Use this command after design import.

dbGetLocAndSide

```
dbGetLocAndSide ptnPtr xcoord ycoord
```

Returns the location and partition side for the pin.

Parameters

<code>ptnPtr</code>	Address of partition
---------------------	----------------------

<i>xcoord</i>	x coordinate
<i>ycoord</i>	y coordinate

Command Order

Use this command after partitioning and pin assignment.

Example

```
dbGetLocAndSide [dbGetPtnByName sheet20] 2430.0 6780.0
```

Returns:

```
1785000 5669000 1
```

The pin at design location 2430 6780 for partition sheet20 is at 1785.0 5669.0 on the east side of this partition.

dbGetLocPowerDomainVoltage

```
dbGetLocPowerDomainVoltage xcoord ycoord
```

Returns the power domain for the specified location.

Parameters

<i>xcoord</i>	x coordinate
<i>ycoord</i>	y coordinate

Command Order

Use this command after floorplanning and specifying power domains.

Example

```
dbGetLocPowerDomain 2430.0 6780.0
```

Returns:

LowPwr

The power domain at location 2430 6780 is LowPwr.

dbGetMetalLayerById

```
dbGetMetalLayerById Id
```

Returns the address of metal layer.

Parameters

<i>Id</i>	ID.
-----------	-----

Command Order

Use this command after design import.

Example

```
dbGetMetalLayerById 3
```

Returns:

0x48d1028

The address of the metal layer with the ID of 3 is 0x48d1028.

dbGetNetByLocalName

`dbGetNetByLocalName name`

Returns the address of the net with the specified local name.

Parameters

<code>name</code>	Name of net.
-------------------	--------------

Command Order

Use this command after design import.

Example

`dbGetNetByLocalName BLU`

Returns:

`0x4f9e4d8`

The address of the net with the local name of `BLU` is `0x4f9e4d8`.

dbGetNetByName

`dbGetNetByName name | cellPtr name`

Returns the address of the net with the specified name.

Parameters

<code>name</code>	Name of net.
<code>cellPtr name</code>	Address of cell and name.

Command Order

Use this command after design import.

Example

```
dbGetNetByName BLU
```

Returns:

```
0x54bece8
```

The address of the net with the name of BLU is 0x54bece8.

dbGetNetDrivenTermsThroughLoc

```
dbGetNetDrivenTermsThroughLoc netPtr x_loc y_loc
```

Returns the address of terminals for the specified net at the specified location.

Parameters

<i>netPtr</i>	Address of net.
<i>x_loc</i>	x location
<i>y_loc</i>	y location

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbGetNetDrivenTermsThroughLoc $netPtr 673 902
```

Returns:

0x83b5cc88

The address of the net at location 0.673 0.902 is 0x83b5cc88.

dbGetNetFrequency

dbGetNetFrequency *netPtr*

Returns the frequency of the specified net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after RC extraction.

Example

dbGetNetFrequency [dbGetNetByName n_7874]

Returns:

256000

The frequency of net n_7874 is 256000.

dbGetNetGroupByName

dbGetNetGroupByName *name* | *netGroupPtr name*

Returns the address of the net group with the specified name.

Parameters

<i>name</i>	Name of net group.
<i>netGroupPtr name</i>	Address of net group and name.

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbGetNetGroupByName my_net_group
```

Returns:

0x9660650

The address of the group `my_net_group` is 0x9660650.

dbGetNetRectBBox

```
dbGetNetRectBBox netrectPtr
```

Returns the bounding box of the specified net rectangle.

Parameters

<i>netrectPtr</i>	Address of net rectangle
-------------------	--------------------------

Command Order

Use this command after RC extraction.

Example

```
dbGetNetRectBBox $netrectPtr
```

Returns:

```
283600 362902 3587200 503000
```

dbGetNetRectID

```
dbGetNetRectID netrectPtr
```

Returns the ID of the specified net rectangle.

Parameters

<i>netrectPtr</i>	Address of net rectangle.
-------------------	---------------------------

Command Order

Use this command after RC extraction.

Examples

```
dbGetNetRectID $netrectPtr
```

Returns:

```
2
```

dbGetNextHInst

```
dbGetNextHInst hinstPtr
```

Returns the address of next hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of the current hierarchical instance.
-----------------	---

Command Order

Use this command after design import.

Example

```
dbGetNextHInst [dbGetHInstByName SH17]
```

Returns:

0x8a0d258

The address of the next hierarchical instance is 0x8a0d258.

dbGetNumOfIoRow

```
dbGetNumOfIoRow
```

Returns the number of I/O rows.

Command Order

Use this command after importing the design.

Example

```
dbGetNumIoRow
```

Returns:

0x96d3fe0

1

There is one I/O row.

dbGetObjByName

dbGetObjByName *name*

Returns the address of the specified object.

Parameters

<i>name</i>	Name of object.
-------------	-----------------

Command Order

Use this command after importing the design.

Example

dbGetObjByName BLU

Returns:

0x96d3fe0

The address of the object named BLU is 0x96d3fe0.

dbGetObstructByName

dbGetObstructByName {*name* | *fplanPtr name*}

Returns the address of the specified obstruction. You can specify the obstruction by name, or specify the address and name of the floorplan.

Parameters

<i>name</i>	Name of placement obstruction.
<i>fplanPtr name</i>	Address of floorplan and name.

Command Order

Use this command after design import.

Example

```
dbGetObstructByName defObstructName
```

Returns:

0x96d3fe0

The address of the obstruction named `defObstructName` is 0x96d3fe0.

dbGetOrCreatePropByName

```
dbGetOrCreatePropByName objPtr propName
```

Returns the address of either the retrieved property or the newly created one.

Parameters

<i>objPtr</i>	Address of property
<i>propName</i>	Name of property.

Command Order

Use this command after design import and adding at least one property to one object.

Example

```
set wcPtr [ dbGetOrCreatePropByName $route WIRE_COUNT ]  
dbSetPropValue $wcPtr $routeCount
```

A new property, `WIRE_COUNT` is created.

dbGetOrCreateSNetByName

```
dbGetOrCreateSNetByName name | cellPtr name
```

Returns the address of the retrieved special net or the newly created special net.

Parameters

<code>name</code>	Name of special net.
<code>cellPtr name</code>	Address and name of cell.

Command Order

Use this command after design import.

Example

```
dbGetOrCreateSNetByName vdd
```

Returns:

`0x964d880`

The address of the special net `vdd` is `0x964d880`.

dbGetPathById

`dbGetPathById headPtr | id`

Returns the address of path specified by ID.

Parameters

<code>headPtr</code>	Address of head.
<code>id</code>	ID.

Command Order

Use this command after timing analysis.

dbGetPGFTermByName

`dbGetPGFTermByName name | cellPtr name`

Returns the address of the f-term specified by name or cell name and f-term.

Parameters

<code>name</code>	Name of f-term.
<code>cellPtr</code>	Address of cell.

Command Order

Use this command after design import.

Example

```
dbGetPGTermByName YFD1 VDD
```

Returns:

```
0x7f752b8
```

The VDD f-term address for cell YFD1 is 0x7f752b8.

dbGetPGTermByName

```
dbGetPGTermByName instPtr name
```

Returns the address of the term specified by the instance and name

Parameters

<i>name</i>	Name of term.
<i>instPtr</i>	Address of instance.

Command Order

Use this command after design import.

Example

```
dbGetPGTermByName [dbGetInstByName SH17/I340] VDD
```

Returns:

```
0x7f752b8
```

The VDD term address for instance SH17/I340 is 0x7f752b8.

dbGetPinGroupByName

dbGetPinGroupByName *fplanPtr name*

Returns the address of the pin group specified by name

Parameters

<i>fplanPtr</i>	Address of floorplan
<i>name</i>	Name of pin group.

Command Order

Use this command after creating a pin group.

Example

dbGetPinGroupByName [dbHeadFPlan] my_group

Returns:

0x7f752b8

The pin group my_group address is 0x7f752b8.

dbGetPowerDomainByName

dbGetPowerDomainByName

dbGetPowerDomainByName *name*

Returns the address of power domain with the specified name.

Parameters

<i>name</i>	Power domain name.
-------------	--------------------

Command Order

Use this command after importing the design.

Example

```
dbGetPowerDomianByName VDD1
```

Returns:

```
0x8b64ef0
```

The address of the power domain with the name `VDD1` is `0x8b64ef0`.

dbGetPrerouteAsObs

```
dbGetPrerouteAsObs
```

Returns layers (in bit map form) as obstructions by preroutes.

Command Order

Use this command after design import.

dbGetPrevHInst

```
dbGetPrevHInst hinstPtr
```

Returns the address of previous hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of current hierarchical instance.
-----------------	---

Command Order

Use this command after design import.

Example

```
dbGetPrevHInst [dbGetHInstByName SH17]
```

Returns:

0x7f752b8

The previous hierarchical instance address is 0x7f752b8.

dbGetPropByName

```
dbGetPropByName objPtr propName
```

Returns the address of property specified by name and object address.

Parameters

<i>objPtr</i>	Address of object.
<i>propName</i>	Name of property.

Command Order

Use this command after design import and registering at least one property.

Example

```
dbGetPropByName 0x74c500c0 WIRE_COUNT
```

Returns:

```
0x82c639c2
```

The address of the property on the object at 0x74c500c0 with the name WIRE_COUNT is 0x32c639c2.

dbGetPtnBorderCoords

```
dbGetPtnBorderCoords ptnPtr side xcoord ycoord
```

Returns the coordinate of the partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>side</i>	Side (N/S/W/E).
<i>xcoord</i>	x ordinate.
<i>ycoord</i>	y ordinate.

Command Order

Use this command after partitioning.

Example

```
dbGetPtnBorderCoords [dbGetPtnByName sheet7] W 3000 3000
```

Returns:

```
0 3000
```

The partition border coordinate for sheet7 is (0, 3000).

dbGetPtnByName

```
dbGetPtnByName name
```

Returns the address of the partition with the specified name.

Parameters

<i>name</i>	Name of partition.
-------------	--------------------

Command Order

Use this command after partitioning.

Example

```
dbGetPtnByName sheet7
```

Returns:

```
0x4dc2e30
```

The address for the partition `sheet7` is `0x4dc2e30`.

dbGetPtnCutByName

```
dbGetPtnCutByName name | fplanPtr name
```

Returns the address of the partition cut with the specified name.

Parameters

<i>name</i>	Name of partition cut.
<i>fplanPtr name</i>	Address of floorplan and name.

Command Order

Use this command after specifying partitions and creating at least one partition cut.

Example

```
dbGetPtnCutByName sheet17ptncut
```

Returns:

```
0xa08be28
```

The address for the partition cut with the name `sheet17ptncut` is `0xa08be28`.

dbGetPtnFeedByName

```
dbGetPtnFeedByName name | fplanPtr name
```

Returns the address of the partition feed specified by name.

Parameters

<i>name</i>	Name of partition feedthrough.
<i>fplanPtr name</i>	Address of floorplan and name.

Command Order

Use this command after specifying partitions and creating at least one feedthrough.

Example

```
dbGetPtnFeedByName defPtnFeedName
```

Returns:

```
0xc6d3e90
```

The address of the partition feed with the name of `defPtnFeedName` is `0xc6d3e90`.

dbGetPtnFPinsInArea

```
dbGetPtnFPinsInArea ptnPtr x1 y1 x2 y2
```

Returns pins in the specified area.

Parameters

<code>ptnPtr</code>	Address of partition.
<code>x1 y1</code>	Lower left xy coordinate.
<code>x2 y2</code>	Upper right xy coordinate.

Command Order

Use this command after partitioning.

Example

```
dbGetPtnFPinsInArea [dbGetPtnByName sheet7] 656 3270 964 3300
```

Returns the FPins within the area (0.656,3.27) (0.964,3.3).

dbGetPtnFPlanBox

`dbGetPtnFPlanBox ptnPtr`

Returns the box of the partition.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after specifying and committing partitions.

Example

`dbGetPtnFPlanBox $ptnPtr`

Returns:

5000 5000 10000 10000

Returns the box coordinates (5.0,5.0) (10.0,10.0).

dbGetPtnLayerHalo

`dbGetPtnLayerHalo ptnPtr`

Returns the layer for the partition halo.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after partitioning.

Example

```
dbGetPtnLayerHalo [dbGetPtnByName sheet7]
```

Returns:

0

Layer 0 is the layer used for the partition halo.

dbGetPtnPinBlkByName

```
dbGetPtnPinBlkByName name | fplanPtr name
```

Returns the address of the partition pin blockage with the specified name.

Parameters

<i>name</i>	Name of partition pin blockage.
<i>fplanPtr name</i>	Address of floorplan and name.

Command Order

Use this command after partitioning and creation of at least one partition pin blockage.

Example

```
dbGetPtnPinBlkByName defPtnPinBlkName
```

Returns:

0xc0084a8

The address of the partition pin blockage with the name `defPtnPinBlkName` is 0xc0084a8.

dbGetRawSlack

`dbGetRawSlack`

Returns the raw slack of the design. This command is similar to the `isTimingMet` command, but it shows an internal worst slack for the whole design.

Command Order

Use this command after timing analysis.

Example

`dbGetRawSlack`

Returns:

```
*** latch STA: #latches = 1592
*info: nrEmptyBatches = 1, nrNonEmptyBatches = 3 (0:00:00.0)
***latch STA: #iterations = 2
-1886
```

dbGetRouteBlkByName

`dbGetRouteBlkByName {name | fplanPtr name}`

Returns the address of the routing blockage with the specified name.

Parameters

<i>name</i>	Name of routing blockage.
<i>fplanPtr name</i>	Address of floorplan and name.

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
dbGetRouteBlkByName defLayerBlkName
```

Returns:

0x4a6ecbc

The address of the routing blockage with the name `defLayerBlkName` is 0x4a6ecbc.

dbGetRouteGuideByName

```
dbGetRouteGuideByName name | fplanPtr name
```

Returns the address of the route guide with the specified name.

Parameters

<i>name</i>	Name of route guide.
<i>fplanPtr name</i>	Address of floorplan and name.

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
dbGetRouteGuideByName defRouteGuide
```

Returns:

0xc6d3e90

The address of the route guide with the name `defRouteGuide` is 0xc6d3e90.

dbGetScreenByName

```
dbGetScreenByName name | fplanPtr name
```

Returns the address of the screen with the specified name.

Parameters

<code>name</code>	Name of density screen.
<code>fplanPtr name</code>	Address of floorplan and name.

Command Order

Use this command after importing a design and creating at least one density screen.

Example

```
dbGetScreenByName defScreenName
```

Returns:

0xb654ed0

The address of the screen with the name `defScreenName` is 0xb654ed0.

dbGetShifterCellsForPD

dbGetShifterCellsForPD

dbGetShifterCellsForPD *pdName*

Returns the address of the shifter cells for the specified power domain.

Parameters

<i>pdName</i>	Name of power domain.
---------------	-----------------------

Command Order

Use this command after importing the design.

Example

dbGetShifterCellsForPD vdd1

Returns:

0x964d880

The address of the cells for the power domain vdd1 is 0x964d880.

dbGetSNetByName

dbGetSNetByName *name* | *cellPtr name*

Returns the address of the special net with the specified name.

Parameters

<i>name</i>	Name of special net.
<i>cellPtr name</i>	Address of cell and name.

Command Order

Use this command after design import.

Example

```
dbGetSNetByName vdd
```

Returns:

0x964d880

The address of the special net `vdd` is 0x964d880.

dbGetSNodeByIdx

```
dbGetSNodeByIdx i j
```

Returns the address of SNode.

Parameters

<i>i</i>	
<i>j</i>	

Command Order

Use this command after global route.

dbGetSpefInFileName

```
dbGetSpefInFileName
```

Returns the name of the SPEF file that was input. If no SPEF file was input, this command returns nothing.

Command Order

Use this command after running the `spefIn` command.

Example

```
set spefInFileName [dbGetSpefInFileName]
```

dbGetSrcClks

```
dbGetSrcClks
```

Retrieves clocks from the constraint file.

Command Order

Use this command after design import including constraint file with at least one clock defined.

Example

```
dbGetSrcClks
```

The database is updated with the clocks from the constraint file.

dbGetStripByName

`dbGetStripByName name | fplanPtr name`

Returns the address strip by the specified name.

Parameters

<code>name</code>	Name of strip.
<code>fplanPtr name</code>	Address of floorplan and name.

Command Order

Use this command after design import and creation of at least one power/ground strap.

Example

`dbGetStripByName gnd`

Returns:

`0x964d9d8`

The address of the strip with the name `gnd` is `0x964d9d8`.

dbGetTechSiteByName

`dbGetTechSiteByName name`

Returns the address for the specified tech site.

Parameters

<i>name</i>	Name of tech site.
-------------	--------------------

Command Order

Use this command after importing the design.

Example

```
dbGetTechSiteByName tsm3site
```

Returns:

0x964d9d8

The address of the tech site with the name `tsm3site` is `0x964d9d8`.

dbGetTermByInstTermName

```
dbGetTermByInstTermName name
```

Returns the address of the specified instance terminal.

Parameters

<i>name</i>	Name of instance terminal.
-------------	----------------------------

Command Order

Use this command after design import.

Example

```
dbGetTermByInstTermName SH28/I2/I96/I179/A
```

Returns:

0x7d5e140

The address for the instance terminal SH28/I2/I96/I179/A is 0x7d5e140.

dbGetTermByName

dbGetTermByName *instPtr name*

Returns the address of the terminal specified by the instance and terminal name.

Parameters

<i>instPtr</i>	Address of instance.
<i>name</i>	Name of terminal.

Command Order

Use this command after design import.

Example

set s [dbHeadSelList]

Returns:

0x96dc194

set objPtr [dbSelPtr \$s]

Returns:

0x8086720

```
dbGetTermByName $objPtr A
```

Returns:

0x80e9db0

The address for the selected instance terminal A's terminal is 0x80e9db0.

dbGetTileCellByName

```
dbGetTileCellByName name
```

Returns the address of the tile cell with the specified name.

Parameters

<i>name</i>	Name of tile.
-------------	---------------

Command Order

Use this command after design import.

Example

```
dbGetTileCellByName core_3T
```

Returns:

0x7c55358

The address of the tile cell core_3T is 0x7c55358.

dbGetTimeLibByName

```
dbGetTimeLibByName name
```

Returns the address of the time library specified by name.

Parameters

<i>name</i>	Name of timing library.
-------------	-------------------------

Command Order

Use this command after design import.

Example

```
dbGetTimeLibByName t25lib_tt
```

Returns:

0x7c55358

The address for the time library `t25lib_tt` is `0x7c55358`.

dbGetTimeUnitStr

```
dbGetTimeUnitStr floatingPtNumber
```

Returns time units in picoseconds.

Parameters

<i>floatingPtNumber</i>	Floating point number.
-------------------------	------------------------

Command Order

Use this command after design import.

Example

```
dbGetTimeUnitStr 5.5
```

Returns:

100ps

5.5 is converted to 100ps.

dbGetViaCellByName

```
dbGetViaCellByName name
```

Returns the address of the via cell specified by name.

Parameters

<i>name</i>	Name of via.
-------------	--------------

Command Order

Use this command after design import.

Example

```
dbGetViaCellByName VIA3
```

Returns:

0x2c332c0

The address of the via cell VIA# is 0x2c332c0.

dbGetVInstByName

```
dbGetVInstByName {name | cellPtr name}
```

Returns the address of Verilog instance specified by name.

Parameters

<i>cellPtr</i>	Address of cell.
<i>name</i>	Name of the Verilog instance. Note: If only <i>name</i> is specified, the <i>cellPtr</i> is assumed by default to be the Vcell of the top cell.

Command Order

Use this command after design import.

dbGetVNetByName

```
dbGetVNetByName {name | cellPtr name}
```

Returns the address of the specified Verilog net.

Parameters

<i>cellPtr</i>	Address of cell.
<i>name</i>	Name of the Verilog net. Note: If only <i>name</i> is specified, the <i>cellPtr</i> is assumed by default to be the Vcell of the top cell.

Command Order

Use this command after importing the design.

dbGlobalNetConnectionConnectType

```
dbGlobalNetConnectionConnectType globalNetConnectionPtr
```

Returns the type of the global connection, 0 for pin, 1 for net, 2 for tiehi, 3 for tielo.

Parameters

globalNetConnectionPtr
Address of group.

Command Order

Use this command after importing the design and creating at least one global connection.

Example

```
dbForEachFPlanGlobalNetConnection [dbHeadFPlan] glbnetPtr {  
    set type [dbGlobalNetConnectionConnectType $glbnetPtr]  
    puts "type = $type"  
}
```

Returns:

```
type = 0  
type = 0  
type = 0  
type = 0
```

```
type = 2
type = 3
```

There are four global net connections with a connect type of 0/pin, one with a connect type of 2/tiehi, and one with a connect type of 3/tielo.

dbGlobalNetConnectionToGlobalNet

```
dbGlobalNetConnectionToGlobalNet globalNetConnectionPtr
```

Returns the name of the net associated with the specified global net address.

Parameters

<i>globalNetConnectionPtr</i>
Address of group.

Command Order

Use this command after importing the design and creating at least one group.

Example

```
dbForEachFPlanGlobalNetConnection [dbHeadFPlan] glbnetPtr {
    set net [dbGlobalNetConnectionToGlobalNet $glbnetPtr]
    puts "net = $net"
}
```

Returns:

```
net = VDD
net = VSS
net = VDD
```

```
net = VSS  
net = VDD  
net = VSS
```

The global net names for the global net connections are VDD and VSS.

dbGlobalNetConnectionUnderModule

```
dbGlobalNetConnectionUnderModule globalNetConnectionPtr
```

Returns the module name for scope type of 1 - see dbGlobalNetConnectionScopeType.

Parameters

globalNetConnectionPtr
Address of group.

Command Order

Use this command after importing the design and creating at least one global net connection.

Example

```
dbForEachFPlanGlobalNetConnection [dbHeadFPlan] glbnetPtr {  
    set mod [dbGlobalNetConnectionUnderModule $glbnetPtr]  
    puts "mod = $mod"  
}
```

Returns:

```
mod = 0  
mod = 0
```

```
mod = 0
mod = 0
mod = 0
mod = 0
```

There are no module names for any of the global net connections in this design.

dbGlobalNetConnectionVerbose

```
dbGlobalNetConnectionVerbose globalNetConnectionPtr
```

Returns a verbose output for the specified global net connection.

Parameters

<i>globalNetConnectionPtr</i>
Address of global net connection.

Command Order

Use this command after importing the design and creating at least one global net connection.

Example

```
dbForEachFPlanGlobalNetConnection [dbHeadFPlan] glbnetPtr {
    set veb[dbGlobalNetConnectionVerbose $glbnetPtr]
    puts $vreb
}
```

Returns:

```
0
0
```

0
0
0
0

There are no global net connections in this design.

dbGroundSpef

`dbGroundSpef inputSpefName outputSpefName`

Takes the input SPEF, grounds it, and writes it to the specified output SPEF file.

Parameters

<i>inputSpefName</i>	Name of input SPEF file.
<i>outputSpefName</i>	Name of output SPEF file.

Command Order

Use this command after extraction.

Example

`dbGroundSpef TOPCHIP_SP.spef output.spef`

Returns:

Start to grounding spef file TOPCHIP_SP.spef.

Grounding spef file completed. Grounded spef file is output.spef.

dbGroupBox

dbGroupBox *groupPtr*

Returns the box specified by the group.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after importing the design and creating at least one group.

Example

```
dbGroupBox [dbGetGroupByName my_group]
```

Returns:

2606300 4021300 2634137 4051365

The box for the group, my_group, is (2606.3, 4021.3) (2634.137, 4051.365).

dbGroupConType

dbGroupConType *groupPtr*

Returns the constraint type. Possible values are:

dbcGuide

dbcFence

dbcRegion

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import.

Example

```
dbGroupConType [dbGetGroupByName my_group]
```

Returns:

```
dbcGuide
```

The constraint for `my_group` is a guide.

dbGroupDensity

```
dbGroupDensity groupPtr
```

Returns the density.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import.

Example

```
dbGroupDensity [dbGetGroupByName my_group]
```

Returns:

1.0

The density for my_group is 1.0.

dbGroupGroup

dbGroupGroup *groupPtr*

Returns the address of group.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after importing the design and creating at least one group.

dbGroupHInstList

dbGroupHInstList *groupPtr*

Returns the address of the list of hierarchical instances in the specified group.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import, creation of group, and adding instances to group.

Example

```
dbGroupHInstList [dbGetGroupByName my_group]
```

Returns:

0xc009480

The address of the hierarchical instance list for the group, `my_group`, is 0xc009480.

dbGroupName

```
dbGroupName groupPtr
```

Returns the name specified by the group address.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import and creation of at least one group.

Example

```
dbGroupName $groupPtr
```

Returns:

my_group

The name of the group is my_group.

dbGroupNext

dbGroupNext *groupPtr*

Returns the address of the next group, 0x0 if there is none.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import and creation of at least one group.

Example

```
dbGroupNext [dbGetGroupByName my_group]
```

Returns:

0x0

There is not a next group.

dbGroupNrHInst

dbGroupNrHInst *groupPtr*

Returns the number of hierarchical instances in the specified group.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import, creation of a group, and adding an instance to the group.

Example

```
dbGroupNrHInst [dbGetGroupByName my_group]
```

Returns:

2

The number of hierarchical instances in `my_group` is 2.

dbGroupPowerDomain

```
dbGroupPowerDomain groupPtr
```

Returns the power domain for the specified group.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after creating groups and defining power domains.

H

dbHasBlackBoxAreaValueBeenSet

dbHasBlackBoxAreaValueBeenSet *blackBoxPtr*

Returns a 1 if the blackbox area has been set, else a 0.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after design import.

Example

dbHasBlackBoxAreaValueBeenSet \$blackBoxPtr

Returns:

1.0

The blackbox area has been set.

dbHasBlackBoxXYBeenSet

dbHasBlackBoxXYBeenSet *blackBoxPtr*

Returns a 1 if the XY has been set, else a 0.

Parameters

<i>blackBoxPtr</i>	Address of blackbox.
--------------------	----------------------

Command Order

Use this command after design import.

Example

```
dbHasBlackBoxXYBeenSet [dbgHead]
```

Returns:

1

The blackbox XY has been set.

dbHeadAspectRatio

```
dbHeadAspectRatio headPtr
```

Returns the aspect ratio.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadAspectRatio [dbgHead]
```

Returns:

1.0

The aspect ratio of the design is 1.0.

dbHeadBlockAsStdCell

dbHeadBlockAsStdCell *headPtr*

Returns a 1 if block has standard cells.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadBlockAsStdCell [dbgHead]

Returns:

1

The block has standard cells.

dbHeadBlockMargin

dbHeadBlockMargin *headPtr*

Returns a 1 if there is a margin, else a 0.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadBlockMargin [dbgHead]
```

Returns:

0

There is no margin for this design.

dbHeadBox

```
dbHeadBox headPtr
```

Returns the design area.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadBox [dbgHead]
```

Returns:

```
0 0 7237450 7232600
```

The design is area is (0, 0) (7237.45, 7232.6).

dbHeadCellList

```
dbHeadCellList headPtr
```

Address of list of cells in the design.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadCellList [dbgHead]
```

Returns:

```
0x72b0de0
```

The address of the list of all the cells in the design is 0x72b0de0.

dbHeadChipTopCell

dbHeadChipTopCell *headPtr*

Returns the address of the top cell for the design.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadChipTopCell [dbgHead]

Returns:

0x72b0de0

The address of the top cell is 0x72b0de0.

dbHeadCoreBox

dbHeadCoreBox *headPtr*

Returns box of core area.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadCoreBox [dbgHead]
```

Returns:

```
430300 430300 6802300 6802300
```

The core box coordinates are (430.3, 430.3) (6802.3, 6802.3).

dbHeadCustomLayerList

```
dbHeadCustomLayerList headPtr
```

Returns the address of custom layer list

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadCustomLayerList [dbgHead]
```

Returns:

```
0x89033672
```

The address for the custom layers is 0x89033672.

dbHeadDBUPerIGU

dbHeadDBUPerIGU *headPtr*

Returns database units per instance grid.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadDBUPerIGU [dbgHead]

Returns:

1000

The database units per instance grid is 1000.

dbHeadDBUPerIoGU

dbHeadDBUPerIoGU *headPtr*

Returns database units per I/O grid.

Parameters

headPtr

Address of head.

Command Order

Use this command after design import.

Example

```
dbHeadDBUPerIoGU [dbgHead]
```

Returns:

0

The database units per I/O grid is 0.

dbHeadDBUPerMGrid

```
dbHeadDBUPerMGrid headPtr
```

Returns the database units per manufacturing grid.

Parameters

headPtr

Address of head.

Command Order

Use this command after design import.

Example

```
dbHeadDBUPerMGrid [dbgHead]
```

Returns:

1

The database units per manufacturing grid is 1.

dbHeadExtRuleArr

dbHeadExtRuleArr *headPtr*

Returns the address of the extension rules, 0x0 if there none.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadExtRuleArr [dbgHead]

Returns:

0x4ad3320

The address for the external rule array is 0x4ad3320.

dbHeadExtRuleList

dbHeadExtRuleList *headPtr*

Returns the address of the list of extension rules, 0x0 if there are none.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadExtRuleList [dbgHead]
```

Returns:

0x4a83908

The address for the external rule list is 0x4a83908.

dbHeadFELayerList

```
dbHeadFELayerList headPtr
```

Returns the address for the list of layers.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadFELayerList [dbgHead]
```

Returns:

0x48c62b0

The address for the layer list is 0x48c62b0.

dbHeadFirstSelPtr

```
dbHeadFirstSelPtr headPtr
```

Returns the address if first selected object.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import and at least one object is selected.

Example

```
dbHeadFirstSelPtr [dbgHead]
```

Returns:

0x79360c8

The address of select pointer is 0x79360c8.

dbHeadFPlan

dbHeadFPlan *headPtr*

Returns the address for the floorplan.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadFPlan [dbgHead]

Returns:

0x9651f78

The address of the floorplan is 0x9651f78.

dbHeadHiliteList

dbHeadHiliteList *headPtr*

Returns the address of list of highlighted objects, or 0x0 if there are no highlighted objects.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadHiliteList [dbgHead]
```

Returns:

0x0

There are no highlighted objects in the design.

dbHeadIoBox

```
dbHeadIoBox headPtr
```

Returns I/O box.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadIoBox [dbgHead]
```

Returns:

330300 330300 6902300 6902300

The coordinates of the I/O box are (330.3,330.3) (6902.3,6902.3).

dbHeadIoHgt

dbHeadIoHgt *headPtr*

Returns I/O height.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadIoHgt [dbgHead]

Returns:

335150

The I/O height is 335.14.

dbHeadLEFLayerArr

dbHeadLEFLayerArr *headPtr*

Returns the address of the LEF layers.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadLEFLayerArr [dbgHead]
```

Returns:

0x4a7a0c8

The LEF layer array address is 0x4a7a0c8.

dbHeadLEFLayerList

```
dbHeadLEFLayerList headPtr
```

Returns the address of list of LEF layers.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadLEFLayerList [dbgHead]
```

Returns:

0x48c62b0

The address of the LEF layer list is 0x48c62b0.

dbHeadMicronPerDBU

dbHeadMicronPerDBU *headPtr*

Returns microns per database unit.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadMicronPerDBU [dbgHead]

Returns:

0.001

The microns per database unit is 0.001.

dbHeadName

dbHeadName *headPtr*

Returns the design name.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadName [dbgHead]
```

Returns:

```
trainp
```

The name of the design is `trainp`.

dbHeadNrExtRule

```
dbHeadNrExtRule headPtr
```

Returns the number of extended rules, 0x0 if there are none.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadNrExtRule [dbgHead]
```

Returns:

1

The number of external rules is 1.

dbHeadNrExtRulePlus1

```
dbHeadNrExtRulePlus1 headPtr
```

Used for displaying rules using the old db object displayer.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadNrExtRulePlus1 [dbgHead]
```

Returns:

1

dbHeadNrHilite

dbHeadNrHilite *headPtr*

Returns the number of highlighted objects.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadNrHilite [dbgHead]

Returns:

0

The number of highlighted objects is 0.

dbHeadNrLayer

dbHeadNrLayer *headPtr*

Returns the number of layers.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadNrLayer [dbgHead]
```

Returns:

8

The number of layers is 8.

dbHeadNrLEFLayer

```
dbHeadNrLEFLayer headPtr
```

Returns the number of LEF layers.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadNrLEFLayer [dbgHead]
```

Returns:

5

The number of LEF layers is 5.

dbHeadNrSel

dbHeadNrSel *headPtr*

Returns the number of selected objects.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import, but makes more sense if some objects are selected.

Example

dbHeadNrSel [dbgHead]

Returns:

1

The number of selected objects is 1.

dbHeadNrWireLayer

dbHeadNrWireLayer *headPtr*

Returns the number of layers for wires.

Parameters

headPtr

Address of head.

Command Order

Use this command after design import.

Example

```
dbHeadNrWireLayer [dbgHead]
```

Returns:

4

The number of wire layers is 4.

dbHeadOhmPerDBU

```
dbHeadOhmPerDBU headPtr
```

Returns ohms per database unit.

Parameters

headPtr

Address of head.

Command Order

Use this command after design import.

Example

```
dbHeadOhmPerDBU [dbgHead]
```

Returns:

0.1

The number of ohms per database unit is 0.1.

dbHeadPathList

dbHeadPathList *headPtr*

Returns the address of the list of paths.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

dbHeadPicoFPerDBU

dbHeadPicoFPerDBU *headPtr*

Returns pico farads per database unit.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadPicoFPerDBU [dbgHead]
```

Returns:

0.001

The number of pico farads per database unit is 0.001.

dbHeadPicoSecPerDBU

```
dbHeadPicoSecPerDBU headPtr
```

Returns pico seconds per database unit.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadPicoSecPerDBU [dbgHead]
```

Returns:

0.1

The number of pico seconds per database unit is 0.1.

dbHeadPropTypeList

dbHeadPropTypeList *headPtr*

Returns the address of the types of properties.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadPropTypeList [dbgHead]

Returns:

0x47cce50

The address of the property types is 0x47cce50.

dbHeadPtnList

dbHeadPtnList *headPtr*

Returns the address of list of partitions.

Parameters

headPtr

Address of head.

Command Order

Use this command after design import.

Example

```
dbHeadPtnList [dbgHead]
```

Returns:

0x644a8f0

The address of the list of partitions is 0x644a8f0.

dbHeadRule

```
dbHeadRule headPtr
```

Returns the address of the head rule.

Parameters

headPtr

Address of head.

Command Order

Use this command after design import.

Example

```
dbHeadRule [dbgHead]
```

Returns:

0x3d6d1a0

The address of the rules is 0x3d6d1a0.

dbHeadSelList

dbHeadSelList *headPtr*

Returns the address of the list of selected objects.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadSelList [dbgHead]

Returns:

0x96e09e0

The address of the selected list is 0x96e09e0.

dbHeadSpDetailLen

dbHeadSpDetailLen *headPtr*

Returns the horizontal and vertical length for the special nets.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after creating at least one P/G.

Example

```
dbHeadSpDetailLen [dbgHead]
```

Returns:

8206788.302000 7982043.755000

The horizontal length for the special nets is 8206788.302; the vertical length is 7982043.755.

dbHeadSpGlobalLen

```
dbHeadSpGlobalLen headPtr
```

Returns the horizontal and vertical global net length for special routes.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadSpGlobalLen [dbgHead]
```

Returns:

```
0.000000 0.000000
```

There is no global length for the special nets.

dbHeadStdCellHgt

```
dbHeadStdCellHgt headPtr
```

Returns the standard cell height.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbHeadStdCellHgt [dbgHead]
```

Returns:

```
27000
```

The standard cell height is 27.0.

dbHeadTopCell

dbHeadTopCell *headPtr*

Returns the address of the top cell.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbHeadTopCell [dbgHead]

Returns:

0x72b0de0

The address of the top cell is 0x72b0de0.

dbHiliteObj

dbHiliteObj *objPtr* <0...31>

Highlights the specified object.

Parameters

<i>hinstPtr</i>	Address of object.
0...31	Highlight set. 0 through 15 for internal selection, 16 through 31 for browser.

Command Order

Use this command after importing the design.

Example

```
dbHiliteObj $objPtr 3
```

The object is highlighted.

dbHiliteObjBox

```
dbHiliteObjBox objPtr
```

Highlights the box of the specified object.

Parameters

<i>objPtr</i>	Address of object.
---------------	--------------------

Command Order

Use this command after design import

Example

```
dbHiliteObj $objPtr
```

The object's box is highlighted.

dbHInstArea

```
dbHInstArea hinstPtr
```

Returns the area of the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

You must call `dbHInstResetArea` before using this command.

Example

```
dbHInstArea $objPtr
```

Returns:

```
16018040077500.0
```

The area of the hierarchical instance is 16018040077500.0.

dbHInstBlockAreaWHalo

```
dbHInstBlockAreaWHalo hinstPtr
```

Returns the area including the halo of the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after importing the design.

Example

```
dbHInstBlockAreaWHalo [dbGetHInstByName DTMF_INST]
```

Returns:

1465386248100.0

The area of the hierarchical instance is 1465386248100.0.

dbHInstBox

```
dbHInstBox hinstPtr
```

Returns the coordinates of the box; returns 2147483647 2147483647 -2147483648 -2147483648 if there is none.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstBox $objPtr
```

Returns:

2147483647 2147483647 -2147483648 -2147483648

There is no box for the specified hierarchical instance.

dbHInstCell

dbHInstCell *hinstPtr*

Returns the address of the cell associated with the hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

dbHInstCell \$objPtr

Returns:

0x606e330

The address of the cell for the specified hierarchical instance is 0x606e330.

dbHInstCellName

dbHInstCellName *hinstPtr*

Returns the name of the hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstCellName $objPtr
```

Returns:

```
hn1_35
```

The name of the hierarchical instance is hn1_35.

dbHInstColorId

```
dbHInstColorId hinstPtr
```

Returns the color ID for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstColorId $objPtr
```

Returns:

0

The color ID for the hierarchical instance is 0.

dbHInstCongest

```
dbHInstCongest hinstPtr
```

Returns the congestion for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after route.

Example

```
dbHInstCongest $objPtr
```

Returns:

0.0

The congestion is 0.0.

dbHInstConnection

dbHInstConnection *hinstPtr*

Note: This command currently does not work.

dbHInstConstraint

dbHInstConstraint *hinstPtr*

Returns the address of the constraint for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

dbHInstConstraint \$objPtr

Returns:

0x9663ec8

The address of the constraint for the specified hierarchical instance is 0x9663ec8.

dbHInstDensity

dbHInstDensity *hinstPtr*

Returns the density of the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after place.

Example

```
dbHInstDensity $objPtr
```

Returns:

0.0

The density is 0.0.

dbHInstFenceDensity

```
dbHInstFenceDensity hinstPtr
```

Returns the fence density of the hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import and making at least one fence.

Example

```
dbHInstFenceDensity $objPtr
```

Returns:

0.0

The fence density for the specified hierarchical instance is 0.0.

dbHInstFPlanBox

```
dbHInstFPlanBox hinstPtr
```

Returns floorplan box of the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstFPlanBox $objPtr
```

Returns:

430300 4048300 6802300 6802300

The coordinates for the floorplan box for the specified hierarchical instance are (430.3, 430.03) (6802.3, 6802.3).

dbHInstGroup

dbHInstGroup *hinstPtr*

Returns the address of the groups within the hierarchical instance, 0x0 if there are none.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

dbHInstGroup \$objPtr

Returns:

0x0

There are no groups in this design.

dbHInstHInstList

dbHInstHInstList *hinstPtr*

Address of the list of hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstHInstList $objPtr
```

Returns:

0x7fa9700

The address of the list of hierarchical instances is 0x7fa9700.

dbHInstHorOrder

```
dbHInstHorOrder hinstPtr
```

Returns the horizontal order for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstHorOrder $objPtr
```

Returns:

0

The horizontal order is 0.

dbHInstHTermList

dbHInstHTermList *hinstPtr*

Returns the address of the list of h-terms for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

dbHInstHTermList \$objPtr

Returns:

0x7fc7bb8

The address of the list of h-terms is 0x7fc7bb8.

dbHInstId

dbHInstId *hinstPtr*

Returns the ID for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstId $objPtr
```

Returns:

0

The ID for the specified hierarchical instance is 0.

dbHInstIIm

```
dbHInstIIm hinstPtr
```

Returns the ILM address for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after importing the design.

Example

```
dbHInstILm $objPtr
```

Returns:

0x8962b034

The ILM address for the specified hierarchical instance is 0x8962b034.

dbHInstName

```
dbHInstName hinstPtr
```

Returns the name for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstName $objPtr
```

Returns:

SH17

The name of the hierarchical instance is SH17.

dbHInstNrInst

`dbHInstNrInst hinstPtr`

Returns the number of instances (not including I/O pads) for the specified hierarchical instance.

Parameters

<code>hinstPtr</code>	Address of hierarchical instance.
-----------------------	-----------------------------------

Command Order

Use this command after importing the design.

Example

`dbHInstNrInst $objPtr`

Returns:

27551

The number of instances in the hierarchical instance is 27,551.

dbHInstParent

`dbHInstParent hinstPtr`

Returns the address of the parent for the specified hierarchical instance.

Parameters

<code>hinstPtr</code>	Address of hierarchical instance.
-----------------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstParent [dbGetHInstByName SH17]
```

Returns:

0x61cfcc18

The address of the parent for SH17 is 0x61cfcc18.

dbHInstPCell

```
dbHInstPCell hinstPtr
```

Returns the address of the physical cell for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstPCell $objPtr
```

Returns:

0x72b0de0

The address of the physical cell for the specified hierarchical instance is 0x72b0de0.

dbHInstPerimList

dbHInstPerimList *hinstPtr*

Returns the address of the perimeter list for the specified hierarchical instance, 0x0 if there is none.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstPerimList [dbGetHInstByName SH17]
```

Returns:

0x0

There is no perimeter list for the specified hierarchical instance.

dbHInstPin

dbHInstPin *hinstPtr side order*

Returns the list of pins for the specified hierarchical instance for the specified side.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
<i>side</i>	Side (N/S/E/W/)
<i>order</i>	Order.

Command Order

Use this command after design import.

dbHInstPrefixId

dbHInstPrefixId *hinstPtr*

Returns the prefix ID for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after importing the design.

Example

dbHInstPrefixId \$objPtr

Returns:

335

The prefix ID for the specified hierarchical instance is 335.

dbHInstPropList

dbHInstPropList *hinstPtr*

Returns the address of the property list for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

dbHInstPropList \$objPtr

Returns:

0x5adeee0

The address of the property list for the specified hierarchical instance is 0x5adeee0.

dbHInstPtListBox

dbHInstPtListBox *hinstPtr*

Returns point list box for the specified hierarchical instance; returns 2147483647 2147483647 – 2147483648 –2147483648 if there is not one.

Parameters

hinstPtr

Address of hierarchical instance.

Command Order

Use this command after design import.

Example

```
dbHInstPtListBox $objPtr
```

Returns:

```
2147483647 2147483647 -2147483648 -2147483648
```

dbHInstPtn

```
dbHInstPtn hinstPtr
```

Returns the address of partition for the specified hierarchical instance.

Parameters

hinstPtr

Address of hierarchical instance.

Command Order

Use this command after specifying partitions.

Example

```
dbHInstPtn $objPtr
```

Returns:

```
0x644a8f0
```

The address of the partition for the specified hierarchical instance is 0x644a8f0.

dbHInstResetArea

dbHInstResetArea *hinstPtr*

Resets the area for the specified hierarchical instance

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

dbHInstStdCellArea

dbHInstStdCellArea *hinstPtr*

Returns the standard cell area for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

You must call `dbHInstResetArea` before using this command.

Example

```
dbHInstStdCellArea $objPtr
```

Returns:

```
16018040077500.0
```

The standard cell area for the specified hierarchical instance is 160180400775500.0.

dbHInstSuggestLoc

```
dbHInstSuggestLoc hinstPtr
```

Returns the suggested location for the specified hierarchical instance; returns -2147483648 -2147483648 if there is none.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance
-----------------	----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstSuggestLoc $objPtr
```

Returns:

```
-2147483648 -2147483648
```

There is no suggested location.

dbHInstSuggestOrient

dbHInstSuggestOrient *hinstPtr*

Returns the suggested orientation for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstSuggestOrient $objPtr
```

Returns:

dbcR0

The orientation for the specified hierarchical instance is R0.

dbHInstType

dbHInstType *hinstPtr*

Returns the type for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstType $objPtr
```

Returns:

```
dbcHInstHierarchy
```

The type of hierarchical instance is `InstHierarchy`.

dbHInstUserDensity

```
dbHInstUserDensity hinstPtr
```

Returns user density for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbHInstUserDensity $objPtr
```

Returns:

0.0

The density is 0.0 for the specified hierarchical instance.

dbHInstVerOrder

`dbHInstVerOrder hinstPtr`

Returns vertical order for the specified hierarchical instance.

Parameters

<code>hinstPtr</code>	Address of hierarchical instance
-----------------------	----------------------------------

Command Order

Use this command after design import.

Example

`dbHInstVerOrder $objPtr`

Returns:

0

The vertical order is 0.

dbHTermFTerm

`dbHTermFTerm htermPtr hinstPtr`

Returns the address of the f-term.

Parameters

<i>htermPtr</i>	Address of h-term.
<i>hinstPtr</i>	Address of hierarchical instance.

Command Order

Use this command after design import.

dbHTermHInst

`dbHTermHInst htermPtr`

Returns the name of the module for the specified h-term.

Parameters

<i>htermPtr</i>	Address of h-term.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachHInstHTerm $objPtr hinstPtr {  
    puts "Module name = [dbHInstName $hinstPtr]"  
}
```

Returns:

```
Mmodule name = DMA  
Module name = ADDER1  
Module name = ADDER2
```

dbHTermNet

dbHTermNet *htermPtr*

Returns the address of net for the specified h-term.

Parameters

<i>htermPtr</i>	Address of h-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbForEachHInstHTerm $objPtr dpinstPtr {  
    set net [dbHTermNet $dpinstPtr]  
    set net_name [dbNetName $net]  
    puts "net name = $net_name"  
}
```

```
net name = WACKS  
net name = WACKZ  
net name = WBUFULLN
```

The names of the nets attached to the h-terms are WACKS, WACKZ and WBUFULLN.

dbHTermNext

dbHTermNext *htermPtr*

Returns the address of next h-term.

Parameters

<i>htermPtr</i>	Address of h-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

dbHTermNext \$htermPtr

Returns:

0x96e09e0

The address of the next h-term is 0x96e09e0.

dbHTermOrder

dbHTermOrder *htermPtr*

Returns the order for the specified h-term.

Parameters

<i>htermPtr</i>	Address of h-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbForEachHInstHTerm $objPtr dpinstPtr {  
    set order [dbHTermOrder $dpinstPtr]  
    puts "order = $order"  
}
```

Returns:

```
order = 0  
order = 0  
order = 0
```

The order of each h-term is 0.

dbHTermSide

```
dbHTermSide htermPtr
```

Returns the side for the specified h-term. Possible values are:

```
dbcN  
dbcS  
dbcW  
dbcE
```

Parameters

<i>htermPtr</i>	Address of h-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbForEachHInstHTerm $objPtr dpinstPtr {  
    set side [dbHTermSide $dpinstPtr]  
    puts "side = $side"  
}
```

Returns:

```
side = dbcN  
side = dbcN  
side = dbcN
```

All of the h-terms are on the north side.

Database Commands I

- I

dbIncrDelayUpdate

```
dbIncrDelayUpdate netPtr
```

Incrementally updates the specified net delay. For example, you resize a cell, or touch a net, and want only the fanout cone of this net or cell to be updated in terms of delay.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after building timing graph.

Example

```
dbIncrDelayUpdate [dbGetNetByName SH13/IPB<13>]
```

dbInfoStripBox

```
dbInfoStripBox stripbox
```

Returns information about the strip box.

Parameters

<i>stripbox</i>	Address of stripbox
-----------------	---------------------

Command Order

Use this command after creating at least one strip box.

Example

```
dbForEachFPlanStrip [dbCellFPlan [dbgTopCell]] stpPtr {  
    set info [dbInfoStripBox $stpPtr]  
    puts $info  
}
```

dbInfoVia

```
dbInfoVia via
```

Returns information about the via specified such as the net, the via cell, the coordinate and the shape (for shape information see dbInfoWire).

Parameters

via	Address of via
-----	----------------

Command Order

Use this command after routing.

Example

```
set vc [dbCreateVia BLU $viaPtr 5000 5000]  
0xb188670  
dbInfoVia $vc
```

The first command creates an instantiation of a via cell, the second reports the information.

dbInfoViaCellLayer

```
dbInfoViaCellLayer viacell
```

Returns information like upper metal layer, lower layer metal and cut layer rectangles.

Parameters

viacell	Address of via cell
---------	---------------------

Command Order

Use this command after routing.

Example

```
set vc [dbCreateRestrictViaCell 2 513000 513000 514000 514000 1000 1000 1000 1000 1 1 1  
1]
```

0xb188670

```
dbInfoViaCellLayer $vc  
1000.0 1000.0 1000.0 1000.0 1 1 1.0 1.0
```

The first command creates a restricted via, while the second reports the via cell layer information.

dbInfoViaCellRegularCuts

```
dbInfoViaCellRegularCuts cellPtr
```

Returns the following list about regular cut vias:

```
cut_lx cut_ly cut_width cut_height xtimes ytimes xpitch ypitch
```

Note that *xtimes * ytimes* could be 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after routing.

Example

```
set cut_geom [dbInfoViaCellRegularCuts $vc]
```

Stores the via information in the variable *cut_geom*.

dbInfoWire

```
dbInfoWire wire
```

Returns information for the specified wire, such as the net name, the wire rectangle, the layer, the direction and the shape. Wires can have one of the following shapes:

Shape	Definition
NOTYPE	Anything not having a specific shape
RING	Used as core rings
STRIPE	Trunks not classified as a RING or BLOCKRING
FOLLOWPIN	Connects the standard cell rails
IOWIRE	Connects I/O to the core
COREWIRE	Connects followpin wires to trunks (usually RING)
BLOCKWIRE	Connects blocks to trunks
FILLWIRE	Connects metal fills
PADRING	Connects rings in the peripheral I/O area
BLOCKRING	Connects rings around a block or blocks

Parameters

<i>wire</i>	Address of wire.
-------------	------------------

Command Order

Use this command after routing.

Example

```
set net [dbGetNetByName vdd]
set wire_info [dbInfoWire $wire]
```

This command gets the net "vdd" and outputs the wire information.

dbInitBumpGrid

`dbInitBumpGrid xspace xpitch xnum yspace ypitch ynum`

Initializes the bump grid.

Parameters

<i>xspace</i>	Horizontal distance to be reserved outside the edges
<i>xpitch</i>	Pitch between bumps in the horizontal direction
<i>xnum</i>	Number of bumps in the horizontal direction
<i>yspace</i>	Vertical distance to be reserved outside the edges
<i>ypitch</i>	Pitch between bumps in the vertical direction
<i>ynum</i>	Number of bumps in the vertical direction

Command Order

Use this command after design import.

Example

```
dbInitBumpGridb 3000 3000 40 3000 3000 40
```

A bump grid is created starting 3.0 microns from the outside edges, with the horizontal and vertical spacing between bumps of 3.0 microns and 40 bumps in the horizontal and vertical directions.

dbInitCellNetList

dbInitCellNetList *cellPtr*

Initializes the netlist of the cell.

Parameters

<i>cellPtr</i>	Address of cell
----------------	-----------------

Command Order

Use this command after design import.

Example

```
dbInitInitCellNetList [dbgTopCell]
```

The top cell netlist has been initialized.

dbInitVCellNetList

dbInitVCellNetList *cellPtr*

Initializes the Verilog netlist of the cell.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

dbInstBaseName

dbInstBaseName *instPtr*

Returns the base name of the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

For the instance, I526/I22/LIN_ADDR/U7/U27:

```
set cellPtr [dbInstCell $objPtr]  
0x4917910  
dbInstBaseName $objPtr
```

Returns:

U27

The base name is 027 for the specified instance.

dbInstBox

dbInstBox *instPtr*

Returns the box of the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstBox $objPtr
```

Returns:

```
563000 486000 598000 513000
```

The lower left coordinate is (563.0 486.0) and the upper right coordinate is (598.0 513.0).

dbInstCell

```
dbInstCell instPtr
```

Returns the address of the cell for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstCell $objPtr
```

Returns:

0x4917910

The master cell address is 0x4917910.

dbInstCellName

```
dbInstCellName instPtr
```

Returns the name of the cell master for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstCellName $objPtr
```

Returns:

EYFD1

The name of the master cell is EYFD1.

dbInstCluld

dbInstCluId *instPtr*

Returns the cluster ID for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after placement.

Example

dbInstCluId \$objPtr

Returns:

0

The cluster ID is 0.

dbInstCongest

dbInstCongest *instPtr*

Returns the congestion for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after placement.

Example

```
dbInstCongest $objPtr
```

Returns:

0.0

The congestion is 0.0 for the specified instance.

dbInstEffIGLen

```
dbInstEffIGLen instPtr
```

Returns the effective grid length for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbInstEffIGLen $objPtr
```

Returns:

0.0

The effective grid length is 0.0.

dbInstGroup

dbInstGroup *instPtr*

Returns the group address for the specified instance. The command returns 0x0 if the instance does not belong to a group.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbInstGroup [dbGetInstByName DTMF_INST/i_10048]
```

Returns:

0x0

The instance DTMF_INST/i_10048 does not belong to any group.

dbInstHasObstruct

dbInstHasObstruct *instPtr*

Returns a 1 if the specified instance has an obstruct, else a 0.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstHasObstruct $instPtr
```

Returns:

0

The instance does not have an obstruction.

dbInstHierHInst

dbInstHierHInst

```
dbInstHierHInst hinstPtr
```

Returns the address of the associated hierarchical instance (module) for the specified hierarchical instance, 0x0 if there is none.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbInstHierHInst $instPtr
```

Returns:

0xd911168

The address of the module for the specified instance is 0xd911168.

dbInstHInst

```
dbInstHInst instPtr
```

Returns the address of the associated hierarchical instance for the specified instance, 0x0 if there is none.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstHInst $objPtr
```

Returns:

0x0

There is no hierarchical instance for the specified instance.

dbInstHorOrder

dbInstHorOrder *instPtr*

Returns the horizontal order for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstHorOrder $objPtr
```

Returns:

0

The horizontal order is 0.

dbInstIGArea

dbInstIGArea *instPtr*

Returns the instance grid area for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstIGArea $objPtr
```

Returns:

7378.0

The instance grid area is 7378.0 square microns for the specified instance.

dbInstIGLen

```
dbInstIGLen instPtr
```

Returns the instance grid length for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbInstIGLen $objPtr
```

Returns:

434

The instance grid length is 0.434.

dbInstInternalPower

dbInstInternalPower *instPtr*

Returns the internal power for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

dbInstInternalPower [dbGetInstByName DTMF_INST/i_10048]

Returns:

0.0

The internal power for the instance DTMF_INST/i_10048 is 0.0.

dbInstLeakagePower

dbInstLeakagePower *instPtr*

Returns the leakage power for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbInstLeakagePower [dbGetInstByName DTMF_INST/i_10048]
```

Returns:

0.0

The leakage power for the instance DTMF_INST/i_10048 is 0.0.

dbInstLibraryContext

```
dbInstLibraryContext inst
```

Returns the library context for the specified instance.

Parameters

<i>inst</i>	Address of instance.
-------------	----------------------

Command Order

Use this command after importing the design.

dbInstLoc

dbInstLoc *instPtr*

Returns the location for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

dbInstLoc \$objPtr

Returns:

814111 785088

The specified instance is at 814.111 785.088.

dbInstName

dbInstName *instPtr*

Returns the instance name for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstName $objPtr
```

Returns:

```
I167
```

The instance name is I167.

dbInstNext

```
dbInstNext instPtr
```

Returns the address of the next instance for the specified instance, 0x0 if there is none.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstNext $objPtr
```

Returns:

0x0

There is not another instance.

dbInstNrBidi

dbInstNrBidi *instPtr*

Returns the number of bidirectional terminals for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

dbInstNrBidi \$objPtr

Returns:

0

There are no bidirectional terminals for the specified instance.

dbInstNrInput

dbInstNrInput *instPtr*

Returns the number of inputs for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstNrInput $objPtr
```

Returns:

171

There are 171 input terminals.

dbInstNrOutput

```
dbInstNrOutput instPtr
```

Returns the number of outputs for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstNrOutput $objPtr
```

Returns:

64

The instance has 64 output terminals.

dbInstNrRow

```
dbInstNrRow instPtr
```

Returns the number of rows the specified instance occupies.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstNrRow $objPtr
```

Returns:

17

The instance occupies 17 rows.

dbInstNrTerm

dbInstNrTerm *instPtr*

Returns the number of terminals for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

dbInstNrTerm \$objPtr

Returns:

235

The specified instance has 235 terminals.

dbInstObstruct

dbInstObstruct *instPtr*

Returns the metal layers obstructed in specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstObstruct $objPtr
```

Returns:

0x1

Only `metal 1` is obstructed for the specified instance.

dbInstObstructLayers

```
dbInstObstructLayers instPtr
```

Returns the metal layers obstructed in the specified instance, returns nothing if there are none.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstObstructLayers $instPtr
```

Returns:

There are no metal layers obstructed for the specified instance.

dbInstOrient

`dbInstOrient instPtr`

Returns the orientation for the specified instance. Possible values are:

`dbcR0`

`dbcMY`

`dbcMX`

`dbcR180`

`dbcMX90`

`dbcR90`

`dbcR270`

`dbcMY90`

Parameters

<code><i>instPtr</i></code>	Address of instance.
-----------------------------	----------------------

Command Order

Use this command after design import, but makes more sense after placement.

Example

`dbInstOrient $objPtr`

Returns:

`dbcR0`

The orientation is R0.

dbInstPCell

dbInstPCell *instPtr*

Returns the address of the physical cell for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

dbInstPCell \$objPtr

Returns:

0x4e9ddc0

The address of the physical cell for the specified instance is 0x4e9ddc0.

dbInstPdefName

dbInstPdefName *instPtr*

Returns the pdef name of the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstPdefName $instPtr
```

Returns:

DTMF_INST/ROM_512x16_0_INST

The PDEF name for the specified instance is DTMF_INST/ROM_512x16_0_INST.

dbInstPGTermArr

```
dbInstPGTermArr instPtr
```

Returns the address of the power/ground terminals for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstPGTermArr $objPtr
```

Returns:

0x0

There is no address for the power/ground terminal array.

dbInstPlacementStatus

dbInstPlacementStatus *instPtr*

Returns the placement status for the specified instance. The possible states are:

dbcUnplaced

dbcFixed

dbcPlaced

dbcCover

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbInstPlacementStatus [dbGetInstByName DTMF_INST/i_10048]
```

Returns:

dbcUnplaced

The placement status for the instance DTMF_INST/i_10048 is unplaced.

dbInstPrefixId

dbInstPrefixId *instPtr*

Returns the prefix ID for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
bInstPrefixId $objPtr
```

Returns:

0

The prefix ID for the specified instance is 0.

dbInstPrefixName

dbInstPrefixName *instPtr*

Returns prefix name if there is one for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstPrefixName $objPtr
```

Returns:

There is no prefix name.

dbInstPrev

```
dbInstPrev instPtr
```

Returns the address of the previous instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstPrev $objPtr
```

Returns:

0x4d3d82c

The address of the previous instance is 0x4d3d82c.

dbInstPriority

dbInstPriority *instPtr*

Returns the instance priority for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

dbInstPriority \$objPtr

Returns:

0

The instance priority is 0.

dbInstRoutingHaloBottomLayer

dbInstRoutingHaloBottomLayer *instPtr*

Returns the address of the bottom layer of the routing halo for the specified instance. The command returns 0x0 if no routing halo exists.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbInstRoutingHaloBottomLayer [dbGetInstByName DTMF_INST/i_10048]
```

Returns:

0x0

There is no routing halo.

dbInstRoutingHaloSize

```
dbInstRoutingHaloSize instPtr
```

Returns the size of the routing halo for the specified instance. The command returns 0 if no routing halo exists.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbInstRoutingHaloSize[dbGetInstByName DTMF_INST/i_10048]
```

Returns:

0

There is no routing halo.

dbInstRoutingHaloTopLayer

```
dbInstRoutingHaloTopLayer instPtr
```

Returns the address of the top layer of the routing halo for the specified instance. The command returns `0x0` if no routing halo exists.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbInstRoutingHaloTopLayer [dbGetInstByName DTMF_INST/i_10048]
```

Returns:

0x0

There is no routing halo.

dbInstTempId

dbInstTempId *instPtr*

Returns the temporary ID for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstTempId $objPtr
```

Returns:

80992300

The temporary ID for the specified instance is 80992300.

dbInstTermArr

dbInstTermArr *instPtr*

Returns the address of the array terminals for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstTermArr $objPtr
```

Returns:

0x4d91b28

The address for the terminal array for the specified instance is 0x4d91b28.

dbInstTermList

```
dbInstTermList instPtr
```

Returns address of the list of terminals associated with the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbInstTermList $objPtr
```

Returns:

0x4d91b28

The address of the term list for the specified instance is 0x4d91b28.

dbInstVerOrder

dbInstVerOrder *instPtr*

Returns the vertical order for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

dbInstVerOrder \$objPtr

Returns:

0

The vertical order for the specified instance is 0.

dbIoBox

dbIoBox *ioPtr*

Returns the box for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import.

Example

```
dbInstBox $ioPtr
```

Returns:

```
0 589650 330300 6589650
```

The box for the I/O is (0.0 589.650) (330.3 6589.650).

dbIoBump

```
dbIoBump ioPtr
```

Returns the address of the bump cell for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after importing the design.

Example

```
dbInstBump $ioPtr
```

Returns:

```
0x29401938
```

The address of the bump for the specified I/O is 29401938.

dbIoCell

```
dbIoCell ioPtr
```

Returns the address of the cell master for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set io [dbIoCell $ioPtr]  
    puts "io = $io"  
}
```

The addresses for the four I/Os are reported:

```
io = 0x49996a8  
io = 0x4984090
```

```
io = 0x4999870
io = 0x4984258
```

dbIoFTerm

dbIoFTerm *ioPtr*

Returns the address of the f-term for the specified I/O, 0x0 if there is none.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {
    set fterm [dbIoFTerm $ioPtr]
    puts "fterm = $fterm"
}
```

This script reports the address of all f-terms for all I/Os in the top cell.

dbIoIndent

dbIoIndent *ioPtr*

Returns the indent for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after importing the design.

Example

```
dbIoIndent $ioPtr
```

Returns:

0.56

The indent is 0.56 for the specified I/O.

dbIoInst

```
dbIoInst ioPtr
```

Returns the address of the instance specified by the I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set inst [dbIoInst $ioPtr]  
    puts "inst = $inst"  
}
```

The address of the instance for the four I/Os in the top cell are reported:

```
inst = 0x8ccfae4  
inst = 0x4f1b4b0  
inst = 0x8faeb1c  
inst = 0x8da69d4
```

dbIoLayerId

`dbIoLayerId ioPtr`

Returns the layer ID for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after importing the design.

Example

```
dbIoLayerId $ioPtr
```

Returns:

2

The layer ID is `2` for the specified I/O.

dbIoLoc

`dbIoLoc ioPtr`

Returns the location of the specified I/O.

Parameters

<code>ioPtr</code>	Address of I/O.
--------------------	-----------------

Command Order

Use this command after design import, but makes more sense if I/Os have been placed.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set loc [dbIoLoc $ioPtr]  
    puts "loc = $loc"  
}
```

The location for all four I/Os in the top cell are reported:

```
loc = 6849000 589650  
loc = 0 589650  
loc = 589650 0  
loc = 589650 6849000
```

dbIoName

dbIoName *ioPtr*

Returns the name of the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set name [dbIoName $ioPtr]  
    puts "name = $name"  
}
```

The four instance names are returned for the four top-level I/Os:

```
name = SH16/I209  
name = SH15/I194  
name = SH14/I412  
name = SH13/I303
```

dbIoNext

dbIoNext *ioPtr*

Returns the address of the next I/O specified by the given I/O, 0x0 if there is none.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set next [dbIoNext $ioPtr]  
    puts "next = $next"  
}
```

The next address is reported. Note, the last "next" address is 0x0.

```
next = 0x9651bf0  
next = 0x9651bb0  
next = 0x9651b70  
next = 0x0
```

dbIoOffset

```
dbIoOffset ioPtr
```

Returns the offset for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after importing the design.

Example

```
dbIoOffset $ioPtr
```

Returns:

0.560

The offset is 0.56 for the specified I/O.

dbIoOrder

```
dbIoOrder ioPtr
```

Returns the order of the I/O (consult user guide) for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after I/O placement.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {
```

```
set order [dbIoOrder $ioPtr]  
puts "order = $order"  
}
```

All four top level I/Os have an order of 1:

```
order = 1  
order = 1  
order = 1  
order = 1
```

dbIoOrient

```
dbIoOrient ioPtr
```

Returns the orientation for the specified Io.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import and I/O placement.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
  
set orient [dbIoOrient $ioPtr]  
  
puts "orient = $orient"  
}
```

The orientation for each of the four top-level I/Os is reported:

```
orient = dbcR180
orient = dbcR0
orient = dbcR90
orient = dbcR270
```

dbIoRow

dbIoRow *ioPtr*

Returns the row for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import and I/O placement.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {
    set row [dbIoRow $ioPtr]
    puts "row = $row"
}
```

The script will report the row address of each top level I/O.

dbIoSide

dbIoSide *ioPtr*

Returns the side for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import and I/O placement.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set side [dbIoSide $ioPtr]  
    puts "side = $side"  
}
```

Each side for each of the four top level I/Os is reported:

```
side = dbcE  
side = dbcW  
side = dbcS  
side = dbcN
```

dbIoSpacing

dbIoSpacing *ioPtr*

Returns the spacing for the specified I/O.

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after importing the design.

Example

```
dbIoSpacing $ioPtr
```

Returns:

0.28

The spacing is 0.28 for the specified I/O.

dbIoTdfLibName

```
dbIoTdfLibName ioPtr
```

Returns the name.

Parameters

<i>ioPtr</i>	Address of Io library.
--------------	------------------------

Command Order

Use this command after design import.

Example

```
dbIoTdfLibName $objPtr  
tsmc_025_io
```

dbIsAllInstPlaced

dbIsAllInstPlaced

Reports whether all instances are placed. The command returns 1 if all instances are placed and 0 if they are not.

Parameters

None

Command Order

Use this command after importing the design.

Example

dbIsAllInstPlaced

Returns:

1

All instances are placed.

dbIsBackslashInNamesHidden

dbIsBackslashInNamesHidden

Reports whether the backslash character is hidden in names. The command returns `1` if the backslash character is hidden and `0` if it is not.

Parameters

None

Command Order

Use this command after importing the design.

Example

```
dbIsBackslashInNamesHidden
```

Returns:

`1`

The backslash character is hidden in names.

dbIsBoxOverlappingBox

```
dbIsBoxOverlappingBox box1 box2
```

Reports whether a specified box overlaps another specified box. The command returns `1` if the first box overlaps the second box and `0` if it does not.

Parameters

<code>box1</code>	<code>llx_coord lly_coord urx_coord ury_coord</code>
<code>box2</code>	<code>llx_coord lly_coord urx_coord ury_coord</code>

Command Order

Use this command after design import.

Example

```
set box1 [dbHInstBox [dbGetHInstByName DTMF_INST]]  
345200 723400 482000 845000  
  
set box1 [dbHInstBox [dbGetHInstByName IOPADS_INST]]  
346200 623400 487000 745000  
  
dbIsBoxOverlappingOrTouchingBox $box1 $box2
```

Returns:

1

The module `DTMF_INST` overlaps module `IOPADS_INST`.

dbIsBoxOverlappingOrTouchingBox

`dbIsBoxOverlappingOrTouchingBox bumpPtr`

Reports whether the box of the specified bump is overlapping or touching another box. The command returns a `1` if the bump's box is overlapping or touching and `0` if it is not.

Parameters

<code>bumpPtr</code>	Address of bump.
----------------------	------------------

Command Order

Use this command after importing the design.

Example

```
dbIsBoxOverlappingOrTouchingBox $objPtr
```

Returns:

1

The bump's box is overlapping or touching another box.

dbIsBumpHilite

```
dbIsBumpHilite bumpPtr
```

Reports whether the specified bump is highlighted. The command returns `1` if the bump is highlighted and `0` if it is not.

Parameters

<i>bumpPtr</i>	Address of bump.
----------------	------------------

Command Order

Use this command after importing the design and loading an appropriate I/O file.

Example

```
dbIsBumpHilite $objPtr
```

Returns:

1

The bump is highlighted.

dbIsBumpSel

`dbIsBumpSel bumpPtr`

Returns a 1 if the specified bump is selected, else a 0.

Parameters

<code>bumpPtr</code>	Address of bump.
----------------------	------------------

Command Order

Use this command after design import (including loading of an appropriate I/O file).

Example

`dbIsBumpSel $objPtr`

Returns:

1

The bump is selected.

dbIsCaseSensitive

`dbIsCaseSensitive`

Returns a 1 if names are case sensitive, else a 0.

Parameters

None

Command Order

Use this command after design import.

Example

```
dbIsCaseSensitive
```

Returns:

1

The names are case sensitive.

dbIsCellAreaIo

```
dbIsCellAreaIo instPtr
```

Returns a 1 if the specified cell is an area I/O, else a 0.

Parameters

<i>instPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellAreaIo $cellPtr
```

Returns:

0

The cell is not an area I/O cell.

dbIsCellBlackBox

`dbIsCellBlackBox cellPtr`

Returns a 1 if the specified cell is a blackbox, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

`dbIsCellBlackBox $cellPtr`

Returns:

0

The cell is not a blackbox.

dbIsCellBlock

`dbIsCellBlock cellPtr`

Returns a 1 if the specified cell is a block, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellBlock $cellPtr
```

Returns:

0

The cell is not a block.

dbIsCellCdumpDefined

```
dbIsCellCdumpDefined cellPtr
```

Returns a 1 if the specified cell has an associated cdump file, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellCdumpDefined $cellPtr
```

Returns:

1

The cell has an associated cdump file.

dbIsCellClockSynthesized

```
dbIsCellClockSynthesized cellPtr
```

Reports whether the specified cell has been clock synthesized. The command returns 1 if the cell has been clock synthesized and 0 if it has not.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after importing the design.

Examples

```
dbIsCellClockSynthesized $cellPtr
```

Returns:

1

The cell has been clock synthesized.

dbIsCellDisplayable

```
dbIsCellDisplayable cellPtr
```

Returns a 1 if the specified cell is displayable, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellDisplayable $cellPtr
```

Returns:

0

The cell is not displayable.

dbIsCellDontTouch

```
dbIsCellDontTouch cellPtr
```

Returns a 1 if the specified cell is a dont_touch cell, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellDont Touch$cellPtr
```

Returns:

0

The cell is not a dont_touch cell.

dbIsCellDontUse

```
dbIsCellDontUse cellPtr
```

Returns a 1 if the specified cell is a dont_use cell, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellDontUse $cellPtr
```

Returns:

0

The cell is not a dont_use cell.

dbIsCellDummy

`dbIsCellDummy cellPtr`

Returns a 1 if the specified cell is a dummy cell, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

`dbIsCellDummy $cellPtr`

Returns:

0

The cell is not a dummy cell.

dbIsCellECOMarked

`dbIsCellECOMarked cellPtr`

Returns a 1 if the specified cell has been marked by ECO, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellECOMarked $cellPtr
```

Returns:

0

The cell is not ECO marked.

dbIsCellECOMarked2

```
dbIsCellECOMarked2 cellPtr
```

Returns a 1 if the specified cell is ECO marked 2, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellECOMarked2 $cellPtr
```

Returns:

0

The cell is not ECO marked 2.

dbIsCellECOMarked4

`dbIsCellECOMarked4 cellPtr`

Returns a 1 if the specified cell is ECO marked 4, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

`dbIsCellECOMarked4 $cellPtr`

Returns:

0

The cell is not ECO marked 4.

dbIsCellEEQCell

`dbIsCellEEQCell cellPtr`

Returns a 1 if the specified cell is an electrically equivalent cell, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellEEQCell $cellPtr
```

Returns:

0

The cell is not electrically equivalent.

dbIsCellFlat

```
dbIsCellFlat cellPtr
```

Returns a 1 if the specified cell is flat, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellFlat [dbgTopCell]
```

Returns:

1

The top level cell is flat.

dbIsCellGate

`dbIsCellGate cellPtr`

Returns a 1 if the specified cell is a gate, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

`dbIsCellGate $cellPtr`

Returns:

1

The cell is a gate.

dbIsCellGRouted

`dbIsCellGRouted cellPtr`

Reports whether the specified cell has been globally routed. The command returns 1 if the cell has been globally routed and 0 if it has not.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after importing the design.

Examples

```
dbIsCellGRouted $cellPtr
```

Returns:

1

The cell has been globally routed.

dbIsCellHier

```
dbIsCellHier cellPtr
```

Reports whether the specified cell is hierarchical. The command returns 1 if it is hierarchical and 0 if it is not.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
dbIsCellHier [dbgTopCell]
```

Returns:

0

The cell is not hierarchical.

dbIsCellIo

```
dbIsCellIo cellPtr
```

Reports whether the specified cell is an I/O. The command returns 1 if it is an I/O and 0 if it is not.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellIo $cellPtr
```

Returns:

1

The cell is an I/O.

dbIsCellIoPlaced

`dbIsCellIoPlaced cellPtr`

Returns a 1 if the specified I/O cell is placed, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

`dbIsCellIoPlaced $cellPtr`

Returns:

0

The I/O cell is not placed.

dbIsCellJtagCell

`dbIsCellJtagCell cellPtr`

Returns a 1 if specified cell is a Jtag cell, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellJtagCell $cellPtr
```

Returns:

0

The cell is not a Jtag cell.

dbIsCellLatch

```
dbIsCellLatch cellPtr
```

Reports whether the specified cell is a latch cell. The command returns 1 if the cell is a latch cell and 0 if it is not.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
dbIsCellLatch [dbGetCellByName BUFX1]
```

Returns:

0

The cell is not a latch cell.

dbIsCellLeafCell

`dbIsCellLeafCell cellPtr`

Returns a 1 if specified cell is a leaf cell, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

`dbIsCellLeafCell [dbGetCellByName YFD1]`

Returns:

1

The cell is a leaf cell.

dbIsCellMarked

`dbIsCellMarked cellPtr`

Returns a 1 if the specified cell is marked, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellMarked $cellPtr
```

Returns:

0

The cell is not marked.

dbIsCellMarked2

```
dbIsCellMarked2 cellPtr
```

Returns a 1 if specified cell is marked 2, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellMarked2 $cellPtr
```

Returns:

0

The cell is not marked 2.

dbIsCellMarked3

```
dbIsCellMarked3 cellPtr
```

Returns a 1 if the specified cell is marked 3, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellMarked3 $cellPtr
```

Returns:

0

The cell is not marked 3.

dbIsCellMarked4

dbIsCellMarked4 *cellPtr*

Returns a 1 if cell is marked 4, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

dbIsCellMarked4 \$cellPtr

Returns:

0

The cell is not marked 4.

dbIsCellMaybeHier

dbIsCellMaybeHier *cellPtr*

Returns a 1 if the specified cell might be hierarchical, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellMaybeHier $cellPtr
```

Returns:

0

The cell is not hierarchical.

dbIsCellNLDefined

```
dbIsCellNLDefined cellPtr
```

Returns a 1 if the specified cell netlist is defined.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellNLDefined $cellPtr
```

Returns:

1

The netlist is defined.

dbIsCellNLRefed

`dbIsCellNLRefed cellPtr`

Returns a 1 if the specified cell is referenced in the netlist, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellNLRefed $cellPtr
```

Returns:

1

The cell is referenced in the netlist.

dbIsCellPartition

`dbIsCellPartition cellPtr`

Returns a 1 if the specified cell is a partition, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellPartition $cellPtr
```

Returns:

0

The cell is not a partition.

dbIsCellPlaced

```
dbIsCellPlaced cellPtr
```

Returns a 1 if the specified cell is placed, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellPlaced $cellPtr
```

Returns:

0

The cell is not placed.

dbIsCellPower

`dbIsCellPower cellPtr`

Returns a 1 if the specified cell is a power cell, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

`dbIsCellPower $cellPtr`

Returns:

0

The cell is not a power cell.

dbIsCellPowerAnalyzed

`dbIsCellPowerAnalyzed cellPtr`

Reports whether the specified cell has been power analyzed. The command returns 1 if the cell has

been power analyzed and 0 if it has not.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after importing the design.

Examples

```
dbIsCellPowerAnalyzed $cellPtr
```

Returns:

0

The cell has not been power analyzed.

dbIsCellPrototype

```
dbIsCellPrototype cellPtr
```

Returns a 1 if the specified cell is a prototype, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellPrototype $cellPtr
```

Returns:

0

The cell is not a prototype cell.

dbIsCellRCExtracted

```
dbIsCellRCExtracted cellPtr
```

Returns a 1 if the specified cell has been extracted, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import, but makes more sense after route.

Example

```
dbIsCellRCExtracted [dbgTopCell]
```

Returns:

0

The cell has not been extracted.

dbIsCellRouted

`dbIsCellRouted cellPtr`

Returns a 1 if the specified cell has been routed, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellRouted [dbgTopCell]
```

Returns:

0

The cell has not been routed.

dbIsCellScanCell

`dbIsCellScanCell cellPtr`

Returns a 1 if the specified cell is a scan cell, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellScanCell $cellPtr
```

Returns:

0

The cell is not a scan cell.

dbIsCellScanOpted

```
dbIsCellScanOpted cellPtr
```

Reports whether the specified cell has been scan optimized. The command returns 1 if the cell has been scan optimized and 0 if it has not.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after importing the design.

Examples

```
dbIsCellScanOpted $cellPtr
```

Returns:

0

The cell has not been scan optimized.

dbIsCellSequential

`dbIsCellSequential cellPtr`

Returns a 1 if the specified cell is sequential, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

- `dbIsCellSequential $cellPtr`

Returns:

0

The cell is not a sequential cell.

- You can use this command along with other Innovus database commands to report the number of sequential cells in a design.

```
set cellPtr [dbgTopCell]  
  
set countSeq 0  
  
dbForEachCellInst $cellPtr instPtr {  
  
    set leafCell [ dbInstCellName $instPtr ]  
  
    if { [ dbIsCellSequential $leafCell ] } {  
  
        incr countSeq  
  
    }  
}
```

}

Note: This method includes Hard Macro, BSR cells (scan registers) etc. in the count.

dbIsCellSpareGate

dbIsCellSpareGate *cellPtr*

Returns a 1 if the specified cell is a spare gate, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

dbIsCellSpareGate \$cellPtr

Returns:

0

The cell is not a spare gate cell.

dbIsCellStdCell

dbIsCellStdCell *cellPtr*

Returns a 1 if the specified cell is a standard cell, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellStdCell $cellPtr
```

Returns:

0

The cell is not a standard cell.

dbIsCellStor

```
dbIsCellStor cellPtr
```

Returns a 1 if the specified cell is a storage cell, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellStor $cellPtr
```

Returns:

0

The cell is not a storage cell.

dbIsCellSuper

```
dbIsCellSuper cellPtr
```

Returns a 1 if the specified cell is a super cell, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellSuper [dbgTopCell]
```

Returns:

1

The cell is a super cell.

dbIsCellSymDegenerate

`dbIsCellSymDegenerate cellPtr`

Returns a 1 if the symbol table has been generated for the specified cell, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

`dbIsCellSymDegenerate $cellPtr`

Returns:

1

The symbol table for the cell has been generated.

dbIsCellSymmetryR90

`dbIsCellSymmetryR90 cellPtr`

Returns a 1 if the specified cell symmetry is allowed to be rotated 90, else a 0.

Parameters

<code>cellPtr</code>	Address of cell.
----------------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellSymmetryR90 $cellPtr
```

Returns:

1

The cell is allowed to rotate 90 degrees.

dbIsCellSymmetryX

```
dbIsCellSymmetryX cellPtr
```

Returns a 1 if the specified cell symmetry is allowed to be flipped about the X axis, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellSymmetryX $cellPtr
```

Returns:

1

The cell is allowed to be flipped about the X axis.

dbIsCellSymmetryY

```
dbIsCellSymmetryY cellPtr
```

Returns a 1 if the specified cell symmetry is allowed to be flipped about the Y axis, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellSymmetryY $cellPtr
```

Returns:

1

The cell is allowed to be flipped about the Y axis.

dbIsCellTGDelayUpdated

```
dbIsCellTGDelayUpdated cellPtr
```

Returns a 1 if timing graph's delay has been updated, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after

Example

```
dbIsCellTGDelayUpdated $cellPtr
```

Returns:

0

The timing graph has not been updated.

dbIsCellTimeDefined

```
dbIsCellTimeDefined cellPtr
```

Returns a 1 if the specified cell timing has been defined, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellTimeDefined $cellPtr
```

Returns:

1

The cell timing has been defined.

dbIsCellTop

`dbIsCellTop cellPtr`

Returns a 1 if the specified cell is the top cell, else a 0.

Parameters

<code>cellPtr</code>	Specifies
----------------------	-----------

Command Order

Use this command after design import.

Example

`dbIsCellTop $cellPtr`

Returns:

0

The cell is not the top cell.

dbIsCellVCell

`dbIsCellVCell cellPtr`

Returns a 1 if the specified cell is a Verilog cell, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after

Example

```
dbIsCellIVCell $cellPtr
```

Returns:

1

The cell is a Verilog cell.

dbIsCellIVDDOnBottom

```
dbIsCellIVDDOnBottom cellPtr
```

Returns a 1 if the specified cell has power on the bottom, else a 0.

Parameters

<i>cellPtr</i>	Address of cell.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsCellVDDOnBottom $cellPtr
```

Returns:

0

The cell does not have power on the bottom.

dbIsClockHilite

```
dbIsClockHilite clockPtr
```

Reports whether the specified clock is highlighted. The command returns 1 if the clock is highlighted and 0 if it is not.

Parameters

<i>clockPtr</i>	Address of clock.
-----------------	-------------------

Command Order

Use this command after design import.

Example

```
dbForEachHeadClock [dbgHead] clockPtr {  
    set hilite [dbIsClockHilite $clockPtr]  
    puts " hilite = $hilite"  
}
```

Returns:

hilite = 0

The clock is not highlighted.

dbIsClockMarked

```
dbIsClockMarked clockPtr
```

Returns a 1 if the specified clock is marked, else a 0.

Parameters

<i>clockPtr</i>	Address of clock.
-----------------	-------------------

Command Order

Use this command after design import.

Example

```
dbForEachHeadClock [dbgHead] clockPtr {  
    set marked [dbIsClockMarked $clockPtr]  
    puts " marked = $marked"  
}
```

```
marked = 0
```

The clock is not marked.

dbIsClockSel

`dbIsClockSel clockPtr`

Returns a 1 if the specified clock is selected, else a 0.

Parameters

<code>clockPtr</code>	Address of clock.
-----------------------	-------------------

Command Order

Use this command after design import.

Example

```
dbForEachHeadClock [dbgHead] clockPtr {  
    set sel [dbIsClockSel $clockPtr]  
    puts " sel = $sel"  
}
```

Returns:

`sel = 0`

The clock is not selected.

dbIsExtractRCRandomMode

`dbIsExtractRCRandomMode`

Returns a 1 if the RC extraction is random access IPDB mode, else a 0.

Command Order

Use this command after extraction.

Example

dbIsExtractRCRandomMode

Returns:

0

The design is not extracted in random mode.

dbIsExtractRCRCDBMode

dbIsExtractRCRCDBMode

Returns a 1 if the extraction mode is rcdb, else a 0.

Command Order

Use this command after extraction.

Example

dbIsExtractRCRCDBMode

Returns:

0

The FPin is not a must join.

dbIsFPlanX

dbIsFPlanX *fplanPtr*

Returns a 1 if the specified floor plan has a duplicate (X, flat/hier), else a 0.

Parameters

<i>fplanPtr</i>	Address of floorplan
-----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbIsFPlanX [dbHeadFPlan]
```

Returns:

0

The floorplan has no duplicate.

dbIsFTermAssigned

```
dbIsFTermAssigned ftermPtr
```

Returns a 1 if the specified f-term is assigned, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermAssigned [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term has not been unassigned.

dbIsFTermAsyncCtrl

dbIsFTermAsyncCtrl

```
dbIsFTermAsyncCtrl ftermPtr
```

Reports whether the specified f-term is asynchronously controlled. The command returns 1 if the f-term is asynchronously controlled and 0 if it is not.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

```
dbIsFTermAsyncCtrl [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term PIPBY is not asynchronously controlled.

dbIsFTermBidi

dbIsFTermBidi *ftermPtr*

Returns a 1 if the specified f-term is bidirectional, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermBidi [dbGetFTermByName PIPBY]
```

Returns:

0

The specified f-term is not bidirectional.

dbIsFTermBrkLoop

dbIsFTermBrkLoop *ftermPtr*

Returns a 1 if the specified f-term is used for breaking combinational loops, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermBrkLoop [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not used for breaking combinational loops.

dbIsFTermBus

```
dbIsFTermBus ftermPtr
```

Returns a 1 if f-term is a bus, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermBus [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not a bus.

dbIsFTermClk

dbIsFTermClk *ftermPtr*

Returns a 1 if the specified f-term is a clock, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

dbIsFTermClk [dbGetFTermByName PIPBY]

Returns:

0

The f-term is not a clock.

dbIsFTermContAssignLHS

dbIsFTermContAssignLHS *ftermPtr*

Returns a 1 if the specified f-term continues assigned on the left hand side, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import

Example

```
dbIsFTermContAssignLHS [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not continuously assigned to the left hand side.

dbIsFTermContinuousAssign

```
dbIsFTermContinuousAssign ftermPtr
```

Returns a 1 if the specified f-term is a continuously assigned, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermContinuousAssign [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not continuously assigned.

dbIsFTermD

```
dbIsFTermD ftermPtr
```

Returns a 1 if the specified f-term is a D input, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermD [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not a D input.

dbIsFTermFTerm

dbIsFTermFTerm *ftermPtr*

Returns a 1 if the specified f-term is a master f-term, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermFTerm [dbGetFTermByName PIPBY]
```

Returns:

1

The f-term is a master f-term.

dbIsFTermHilite

dbIsFTermHilite *ftermPtr*

Reports whether the specified f-term is highlighted. The command returns 1 if the f-term is highlighted and 0 if it is not.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermHilite [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not highlighted.

dbIsFTermIgnored

```
dbIsFTermIgnored ftermPtr
```

Returns a 1 if specified f-term is ignored, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermIgnored [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not ignored.

dbIsFTermInput

`dbIsFTermInput ftermPtr`

Returns a 1 if specified f-term is an input, else a 0.

Parameters

<code><i>ftermPtr</i></code>	Address of f-term.
------------------------------	--------------------

Command Order

Use this command after design import.

Example

`dbIsFTermInput [dbGetFTermByName PIPBY]`

Returns:

0

The f-term is not an input.

dbIsFTermInternal

`dbIsFTermInternal ftermPtr`

Reports whether the specified f-term is internal. The command returns 1 if the f-term is internal and 0 if it is not.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

```
dbIsFTermInternal [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not an internal f-term.

dbIsFTermMarked

```
dbIsFTermMarked ftermPtr
```

Returns a 1 if the specified f-term is marked, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermMarked [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not marked.

dbIsFTermMarked2

dbIsFTermMarked2 *ftermPtr*

Returns a 1 if the specified f-term is marked 2, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermMarked2 [dbGetFTermByName BLU]
```

Returns:

0

The f-term is not marked 2.

dbIsFTermMarked3

dbIsFTermMarked3 *ftermPtr*

Returns a 1 if the specified f-term is marked 3, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermMarked3 [dbGetFTermByName BLU]
```

Returns:

0

The f-term is not marked 3.

dbIsFTermMPW

dbIsFTermMPW

```
dbIsFTermMPW ftermPtr
```

Reports whether the specified f-term is minimum pulse width. The command returns 1 if the f-term is minimum pulse width and 0 if it is not.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

```
dbIsFTermMPW [dbGetFTermByName vcom]
```

Returns:

0

The f-term is not minimum pulse width.

dbIsFTermNLDefined

```
dbIsFTermNLDefined ftermPtr
```

Returns a 1 if the specified f-term is defined in the netlist, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermNLDefined [dbGetFTermByName PIPBY]
```

Returns:

1

The f-term is defined in the netlist.

dbIsFTermNLRefed

dbIsFTermNLRefed *ftermPtr*

Returns a 1 if the specified f-term is referenced in the netlist, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermNLRefed [dbGetFTermByName PIPBY]
```

Returns:

1

The f-term is referenced in the netlist.

dbIsFTermOutput

dbIsFTermOutput *ftermPtr*

Returns a 1 if the specified f-term is an output, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermOutput [dbGetFTermByName PIPBY]
```

Returns:

1

The f-term is an output.

dbIsFTermPreassigned

```
dbIsFTermPreassigned ftermPtr
```

Returns a 1 if the specified f-term has been preassigned, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermPreassigned [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not preassigned.

dbIsFTermQ

`dbIsFTermQ ftermPtr`

Returns a 1 if the specified f-term is a Q, else a 0.

Parameters

<code>ftermPtr</code>	Address of f-term.
-----------------------	--------------------

Command Order

Use this command after design import.

Example

`dbIsFTermQ [dbGetFTermByName PIPBY]`

Returns:

0

The f-term is not a Q input.

dbIsFTermScanClk

`dbIsFTermScanClk ftermPtr`

Returns a 1 if the specified f-term is a scan clock, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermScanClk [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is a scan clock.

dbIsFTermScanIn

```
dbIsFTermScanIn ftermPtr
```

Returns a 1 if the specified f-term is a scan in, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermScanIn [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is a scan in.

dbIsFTermScanInv

`dbIsFTermScanInv ftermPtr`

Returns a 1 if the specified f-term is an inverted scan, else a 0.

Parameters

<code><i>ftermPtr</i></code>	Address of f-term.
------------------------------	--------------------

Command Order

Use this command after design import.

Example

`dbIsFTermScanInv [dbGetFTermByName PIPBY]`

Returns:

0

The f-term is an inverted scan.

dbIsFTermScanOut

`dbIsFTermScanOut ftermPtr`

Returns a 1 if the specified f-term is a scan out, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermScanOut [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is a scan out.

dbIsFTermSel

```
dbIsFTermSel ftermPtr
```

Returns a 1 if the specified f-term is selected, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermSel [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is selected.

dbIsFTermSideAssigned

```
dbIsFTermSideAssigned ftermPtr
```

Returns a 1 if the specified f-term side has been assigned, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermSideAssigned [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term has not been assigned a side.

dbIsFTermSpecial

`dbIsFTermSpecial ftermPtr`

Returns a 1 if the specified f-term is a terminal on a special net (power/ground), else a 0.

Parameters

<code><i>ftermPtr</i></code>	Address of f-term.
------------------------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermSpecial [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not a special term.

dbIsFTermTAClkSrc

`dbIsFTermTAClkSrc ftermPtr`

Returns a 1 if the specified f-term is a timing analysis clock source, else a 0.

Parameters

<code><i>ftermPtr</i></code>	Address of f-term.
------------------------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermTAClkSrc [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is a timing analysis clock source.

dbIsFTermTAIgnored

```
dbIsFTermTAIgnored ftermPtr
```

Returns a 1 if specified f-term is ignored during timing analysis, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermTAIgnored [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not ignored during timing analysis.

dbIsFTermTieHi

`dbIsFTermTieHi ftermPtr`

Returns a 1 if the specified f-term is a tie-hi, else a 0.

Parameters

<code><i>ftermPtr</i></code>	Address of f-term.
------------------------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermTieHi [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not a tie-hi.

dbIsFTermTieLo

`dbIsFTermTieLo ftermPtr`

Returns a 1 if the specified f-term is a tie-lo, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermTieLo [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not a tie-lo.

dbIsFTermTimeDefined

```
dbIsFTermTimeDefined ftermPtr
```

Returns a 1 if the specified f-term timing is defined, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermTimeDefined [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term has timing defined.

dbIsFTermTriCtl

dbIsFTermTriCtl *ftermPtr*

Returns a 1 if the specified f-term is tristate controlled, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermTriCtl [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not tristate controlled.

dbIsFTermUnused

dbIsFTermUnused *ftermPtr*

Returns a 1 if specified f-term is unused, else a 0.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsFTermUnused [dbGetFTermByName PIPBY]
```

Returns:

0

The f-term is not unused.

dbIsGroupHilite

```
dbIsGroupHilite groupPtr
```

Reports whether the specified group is highlighted. The command returns `1` if the group is highlighted and `0` if it is not.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after importing the design and creating at least one group.

Example

```
dbIsGroupHilite $groupPtr  
0
```

The group is not highlighted.

dbIsGroupInferred

```
dbIsGroupInferred groupPtr
```

Returns a 1 if the specified group was inferred, else a 0.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import and creation of at least one group.

Example

```
createInstGroup my_group1 -guide 100 100 400 400  
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/U7/U43  
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/I52  
set groupPtr [dbGetGroupByName my_group1]
```

Returns:

0x54e04e8

Then:

dbIsGroupInferred \$groupPtr

Returns:

0

The group is not inferred.

dbIsGroupPhyHier

dbIsGroupPhyHier *groupPtr*

Returns a 1 if the specified group has a physical hierarchy, else a 0.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import and creation of at least one group.

Example

```
createInstGroup my_group1 -guide 100 100 400 400
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/U7/U43
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/I52
set groupPtr [dbGetGroupByName my_group1]
```

Returns:

0x54e04e8

Then:

dbIsGroupPhyHier \$groupPtr

Returns:

0

The group does not have physical hierarchy.

dbIsGroupPowerDomain

dbIsGroupPowerDomain *groupPtr*

Returns a 1 if the specified group is a power domain, else a 0.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import and creation of at least one group.

Example

```
createInstGroup my_group1 -guide 100 100 400 400
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/U7/U43
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/I52
set groupPtr [dbGetGroupByName my_group1]
```

Returns:

0x54e04e8

Then:

```
dbIsGroupPowerDomain $groupPtr
```

Returns:

0

The group is not a power domain.

dbIsGroupSel

`dbIsGroupSel groupPtr`

Returns a 1 if the specified group is selected, else a 0.

Parameters

<code>groupPtr</code>	Address of group.
-----------------------	-------------------

Command Order

Use this command after design import and creation of at least one group.

Example

```
createInstGroup my_group1 -guide 100 100 400 400
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/U7/U43
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/I52
set groupPtr [dbGetGroupByName my_group1]
```

Returns:

0x54e04e8

Then:

```
dbIsGroupSel $groupPtr
```

Returns:

0

The group is not selected.

dbIsGroupUngroup

dbIsGroupUngroup *groupPtr*

Returns a 1 if the specified group is ungrouped, else a 0.

Parameters

<i>groupPtr</i>	Address of group.
-----------------	-------------------

Command Order

Use this command after design import and creation of at least one group.

Example

```
createInstGroup my_group1 -guide 100 100 400 400
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/U7/U43
addInstToInstGroup my_group1 I526/I22/LIN_ADDR/I52
set groupPtr [dbGetGroupByName my_group1]
```

Returns:

0x54e04e8

Then:

```
dbIsGroupUngroup $groupPtr
```

Returns:

0

The group is not ungrouped.

dbIsHeadCustomLayerChanged

dbIsHeadCustomLayerChanged *headPtr*

Returns a 1 if custom layers have changed, else a 0.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsHeadCustomLayerChanged [dbgHead]
```

Returns:

0

The customer layers have not changed.

dbIsHeadDesignDisplayable

dbIsHeadDesignDisplayable *headPtr*

Returns a 1 if design is displayable, else a 0.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsHeadDesignDisplayable [dbgHead]
```

Returns:

1

The design is displayable.

dbIsHeadDesignGRouted

```
dbIsHeadDesignGRouted headPtr
```

Returns a 1 if design has been globally routed, else a 0.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsHeadDesignGRouted [dbgHead]
```

Returns:

1

The design is globally routed.

dbIsHeadDesignInMemory

`dbIsHeadDesignInMemory headPtr`

Returns a 1 if design is in memory, else a 0.

Parameters

<code>headPtr</code>	Address of head.
----------------------	------------------

Command Order

Use this command after design import.

Example

`dbIsHeadDesignInMemory [dbgHead]`

Returns:

1

The design is in memory.

dbIsHeadDesignPlaced

`dbIsHeadDesignPlaced headPtr`

Returns a 1 if design has been placed, else a 0.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsHeadDesignPlaced [dbgHead]
```

Returns:

1

The design is placed.

dbIsHeadDesignTGDelayUpdated

```
dbIsHeadDesignTGDelayUpdated headPtr
```

Returns a 1 if the delay timing graph has been updated, else a 0.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsHeadDesignTGDelayUpdated $objPtr
```

Returns:

0

The design delay timing graph has not been updated.

dbIsHeadFPlanChanged

```
dbIsHeadFPlanChanged headPtr
```

Returns a 1 if the floorplan has changed, else a 0.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsHeadFPlanChanged [dbHeadFPlan]
```

Returns:

1

The floorplan has changed.

dbIsHeadHiliteSticky

`dbIsHeadHiliteSticky headPtr`

Returns 1 if the highlighted set should be cleared with the select set, or 0 if it should not.

Parameters

<code>headPtr</code>	Address of head.
----------------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsHeadHiliteSticky [dbgHead]
```

Returns:

1

The highlighted set should be cleared.

dbIsHeadIGUSet

`dbIsHeadIGUSet headPtr`

Returns a 1 if the instance grid units have been set, else a 0.

Parameters

<code>headPtr</code>	Address of head.
----------------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsHeadIGUSet [dbgHead]
```

Returns:

0

The instance grid units have not been set.

dbIsHeadILMFlattened

```
dbIsHeadILMFlattened
```

Reports whether the ILM models have been flattened at the top level of the design. The command returns `1` if they have been flattened and `0` if they have not.

Parameters

None

Command Order

Use this command after importing the design.

Example

```
dbIsHeadILMFlattened
```

Returns:

0

The ILM models have not been flattened in the design.

dbIsHeadILMSpecified

dbIsHeadILMSpecified

Reports whether the ILM models have been specified at the top level of the design. The command returns 1 if they have been specified and 0 if they have not.

Parameters

None

Command Order

Use this command after importing the design.

Example

dbIsHeadILMSpecified

Returns:

0

The ILM models have not been specified in the design.

dbIsHeadInHierMode

dbIsHeadInHierMode *headPtr*

Returns a 1 if design is in hierarchical mode, else a 0.

Parameters

headPtr

Address of head.

Command Order

Use this command after design import.

Example

```
dbIsHeadInHierMode [dbgHead]
```

Returns:

1

The design is in hierarchical mode.

dbIsHeadLibraryInMemory

```
dbIsHeadLibraryInMemory headPtr
```

Returns a 1 if library is in memory, else a 0.

Parameters

headPtr

Address of head.

Command Order

Use this command after design import.

Example

```
dbIsHeadLibraryInMemory [dbgHead]
```

Returns:

1

The design is in memory.

dbIsHeadTAFuncExtracted

dbIsHeadTAFuncExtracted

Returns a 1 if design has been timing analyzed, else a 0.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

dbIsHeadTAFuncExtracted

Returns:

0

The design has been timing analyzed.

dbIsHeadTALibraryInMemory

dbIsHeadTALibraryInMemory *headPtr*

Returns a 1 if the timing analysis library is in memory, else a 0.

Parameters

<i>headPtr</i>	Address of head.
----------------	------------------

Command Order

Use this command after design import.

Example

```
dbIsHeadTALibraryInMemory [dbgHead]
```

Returns:

0

The design timing library is not in memory.

dbIsHInstAllOrdered

```
dbIsHInstAllOrdered hinstPtr
```

Returns a 1 if all hierarchical instances are ordered, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstAllOrdered $objPtr
```

Returns:

0

The modules are not ordered.

dbIsHInstDontTouch

```
dbIsHInstDontTouch hinstPtr
```

Returns a 1 if the specified hierarchical instance is a dont_touch, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstDontTouch $objPtr
```

Returns:

0

The module is not a dont_touch.

dbIsHInstFPlanChanged

`dbIsHInstFPlanChanged hinstPtr`

Returns a 1 if specified hierarchical instance floorplan has changed, else a 0.

Parameters

<code>hinstPtr</code>	Address of hierarchical instance.
-----------------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstFlanChangedP $objPtr
```

Returns:

0

The module floorplan has not changed.

dbIsHInstFPLeaf

`dbIsHInstFPLeaf hinstPtr`

Returns a 1 if the specified hierarchical instance is a leaf cell, else a 0.

Parameters

<code>hinstPtr</code>	Address of hierarchical instance.
-----------------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstFPLef $objPtr
```

Returns:

0

The hierarchical instance is not a leaf cell.

dbIsHInstHidden

```
dbIsHInstHidden hinstPtr
```

Returns a 1 if specified module is hidden, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance
-----------------	----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstHidden $objPtr
```

Returns:

1

The hierarchical instance is hidden.

dbIsHInstHilite

`dbIsHInstHilite hinstPtr`

Reports whether the specified hierarchical instance is highlighted. The command returns 1 if the hierarchical instance is highlighted and 0 if it is not.

Parameters

<code><i>hinstPtr</i></code>	Address of hierarchical instance.
------------------------------	-----------------------------------

Command Order

Use this command after design import.

Example

`dbIsHInstHilite $objPtr`

Returns:

1

The module is highlighted.

dbIsHInstHInst

`dbIsHInstHInst hinstPtr`

Returns a 1 if it is the master hierarchical instance, else a 0.

Parameters

hinstPtr

Address of hierarchical instance

Command Order

Use this command after design import.

Example

```
dbIsHInstHInst $objPtr
```

Returns:

1

The hierarchical instance is a master hierarchical instance.

dbIsHInstInFPlan

```
dbIsHInstInFPlan hinstPtr
```

Returns a 1 if the specified hierarchical instance is in the floorplan, else a 0.

Parameters

hinstPtr

Address of hierarchical instance.

Command Order

Use this command after design import.

Example

```
dbIsHInstInFPlan $objPtr
```

Returns:

0

The module is in the floorplan.

dbIsHInstJtag

`dbIsHInstJtag hinstPtr`

Returns a 1 if the specified module is a Jtag, else a 0.

Parameters

<code>hinstPtr</code>	Address of hierarchical instance.
-----------------------	-----------------------------------

Command Order

Use this command after design import.

Example

`dbIsHInstJtag $objPtr`

Returns:

0

The hierarchical instance is not Jtag.

dbIsHInstJtagElem

`dbIsHInstJtagElem hinstPtr`

Returns a 1 if the specified hierarchical instance is a Jtag element, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstJtagElem $objPtr
```

Returns:

0

The hierarchical instance is not a Jtag element.

dbIsHInstLeafRegion

```
dbIsHInstLeafRegion hinstPtr
```

Returns a 1 if the specified module is a leaf region, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstLeafRegion $objPtr
```

Returns:

0

The hierarchical instance is not a leaf region.

dbIsHInstMarked

```
dbIsHInstMarked hinstPtr
```

Returns a 1 if the specified hierarchical instance is marked, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstMarked $objPtr
```

Returns:

0

The module is not marked.

dbIsHInstMarked2

dbIsHInstMarked2 *hinstPtr*

Returns a 1 if the specified hierarchical instance is marked 2, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

dbIsHInstMarked2 \$objPtr

Returns:

0

The hierarchical instance is not marked 2.

dbIsHInstMarked3

dbIsHInstMarked3 *hinstPtr*

Returns a 1 if the specified module is marked 3, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstMarked3 $objPtr
```

Returns:

0

The hierarchical instance is not marked 3.

dbIsHInstOnCurrLevel

```
dbIsHInstOnCurrLevel hinstPtr
```

Returns a 1 if the specified hierarchical instance is on the current level, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstOnCurrLevel $objPtr
```

Returns:

1

The hierarchical instance is on the current level.

dbIsHInstOrdered

`dbIsHInstOrdered hinstPtr`

Returns a 1 if the specified module is ordered, else a 0.

Parameters

<code><i>hinstPtr</i></code>	Address of hierarchical instance.
------------------------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstOrdered $objPtr
```

Returns:

0

The hierarchical instance is not ordered.

dbIsHInstPinAssigned

`dbIsHInstPinAssigned hinstPtr`

Returns a 1 if the pins of the specified hierarchical instance have been assigned, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import, but makes more sense after partitioning and pin assignment.

Example

```
dbIsHInstPinAssigned $objPtr
```

Returns:

0

The modules pins have not been assigned.

dbIsHInstPinInitiated

```
dbIsHInstPinInitiated hinstPtr
```

Returns a 1 if the pins have been initialized for the specified hierarchical instance, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstPinInitiated $objPtr
```

Returns:

0

The pins have not been initialized.

dbIsHInstRegionDefined

`dbIsHInstRegionDefined hinstPtr`

Returns a 1 if the specified module's region has been defined, else a 0.

Parameters

<code><i>hinstPtr</i></code>	Address of hierarchical instance.
------------------------------	-----------------------------------

Command Order

Use this command after design import.

Example

`dbIsHInstRegionDefined $objPtr`

Returns:

0

The hierarchical instance's region has not been defined.

dbIsHInstSel

`dbIsHInstSel hinstPtr`

Returns a 1 if the specified hierarchical instance is selected, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstSel $objPtr
```

Returns:

1

The module is selected.

dbIsHInstUngroup

```
dbIsHInstUngroup hinstPtr
```

Returns a 1 if the specified hierarchical instance has been ungrouped, else a 0.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after design import.

Example

```
dbIsHInstUngroup $objPtr
```

Returns:

0

The module has not been ungrouped.

dbIsInstAreaIo

```
dbIsInstAreaIo instPtr
```

Reports whether the specified instance is an area I/O. The command returns a 1 if the instance is an area I/O and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

```
set objPtr [dbSelPtr $s]
```

Returns:

0x711f730

Then:

```
dbIsInstAreaIo $objPtr
```

Returns:

0

The instance is not an area I/O.

dbIsInstBlackBox

```
dbIsInstBlackBox instPtr
```

Reports whether the specified instance is a blackbox. The command returns a 1 if the instance is a blackbox and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

```
set objPtr [dbSelPtr $s]
```

Returns:

0x711f730

Then:

```
dbIsInstBlackBox $objPtr
```

Returns:

0

The instance is not a blackbox.

dbIsInstBlock

```
dbIsInstBlock instPtr
```

Reports whether the specified instance is a block. The command returns a 1 if the instance is a block and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

```
set objPtr [dbSelPtr $s]
```

Returns:

0x711f73

Then:

```
dbIsInstBlock $objPtr
```

Returns:

1

The instance is a block.

dbIsInstDefCovered

```
dbIsInstDefCovered instPtr
```

Reports whether the specified instance is defined in the DEF file. The command returns a 1 if the instance is in the DEF file and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0x711f730
```

Then:

```
dbIsInstDefCovered $objPtr
```

Returns:

```
0
```

The instance is not defined in DEF.

dbIsInstDefSrcTiming

```
dbIsInstDefSrcTiming instPtr
```

Reports whether the specified instance source timing is defined in the DEF file. The command returns a 1 if the instance source timing is in the DEF file and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x965eaf8
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0x711f730
```

Then:

```
dbIsInstDefSrcTiming $objPtr
```

Returns:

```
0
```

The instance does not have timing defined in the DEF.

dbIsInstDontTouch

```
dbIsInstDontTouch instPtr
```

Reports whether the specified instance is a dont_touch. The command returns a 1 if the instance is a dont_touch and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

```
set objPtr [dbSelPtr $s]
```

Returns:

0x711f730

Then:

```
dbIsInstDontTouch $objPtr
```

Returns:

0

The instance is not a dont_touch.

dbIsInstGroupInSelSet

```
dbIsInstGroupInSelSet headPtr ptr
```

Returns a 1 if the specified instance group is in selected set, else a 0.

Parameters

<i>headPtr</i>	Address of Head.
<i>ptr</i>	Address of group

Command Order

Use this command after

Example

```
dbIsInstGroupInSelSet [dbgHead] $objPtr
```

Returns:

0

The group is not in the selected set.

dbIsInstHaloBlock

```
dbIsInstHaloBlock instPtr
```

Reports whether the specified instance has a halo. The command returns a 1 if the instance has a halo and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

```
set objPtr [dbSelPtr $s]
```

Returns:

0x711f730

Then:

```
dbIsInstHaloBlock $objPtr
```

Returns:

0

The instance does not have a halo.

dbIsInstHidden

```
dbIsInstHidden instPtr
```

Reports whether the specified instance is hidden. The command returns 1 if the instance is hidden and 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsInstHidden [dbGetInstByName DTMF_INST/PLLCLK_INST]
```

Returns:

0

The instance is not hidden.

dbIsInstHilite

`dbIsInstHilite instPtr`

Reports whether the specified instance is a highlighted. The command returns `1` if the instance is highlighted and `0` if it is not.

Parameters

<code>instPtr</code>	Address of the instance.
----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

`set s [dbHeadSelList]`

Returns:

`0x965eaf8`

`set objPtr [dbSelPtr $s]`

Returns:

`0x711f730`

Then:

`dbIsInstHilite $objPtr`

Returns:

`0`

The instance is not highlighted.

dbIsInstHInst

`dbIsInstHInst instPtr`

Reports whether the specified instance is a hierarchical instance (module). The command returns a 1 if the instance is a hierarchical instance and a 0 if it is not.

Parameters

<code>instPtr</code>	Address of the instance.
----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

`set s [dbHeadSelList]`

Returns:

`0x965eaf8`

`set objPtr [dbSelPtr $s]`

Returns:

`0x711f730`

Then:

`dbIsInstHInst $objPtr`

Returns:

0

The instance is not a module.

dbIsInstIo

`dbIsInstIo instPtr`

Reports whether the specified instance is an I/O. The command returns 1 if the instance is an I/O and 0 if it is not.

Parameters

<code>instPtr</code>	Address of the instance.
----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

`set s [dbHeadSelList]`

Returns:

`0x965eaf8`

`set objPtr [dbSelPtr $s]`

Returns:

`0x711f730`

Then:

`dbIsInstIo $objPtr`

Returns:

0

The instance is not an I/O.

dbIsInstIPOed

dbIsInstIPOed *instPtr*

Reports whether in-place optimization (IPO) has been run for the specified instance. The command returns a 1 if the instance has been optimized and a 0 if it has not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

```
set objPtr [dbSelPtr $s]
```

Returns:

0x711f730

Then:

```
dbIsInstIPOed $objPtr
```

Returns:

0

The instance has not been optimized.

dbIsInstJtag

```
dbIsInstJtag instPtr
```

Reports whether the specified instance is a JTAG cell. The command returns a 1 if the instance is a JTAG cell and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

```
set objPtr [dbSelPtr $s]
```

Returns:

0x711f730

Then:

```
dbIsInstJtag $objPtr
```

Returns:

0

The instance is not Jtag.

dbIsInstJtagCell

```
dbIsInstJtagCell instPtr
```

Reports whether the specified cell is a JTAG cell. The command returns a 1 if the cell is a JTAG cell and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

```
set objPtr [dbSelPtr $s]
```

Returns:

0x711f730

```
dbIsInstIo $objPtr
```

Returns:

0

Then:

```
dbIsInstJtagCell $objPtr
```

Returns:

0

The instance is not a Jtag cell.

dbIsInstJtagElem

```
dbIsInstJtagElem instPtr
```

Reports whether the specified instance is a JTAG element. The command returns a `1` if the instance is a JTAG element and a `0` if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

0x965eaf8

set objPtr [dbSelPtr \$s]

Returns:

0x711f730

dbIsInstJtagElem \$objPtr

Returns:

<null>

The instance is not a Jtag element.

dbIsInstMarked

dbIsInstMarked *instPtr*

Reports whether the specified instance is marked. The command returns a 1 if the instance is marked and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

set s [dbHeadSelList]

Returns:

0x965eaf8

set objPtr [dbSelPtr \$s]

Returns:

0x711f730

dbIsInstMarked \$objPtr

Returns:

0

The instance is marked.

dbIsInstMarked2

dbIsInstMarked2 *instPtr*

Returns a 1 if the specified instance has been marked 2, else a 0.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

dbIsInstMarked2 \$objPtr

Returns:

0

The instance is not marked 2.

dbIsInstMarked3

dbIsInstMarked3 *instPtr*

Returns a 1 if the specified instance has been marked 3, else a 0.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

dbIsInstMarked3 \$objPtr

Returns:

0

The instance is not marked 3.

dbIsInstMoved

dbIsInstMoved *instPtr*

Returns a 1 if the specified instance has been moved, else a 0.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbIsInstM oved$objPtr
```

Returns:

0

The instance has not been moved.

dbIsInstOrientLegal

```
dbIsInstOrientLegal {instPtr | instPtr orient}
```

Check legality of orientation for the specified instance.

Parameters

<i>instPtr</i>	Address of the instance.
<i>orient</i>	Address of instance and orientation.

Command Order

Use this command after place.

Example

```
dbIsInstOrientLegal $objPtr
```

Returns:

1

The orientation is legal.

dbIsInstParentInSelSet

```
dbIsInstParentInSelSet headPtr ptr
```

Returns a 1 if the parent of the specified instance is in the selected set, else a 0

Parameters

<i>headPtr</i>	Address of Head.
<i>ptr</i>	Address of instance.

Command Order

Use this command after design import, but presumably after at least one object has been selected.

Example

```
dbIsInstParentInSelSet [dbgHead] $objPtr
```

Returns:

0

The parent of the instance is not in the selected set.

dbIsInstPartition

```
dbIsInstPartition instPtr
```

Reports whether the specified instance has been partitioned. The command returns a `1` if the instance has been partitioned and a `0` if it has not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after partitioning.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96ddcb0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0x71a30b8
```

Then:

```
dbIsInstPartition $objPtr
```

Returns:

```
0
```

The instance has not been partitioned.

dbIsInstPhysicalOnly

`dbIsInstPhysicalOnly instPtr`

Reports whether the specified instance is physical only. The command returns a `1` if the instance is physical only and a `0` if it is not.

Parameters

<code>instPtr</code>	Address of the instance.
----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> set s [dbHeadSelList]  
0x96ddcb0  
innovus 2> set objPtr [dbSelPtr $s]  
0x71a30b8  
innovus 7> dbIsInstPhysicalOnly $objPtr
```

Returns:

`0`

The instance is not a physical cell only.

dbIsInstPlaced

`dbIsInstPlaced instPtr`

Reports whether the specified instance is placed. The command returns a `1` if the instance is placed

and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> set s [dbHeadSelList]  
0x96ddcb0  
innovus 2> set objPtr [dbSelPtr $s]  
0x71a30b8  
innovus 8> dbIsInstPlaced $objPtr
```

Returns:

1

The instance is placed.

dbIsInstPreplaced

```
dbIsInstPreplaced instPtr
```

Reports whether the specified instance is preplaced. The command returns a 1 if the instance is preplaced and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> set s [dbHeadSelList]  
0x96ddcb0  
innovus 2> set objPtr [dbSelPtr $s]  
0x71a30b8  
innovus 11> dbIsInstPreplaced $objPtr
```

Returns:

1

The instance has been preplaced.

dbIsInstSel

dbIsInstSel *instPtr*

Returns a 1 if the specified instance is selected, else a 0.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import but makes more sense if at least one object is selected.

Example

```
dbIsInstSel $objPtr
```

Returns:

1

The instance is selected.

dbIsInstSpareGate

```
dbIsInstSpareGate instPtr
```

Returns a 1 if the specified instance is a spare gate, else a 0.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbIsInstSpareGate $objPtr
```

Returns:

0

The instance is not a spare gate.

dbIsInstStdCell

dbIsInstStdCell *instPtr*

Reports whether the specified cell is a standard cell. The command returns a 1 if the cell is a standard cell and a 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 19> set s [dbHeadSelList]  
0x96ddcb0  
innovus 20> set objPtr [dbSelPtr $s]  
0x71a30b8  
innovus 23> dbIsInstStdCell $objPtr
```

Returns:

1

The instance is a standard cell.

dbIsInstStor

`dbIsInstStor instPtr`

Reports whether the specified cell is a storage element. The command returns a `1` if the cell is a storage element and a `0` if it is not.

Parameters

<code>instPtr</code>	Address of the instance.
----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 24> dbIsInstStor $objPtr
0
```

The instance is not a storage element.

dbIsInstStorage

`dbIsInstStorage instPtr`

Reports whether the specified instance is a storage element. The command returns `1` if the instance is a storage element and `0` if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 25> dbIsInstStorage $objPtr  
0
```

The instance is not a storage element.

dbIsInstTAIgnored

dbIsInstTAIgnored *instPtr*

Reports whether the specified instance is ignored during timing analysis. The command returns 1 if the instance is ignored and 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 25> dbIsInstTAIgnored [dbGetInstByName DTMF_INST/i_10048]
```

0

The instance DTMF_INST/i_10048 is ignored during timing analysis.

dbIsInstUnused

dbIsInstUnused *instPtr*

Reports whether the specified instance is used in the design. The command reports 1 if it is used and 0 if it is not.

Parameters

<i>instPtr</i>	Address of the instance.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 25> dbIsInstUnused [dbGetInstByName DTMF_INST/i_10048]
```

0

The instance DTMF_INST/i_10048 is not used.

dbIsIoAssigned

dbIsIoAssigned *ioPtr*

Reports whether the specified I/O has been assigned. The command returns a 1 if the I/O has been assigned and a 0 if it is not.

Parameters

<i>iotPtr</i>	Address of the I/O.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 114> dbIsIoAssigned $ioPtr
```

Returns:

1

The I/O has been assigned.

dbIsIoBump

```
dbIsIoBump ioPtr
```

Reports whether the specified I/O is a bump. The command returns a 1 if the I/O is a bump and a 0 if it is not.

Parameters

<i>iotPtr</i>	Address of the I/O.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 114> dbIsIoBump $ioPtr
```

Returns:

1

The I/O is a bump.

dbIsIoClearance

```
dbIsIoClearance ioPtr
```

Reports whether the specified I/O is a clearance. The command returns a `1` if the I/O is a clearance and a `0` if it is not.

Parameters

<i>iotPtr</i>	Address of the I/O.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 114> dbIsIoClearance $ioPtr
```

Returns:

1

The I/O is a clearance.

dbIsIoCorner

`dbIsIoCorner ioPtr`

Reports whether the specified I/O is a corner cell. The command returns a `1` if the I/O is a corner cell and a `0` if it is not.

Parameters

<code><i>ioPtr</i></code>	Address of the I/O.
---------------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 115> dbIsIoCorner $ioPtr  
0
```

The I/O is not a corner.

dbIsIoCovered

`dbIsIoCovered ioPtr`

Reports whether the specified I/O is covered. The command returns a `1` if the I/O is covered and a `0` if it is not.

Parameters

<i>iotPtr</i>	Address of the I/O.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 114> dbIsIoCovered $ioPtr
```

Returns:

1

The I/O is a covered.

dbIsIoDefCovered

```
dbIsIoDefCovered ioPtr
```

Reports whether the specified I/O is defined in DEF with a status of cover (stronger than fixed).

Parameters

<i>ioPtr</i>	Address of I/O.
--------------	-----------------

Command Order

Use this command after design import.

dbIsIoDummyPad

```
dbIsIoDummyPad ioPtr
```

Reports whether the specified I/O is a dummy I/O pad. The command returns a `1` if the I/O is a dummy I/O pad and a `0` if it is not.

Parameters

<i>ioPtr</i>	Address of the I/O.
--------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set dummy [dbIsIoDummyPad $ioPtr]  
    puts "is dummy = $dummy"  
}
```

```
is dummy = 0  
is dummy = 0  
is dummy = 0
```

```
is dummy = 0
```

None of the 4 IOs are dummy IOs.

dbIsIoFixed

```
dbIsIoFixed ioPtr
```

Reports whether the specified I/O is fixed. The command returns a 1 if the I/O is fixed and a 0 if it is not.

Parameters

<i>ioPtr</i>	Address of the I/O.
--------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set dummy [dbIsIoFixed $ioPtr]  
    puts "fixed = $dummy"  
}
```

```
fixed = 1  
fixed = 1  
fixed = 1
```

fixed = 1

All 4 top level IOs are fixed.

dbIsIoGapFixed

dbIsIoGapFixed *ioPtr*

Reports whether the specified I/O is gap fixed. The command returns a 1 if the I/O is gap fixed and a 0 if it is not.

Parameters

<i>iotPtr</i>	Address of the I/O.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 114> dbIsIoGapFixed $ioPtr
1
```

The I/O is gap fixed.

dbIsIoGroundPad

dbIsIoGroundPad *ioPtr*

Reports whether the specified I/O is a ground pad. The command returns a 1 if the I/O is a ground

pad and a 0 if it is not.

Parameters

<i>ioPtr</i>	Address of the I/O.
--------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 123> dbIsIoGroundPad $ioPtr  
0
```

The I/O is not a ground pad.

dbIsIoHilite

```
dbIsIoHilite ioPtr
```

Reports whether the specified I/O is highlighted. The command returns 1 if the I/O is highlighted and 0 if it is not.

Parameters

<i>ioPtr</i>	Address of the I/O.
--------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set dummy [dbIsIoHilite $ioPtr]  
    puts "hilite = $dummy"  
}
```

```
hilite = 0  
hilite = 0  
hilite = 0  
hilite = 0
```

The four top-level I/Os are not highlighted.

dbIsIoMarked

```
dbIsIoMarked ioPtr
```

Reports whether the specified I/O is marked. The command returns a `1` if the I/O is marked and a `0` if it is not.

Parameters

<i>iotPtr</i>	Address of the I/O.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 114> dbIsIoMarked $ioPtr  
1
```

The I/O is marked.

dbIsIoPowerPad

```
dbIsIoPowerPad ioPtr
```

Reports whether the specified I/O is a power pad. The command returns a `1` if the I/O is a power pad and a `0` if it is not.

Parameters

<i>ioPtr</i>	Address of the I/O.
--------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 132> dbIsIoPowerPad $ioPtr  
1
```

The I/O is a power pad.

dbIsIoPreplaced

`dbIsIoPreplaced ioPtr`

Reports whether the specified I/O is preplaced. The command returns a `1` if the I/O is preplaced and a `0` if it is not.

Parameters

<code><i>iotPtr</i></code>	Address of the I/O.
----------------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 114> dbIsIoPreplaced $ioPtr
```

```
1
```

The I/O is preplaced.

dbIsIoSel

`dbIsIoSel ioPtr`

Reports whether the specified I/O is currently selected. The command returns a `1` if the I/O is currently selected and a `0` if it is not.

Parameters

<i>ioPtr</i>	Address of the I/O.
--------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellIo [dbgTopCell] ioPtr {  
    set dummy [dbIsIoSel $ioPtr]  
    puts "sel = $dummy"  
}
```

```
sel = 0  
sel = 0  
sel = 0  
sel = 0
```

The 4 top level IOs are not selected.

dbIsLayerBlkHilite

```
dbIsLayerBlkHilite layerBlkPtr
```

Reports whether the specified routing layer blockage is highlighted. The command returns `1` if the routing layer blockage is highlighted and `0` if it is not.

Parameters

<i>layerBlkPtr</i>	Address of the routing layer blockage.
--------------------	--

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
innovus 38> set s [dbHeadSelList]  
0x96ddcb0  
innovus 39> set objPtr [dbSelPtr $s]  
0x4a6e3d4  
innovus 40> dbIsLayerBlkHilite $objPtr  
1
```

The layer blockage is highlighted.

dbIsLayerBlkOnCutLayer

`dbIsLayerBlkOnCutLayer layerBlkPtr Int`

Reports whether the specified layer is part of the specified cut layer blockage. The command returns a `1` if the layer is part of the cut layer and a `0` if it is not.

Parameters

<i>layerBlkPtr</i>	Address of the cut layer.
<i>Int</i>	Metal layer.

Command Order

Use this command after importing the design and creating at least one cut layer blockage.if the partition

Example

```
innovus 38> set s [dbHeadSelList]  
0x96ddcb0  
innovus 39> set objPtr [dbSelPtr $s]  
0x4a6e3d4  
innovus 41> dbIsLayerBlkOnCutLayer $objPtr 3  
1
```

There is a cut layer blockage on *via3* for the selected cut blockage.

dbIsLayerBlkOnLayer

`dbIsLayerBlkOnLayer layerBlkPtr Int`

Reports whether the specified layer is part of the specified routing layer blockage. The command returns a `1` if the layer is part of the routing layer blockage and a `0` if it is not.

Parameters

<code>layerBlkPtr</code>	Address of the routing layer blockage.
<code>Int</code>	Metal layer.

Command Order

Use this command after importing the design and creating at least one routing layer blockage.if the partition

Example

```
innovus 38> set s [dbHeadSelList]  
0x96ddcb0  
  
innovus 39> set objPtr [dbSelPtr $s]  
0x4a6e3d4  
  
innovus 41> dbIsLayerBlkOnLayer $objPtr 3  
1
```

There is a routing layer blockage on *meta/3* for the selected routing blockage.

dbIsLayerBlkSel

`dbIsLayerBlkSel layerBlkPtr`

Reports whether the specified routing layer blockage is selected. The command returns `1` if the routing layer blockage is selected and a `0` if it is not.

Parameters

<code>layerBlkPtr</code>	Address of the routing layer blockage.
--------------------------	--

Command Order

Use this command after importing the design and creating at least one routing layer blockage.

Example

```
innovus 42> dbIsLayerBlkSel $objPtr  
1
```

The layer blockage is selected.

dbIsLayerH

dbIsLayerH *layerPtr*

Reports whether the specified layer is horizontal. The command returns 1 if the layer is a horizontal layer and 0 if it is not.

Parameters

<i>layerPtr</i>	Address of the layer
-----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 5> dbIsLayerH [dbGetLayerByName METAL1]
```

```
1
```

The layer *METAL1* is horizontal.

dbIsLayerMSLayer

dbIsLayerMSLayer *layerPtr body*

Reports whether the specified layer is MS.

Parameters

<i>layerPtr</i>	Address of the layer
-----------------	----------------------

Command Order

Use this command after importing the design.

dbIsLayerShapeShapeList

```
dbIsLayerShapeShapeList shapeList
```

Reports whether the specified list is a shape list. The command returns a 1 if the list is a shape list and a 0 if it is not.

Parameters

<i>shapeList</i>	Shape list.
------------------	-------------

Command Order

Use this command after importing the design.

Example

```
proc printOneLayerShape {ls} {
    if {[dbIsLayerShapeShapeList $ls]} {
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
        dbForEachLayerShapeShape $ls s {
            printOneShape $s
        }
    } elseif {[dbIsLayerShapeVia $ls]} {
        puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
        [dbLayerShapeViaViaLoc $ls]"
    } else {
    }
}
```

```
puts "*** error *** layerShape"  
}  
}
```

dbIsLayerShapeVia

`dbIsLayerShapeVia shapePtr`

Reports whether the specified address is a via cell. The command returns `1` if the shape is a via cell and `0` if it is not.

Parameters

<code>shapePtr</code>	Shape address
-----------------------	---------------

Command Order

Use this command after importing the design.

Example

```
proc printOneLayerShape {ls} {  
    if {[dbIsLayerShapeShapeList $ls]} {  
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"  
        dbForEachLayerShapeShape $ls s {  
            printOneShape $s  
        }  
    } elseif {[dbIsLayerShapeVia $ls]} {
```

```
puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc  
[dbLayerShapeViaViaLoc $ls]"  
} else {  
puts "*** error *** layerShape"  
}  
}
```

dbIsLayerV

`dbIsLayerV layerPtr`

Reports whether the specified layer is vertical. The command returns `1` if the layer is a vertical layer and `0` if it is not.

Parameters

<code>layerPtr</code>	Address of the layer.
-----------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 5> dbIsLayerV [dbGetLayerByName METAL2]
```

```
1
```

The layer `METAL1` is vertical.

dbIsLefPortClassCore

```
dbIsLefPortClassCore lefPortPtr
```

Reports whether the LEF port specified is of CLASS CORE. The command returns 1 if the LEF port is of CLASS CORE and 0 if it is not.

Parameters

<i>lefPortPtr</i>	Address of the LEF port.
-------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
proc printOneLefPort {p} {
if {[dbIsLefPortClassNone $p]} {
puts "lefPort class <none>"
} elseif {[dbIsLefPortClassCore $p]} {
puts "lefPort class <core>"
} elseif {[dbIsLefPortClassUndefined $p]} {
puts "lefPort class <undefined>"
} else {
puts "*** error *** lefPort class"
}
}
```

```
proc printOneFTerm {ft} {  
    puts "fterm [dbFTermName $ft]"  
    dbForEachFTermLefPort $ft p {  
        printOneLefPort $p  
    }  
}
```

The above script reports whether an f-term LEF port has a specified CLASS.

dbIsLefPortClassNone

```
dbIsLefPortClassNone lefPortPtr
```

Reports whether the LEF port specified is of CLASS NONE. The command returns 1 if the LEF port is of CLASS NONE and 0 if it is not.

Parameters

<i>lefPortPtr</i>	Address of the LEF port.
-------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
proc printOneLefPort {p} {  
    if {[dbIsLefPortClassNone $p]} {  
        puts "lefPort class <none>"
```

```
} elseif {[dbIsLefPortClassCore $p]} {  
    puts "lefPort class <core>"  
}  
elseif {[dbIsLefPortClassUndefined $p]} {  
    puts "lefPort class <undefined>"  
}  
else {  
    puts "*** error *** lefPort class"  
}  
}  
}  
  
proc printOneFTerm {ft} {  
    puts "fterm [dbFTermName $ft]"  
    dbForEachFTermLefPort $ft p {  
        printOneLefPort $p  
    }  
}
```

The above script reports whether an f-term LEF port has a specified CLASS.

dbIsLefPortClassUndefined

`dbIsLefPortClassUndefined lefPortPtr`

Reports whether the specified LEF port has a specified CLASS. The command returns 1 if the LEF port has a specified CLASS and a 0 if it does not.

Parameters

lefPortPtr

Address of the LEF port.

Command Order

Use this command after importing the design.

Example

```
proc printOneLefPort {p} {
    if {[dbIsLefPortClassNone $p]} {
        puts "lefPort class <none>"
    } elseif {[dbIsLefPortClassCore $p]} {
        puts "lefPort class <core>"
    } elseif {[dbIsLefPortClassUndefined $p]} {
        puts "lefPort class <undefined>"
    } else {
        puts "*** error *** lefPort class"
    }
}

proc printOneFTerm {ft} {
    puts "fterm [dbFTermName $ft]"
    dbForEachFTermLefPort $ft p {
        printOneLefPort $p
    }
}
```

The above script reports whether an f-term LEF port has a specified CLASS.

dbIsNetAnalog

`dbIsNetAnalog netPtr`

Reports whether the specified net is an analog net. The command returns a 1 if net is analog, and a 0 if it is not.

Parameters

<code><i>netPtr</i></code>	Address of the net.
----------------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1>dbIsNetAnalog $netPtr  
0
```

The net is not an analog net.

dbIsNetBus

`dbIsNetBus netPtr`

Reports whether the specified net is part of a bus. The command returns a 1 if net is part of a bus and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> dbIsNetBus $netPtr  
0
```

The net is not a bus.

dbIsNetClock

```
dbIsNetClock netPtr
```

Reports whether the specified net is a clock. The command returns a 1 if the net is a clock and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 164> dbIsNetClock [dbGetNetByName MCK]  
1
```

The net is a clock net.

dbIsNetConnected

`dbIsNetConnected netPtr`

Returns a list of unconnected terminals for the specified net.

Parameters

<code><i>netPtr</i></code>	Address of the net.
----------------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 47> set s [dbHeadSelList]  
0x96ddcb0  
innovus 48> set objPtr [dbSelPtr $s]  
0x7a6ba04  
innovus 49> dbIsNetConnected $objPtr  
Unconnected term at (0 0 0) of net LPB<0>  
Unconnected term at (1407650 1430900 0) of net LPB<0>
```

There are 2 unconnected terms for the selected net.

dbIsNetContAssignLHS

dbIsNetContAssignLHS *netPtr*

Checks whether a net is coming from assign statement's "right"-hand-side, where the associated terms are moved to net in "left"-hand-side.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetContAssignLHS $netPtr  
0
```

The net is not part of an assign statement.

dbIsNetContinuousAssign

dbIsNetContinuousAssign *netPtr*

Returns a 1 if the specified net is part of a continuous assign, else a 0.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after design import.

Example

```
innovus 165> dbIsNetContinuousAssign [dbGetNetByName BLU]  
0
```

The net is a continuous assign.

dbIsNetCritical

```
dbIsNetCritical netPtr
```

Reports whether the specified net is critical. The command returns a 1 if the net is critical and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 52> dbIsNetCritical $objPtr  
0
```

The net is not critical.

dbIsNetDblWidth

`dbIsNetDblWidth netPtr`

Reports whether the specified net is double width. The command returns a `1` if the net is double width and a `0` if it is not.

Parameters

<code>netPtr</code>	Address of the net.
---------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 53> dbIsNetDblWidth $objPtr  
0
```

The net is not double width.

dbIsNetDefInClock

`dbIsNetDefInClock netPtr`

Reports whether the specified net is clock from DEF. The command returns a `1` if the net is a clock from DEF and a `0` if it is not.

Parameters

<code>netPtr</code>	Address of the net.
---------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 53> dbIsNetDefInClock $objPtr
0
```

The net is not defined in DEF as a clock.

dbIsNetDontTouch

`dbIsNetDontTouch netPtr`

Reports whether the specified net is a `dont_touch`. The command returns a `1` if the net is a `dont_touch` and a `0` if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 53> dbIsNetDontTouch $objPtr  
0
```

The net is not a dont_touch.

dbIsNetExternal

```
dbIsNetExternal netPtr
```

Reports whether the specified net is external. The command returns a 1 if the net is external and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 54> dbIsNetExternal $objPtr
```

0

The net is not external.

dbIsNetGnd

`dbIsNetGnd netPtr`

Reports whether the specified net is a ground net. The command returns a `1` if the net is a ground net and a `0` if it is not.

Parameters

<code>netPtr</code>	Address of the net.
---------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 55> dbIsNetGnd $objPtr
```

0

The net is not ground.

dbIsNetGroupMsBuss

`dbIsNetGroupMsBuss netGroupPtr`

Reports whether the specified net group is a mixed signal buss. The command returns a `1` if the net group is a mixed signal buss and a `0` if it is not.

Parameters

<i>netGroupPtr</i>	Address of the net group
--------------------	--------------------------

Command Order

Use this command after importing the design, but it is more relevant after net groups are created.

Example

```
innovus 55> dbIsNetGroupMsBuss $objPtr  
0
```

The net group is not a mixed signal buss.

dbIsNetGroupMsCoax

```
dbIsNetGroupMsCoax netGroupPtr
```

Reports whether the specified net group is a mixed signal coax. The command returns a `1` if the net group is a mixed signal coax and a `0` if it is not.

Parameters

<i>netGroupPtr</i>	Address of the net group
--------------------	--------------------------

Command Order

Use this command after importing the design, but it is more relevant after net groups are created.

Example

```
innovus 55> dbIsNetGroupMsCoax $objPtr  
0
```

The net group is not a mixed signal coax.

dbIsNetGroupMsDiffPair

`dbIsNetGroupMsDiffPair netGroupPtr`

Reports whether the specified net group is a mixed signal differential pair. The command returns a `1` if the net group is a mixed signal differential pair and a `0` if it is not.

Parameters

<code>netGroupPtr</code>	Address of the net group
--------------------------	--------------------------

Command Order

Use this command after importing the design, but it is more relevant after net groups are created.

Example

```
innovus 55> dbIsNetGroupMsDiffPair $objPtr  
1
```

The net group is a mixed signal differential pair.

dbIsNetGroupMsMatchedPairs

dbIsNetGroupMsMatchedPairs *netGroupPtr*

Reports whether the specified net group is a mixed signal matched pair. The command returns a 1 if the net group is a mixed signal matched pair and a 0 if it is not.

Parameters

<i>netGroupPtr</i>	Address of the net group
--------------------	--------------------------

Command Order

Use this command after importing the design, but it is more relevant after net groups are created.

Example

```
innovus 55> dbIsNetGroupMsMatchedPairs $objPtr
0
```

The net group is not a mixed signal matched pair.

dbIsNetHidden

dbIsNetHidden *netPtr*

Reports whether the specified net is hidden. The command returns 1 if the net is hidden and 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 56> dbIsNetHidden [dbGetNetByName BLU]  
1
```

The net is hidden.

dbIsNetHilite

```
dbIsNetHilite netPtr
```

Reports whether the specified net is highlighted. The command returns `1` if the net is highlighted and `0` if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 56> dbIsNetHilite $objPtr
```

1

The net is highlighted.

dbIsNetIgnored

```
dbIsNetIgnored netPtr
```

Reports whether the net is ignored. The command returns a 1 if the net has been ignored and a 0 if it has not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
encounter 57> dbIsNetIgnored $objPtr
0
```

The net is not ignored.

dbIsNetInHInst

```
dbIsNetInHInst netPtr
```

Reports whether the specified net is in hierarchical instance. The command returns a 1 if the net is

in part of a hierarchical instance and a 0 if it has not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 57> dbIsNetInHInst $objPtr
```

```
0
```

The net is not part of a hierarchical instance.

dbIsNetIPOed

```
dbIsNetIPOed netPtr
```

Reports whether in-place optimization (IPO) has been run for the specified net. The command returns a 1 if the net has been optimized and a 0 if it has not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 57> dbIsNetIPOed $objPtr  
0
```

The net has not been IPOed.

dbIsNetIPOIgnored

`dbIsNetIPOIgnored netPtr`

Reports whether the specified net is ignored during timing optimization operations. The command returns `1` if the net is ignored and `0` if it is not.

Parameters

<code>netPtr</code>	Address of the net.
---------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 57> dbIsNetIPOIgnored $objPtr  
0
```

The net is not ignored during timing optimization operations.

dbIsNetMarked

`dbIsNetMarked netPtr`

Reports whether the specified net is marked. The command returns a 1 if the net is marked and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetMarked $objPtr
```

Returns:

0

The net is not marked.

dbIsNetMarked2

```
dbIsNetMarked2 netPtr
```

Returns a 1 if net is marked 2, else a 0.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after design import.

Example

```
dbIsNetMarked2 [dbGetNetByName BLU]
```

Returns:

0

The net is not marked 2.

dbIsNetMarked3

```
dbIsNetMarked3 netPtr
```

Returns a 1 if net is marked 3, else a 0.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after design import.

Example

```
dbIsNetMarked3 [dbGetNetByName BLU]
```

Returns:

0

The net is not marked 3.

dbIsNetMarked4

dbIsNetMarked4 *netPtr*

Returns a 1 if net is marked 4, else a 0.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after design import.

Example

```
dbIsNetMarked4 [dbGetNetByName BLU]
```

Returns:

0

The net is not marked 4.

dbIsNetPhysicalOnly

dbIsNetPhysicalOnly *netPtr*

Reports whether the specified net is physical only. The command returns a 1 if the net is physical only and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetPhysicalOnly $objPtr
```

Returns:

0

The net is not a physical only net.

dbIsNetPostRouteSiFix

```
dbIsNetPostRouteSiFix netPtr
```

Reports whether the specified net is a post routed SI net. The command returns a 1 if the net is a post routed SI net and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetPostRouteSiFix $objPtr
```

Returns:

0

The net is not SI post routed.

dbIsNetPwr

```
dbIsNetPwr netPtr
```

Reports whether the specified net is a power net. The command returns a 1 if the net is a power net and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetPwr $objPtr
```

Returns:

0

The net is not a power net.

dbIsNetPwrOrGnd

```
dbIsNetPwrOrGnd netPtr
```

Reports whether the specified net is a power net or a ground net. The command returns a 1 if the net is a power or ground net and a 0 if it is neither.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetPwrOrGnd $objPtr
```

Returns:

0

The net is not power or ground.

dbIsNetRectActive

```
dbIsNetRectActive netrectPtr
```

Returns a 1 if the specified net rectangle is active, else a 0.

Parameters

<i>netrectPtr</i>	Address of net rectangle.
-------------------	---------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectActive $objPtr
```

Returns:

1

The net rectangle is active.

dbIsNetRectHilite

```
dbIsNetRectHilite netrectPtr
```

Reports whether the specified net rectangle is highlighted. The command returns 1 if the net rectangle is highlighted and 0 if it is not.

Parameters

<i>netrectPtr</i>	Address of net rectangle.
-------------------	---------------------------

Command Order

Use this command after power planning.

Example

```
dbIsNetRectHilite $objPtr
```

Returns:

1

The net rectangle is highlighted.

dbIsNetRectHookup

`dbIsNetRectHookup netrectPtr`

Returns a 1 if the specified net rectangle is hooked up, else a 0.

Parameters

<code>netrectPtr</code>	Address of net rectangle.
-------------------------	---------------------------

Command Order

Use this command after power analysis.

Example

`dbIsNetRectHookup $objPtr`

Returns:

1

The net rectangle is hooked up.

dbIsNetRectMacroPGPin

`dbIsNetRectMacroPGPin netrectPtr`

Returns a 1 if the specified net rectangle is a macro PG pin, else a 0.

Parameters

<i>netrectPtr</i>	Address of net rectangle.
-------------------	---------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectMacroPGPin $objPtr
```

Returns:

1

The net rectangle is a macro PG pin.

dbIsNetRectMacroPGTerm

```
dbIsNetRectMacroPGTerm netrectPtr
```

Returns a 1 if the specified net rectangle is a macro PG term, else a 0.

Parameters

<i>netrectPtr</i>	Address of net rectangle.
-------------------	---------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectMacroPGTerm $objPtr
```

Returns:

1

The net rectangle is a macro PG term.

dbIsNetRectMarked

```
dbIsNetRectMarked netrectPtr
```

Returns a 1 if the specified net rectangle is marked, else a 0.

Parameters

<i>netrectPtr</i>	Address of net rectangle.
-------------------	---------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectMarked $objPtr
```

Returns:

1

The net rectangle is marked.

dbIsNetRectPolygon

`dbIsNetRectPolygon netrectPtr`

Reports whether the specified net rectangle is a polygon. The command returns 1 if it is a polygon and 0 if it is not.

Parameters

<code>netrectPtr</code>	Address of net rectangle.
-------------------------	---------------------------

Command Order

Use this command after power analysis.

Example

`dbIsNetRectRail $objPtr`

Returns:

1

The net rectangle is a polygon.

dbIsNetRectRail

`dbIsNetRectRail netrectPtr`

Returns a 1 if the specified net rectangle is a rail, else a 0.

Parameters

<code>netrectPtr</code>	Address of net rectangle.
-------------------------	---------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectRail $objPtr
```

Returns:

1

The net rectangle is a rail.

dbIsNetRectReference

```
dbIsNetRectReference netrectPtr
```

Returns a 1 if the specified net rectangle is referenced.

Parameters

<i>netrectPtr</i>	Address of net rectangle.
-------------------	---------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectReference $objPtr
```

Returns:

1

The net rectangle is referenced.

dbIsNetRectSel

dbIsNetRectSel *netrectPtr*

Returns a 1 if the specified net rectangle is selected, else a 0.

Parameters

<i>netrectPtr</i>	Address of net rectangle
-------------------	--------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectSel $objPtr
```

Returns:

0

The net rectangle is not selected.

dbIsNetRectSignal

dbIsNetRectSignal *netrectPtr*

Returns a 1 if the specified net rectangle is a signal, else a 0.

Parameters

<i>netrectPtr</i>	Address of net rectangle.
-------------------	---------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectSignal $objPtr
```

Returns:

1

The net rectangle is a signal.

dbIsNetRectStdCellPGTerm

```
dbIsNetRectStdCellPGTerm netrectPtr
```

Returns a 1 if the specified net rectangle is a standard cell P/G term, else a 0.

Parameters

<i>netrectPtr</i>	Address of net rect.angle
-------------------	---------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectStdCellPGTerm $objPtr
```

Returns:

1

The net rectangle is a standard cell PG term.

dbIsNetRectTrunk

`dbIsNetRectTrunk netrectPtr`

Returns a 1 if the specified net rectangle is a trunk, else a 0.

Parameters

<code>netrectPtr</code>	Address of net rectangle.
-------------------------	---------------------------

Command Order

Use this command after power analysis.

Example

`dbIsNetRectTrunk $objPrt`

Returns:

1

The net rectangle is a trunk.

dbIsNetRectVia

`dbIsNetRectVia netrectPtr`

Returns a 1 if the specified net rectangle is a via, else a 0.

Parameters

<i>netrectPtr</i>	Address of net rectangle.
-------------------	---------------------------

Command Order

Use this command after power analysis.

Example

```
dbIsNetRectVia $objPtr
```

Returns:

1

The net rectangle is a via.

dbIsNetRouteGuided

```
dbIsNetRouteGuided netPtr
```

Reports whether the specified net is associated with a route guide. The command returns a 1 if the net is associated with a route guide and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetRouteGuided $objPtr
```

Returns:

0

The net is not part of a route guide.

dbIsNetScanNet

```
dbIsNetScanNet netPtr
```

Reports whether the specified net is a scan net. The command returns a 1 if the net is a scan net and a 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetScanNet $objPtr
```

Returns:

0

The net is not a scan net.

dbIsNetSel

```
dbIsNetSel netPtr
```

Reports whether the specified net is selected. The command returns a `1` if the net is selected and a `0` if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetSel $objPtr
```

Returns:

`1`

The net is selected.

dbIsNetSkipRouting

```
dbIsNetSkipRouting netPtr
```

Reports whether the specified net will be skipped during routing. The command returns a `1` if the net will be skipped and a `0` if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetSkipRouting $objPtr
```

Returns:

0

The net is not skipped during routing.

dbIsNetSpecial

```
dbIsNetSpecial netPtr
```

Reports whether the specified net is a special net. The command returns 1 if the net is a special net and 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetSpecial $objPtr
```

Returns:

0

The net is not special.

dbIsNetTAIgnored

```
dbIsNetTAIgnored netPtr
```

Reports whether the specified net is ignored during timing analysis. The command returns 1 if the net is ignored and 0 if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetTAIgnored [dbGetNetByName n_7862]
```

Returns:

0

The net `n_7862` is not ignored during timing analysis.

dbIsNetTri

```
dbIsNetTri netPtr
```

Reports whether the specified net is a tristate net. The command returns a `1` if the net is a tristate net and a `0` if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetTri $objPtr
```

Returns:

`0`

The net is not a tristate net.

dbIsNetTrialRouted

```
dbIsNetTrialRouted netPtr
```

Reports whether the specified net is a trial routed. The command returns a `1` if the net is trial routed and a `0` if it is not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after routing the design.

Example

```
dbIsNetTrialRouted $objPtr
```

Returns:

0

The net has not been trial routed.

dbIsNetWideWidth

```
dbIsNetWideWidth netPtr
```

Reports whether the specified net has a wide width. The command returns a 1 if the net has a wide width and a 0 if it does not.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbIsNetWideWidth $objPtr
```

Returns:

0

The net does not have wide width.

dbIsObjAPin

```
dbIsObjAPin objPtr
```

Reports whether the specified object is an access pin. The command returns a 1 if the object is an access pin and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjAPin $objPtr
```

Returns:

1

The object is an access pin.

dbIsObjCell

`dbIsObjCell objPtr`

Reports whether the specified object is a cell. The command returns a 1 if the object is a cell and a 0 if it is not.

Parameters

<code><i>objPtr</i></code>	Address of the object.
----------------------------	------------------------

Command Order

Use this command after importing the design.

Example

`dbIsObjCell $objPtr`

Returns:

0

The object is not a cell.

dbIsObjCellPad

`dbIsObjCellPad objPtr`

Reports whether the specified object is a pad. The command returns a 1 if the object is a pad and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjCellPad $objPtr
```

Returns:

0

The object is not a pad.

dbIsObjConstraint

```
dbIsObjConstraint objPtr
```

Reports whether the specified object is a constraint. The command returns a 1 if the object is a constraint and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjConstraint $objPtr
```

Returns:

0

The object is not a constraint.

dbIsObjFTerm

```
dbIsObjFTerm objPtr
```

Reports whether the specified object is an f-term. The command returns a 1 if the object is an f-term and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjFTerm $objPtr
```

Returns:

0

The object is not an f-term.

dbIsObjHead

```
dbIsObjHead objPtr
```

Reports whether the specified object is the head. The command returns a `1` if the object is the head and a `0` if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjHead $objPtr
```

Returns:

`0`

The object is not the head.

dbIsObjHilite

```
dbIsObjHilite objPtr
```

Reports whether the specified object is a highlighted. The command returns `1` if the object is highlighted and `0` if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

dbIsObjHInst

`dbIsObjHInst objPtr`

Reports whether the specified object is a hierarchical instance. The command returns a 1 if the object is a hierarchical instance and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

`dbIsObjHInst $objPtr`

Returns:

0

The object is not a module.

dbIsObjInst

`dbIsObjInst objPtr`

Reports whether the specified object is an instance. The command returns a `1` if the object is an instance and a `0` if it is not.

Parameters

<code><i>objPtr</i></code>	Address of the object.
----------------------------	------------------------

Command Order

Use this command after importing the design.

Example

`dbIsObjInst $objPtr`

Returns:

`0`

The object is not an instance.

dbIsObjLayerBlk

`dbIsObjLayerBlk objPtr`

Reports whether the specified object is a routing blockage. The command returns a `1` if the object is a routing blockage and a `0` if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjLayerBlk $objPtr
```

Returns:

1

The object is a layer blockage.

dbIsObjNet

```
dbIsObjNet objPtr
```

Reports whether the specified object is a net. The command returns a 1 if the object is a net and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjNet $objPtr
```

Returns:

1

The object is a net.

dbIsObjNetRect

```
dbIsObjNetRect objPtr
```

Returns a 1 if the specified object is a net rectangle, else a 0.

Parameters

<i>objPtr</i>	Address of object.
---------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsObjNetRect $objPtr
```

Returns:

1

The object is a net rectangle.

dbIsObjObstruct

`dbIsObjObstruct objPtr`

Reports whether the specified object is a placement obstruction. The command returns a 1 if the object is a placement obstruction and a 0 if it is not.

Parameters

<code>objPtr</code>	Address of the object.
---------------------	------------------------

Command Order

Use this command after importing the design

Example

`dbIsObjObstruct $objPtr`

Returns:

0

The object is not an obstruction.

dbIsObjPath

`dbIsObjPath objPtr`

Returns a 1 if specified object is a path, else a 0.

Parameters

<code>objPtr</code>	Address of object.
---------------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsObjPath $objPtr
```

Returns:

0

The object is not a path.

dbIsObjPowerSink

```
dbIsObjPowerSink objPtr
```

Reports whether the specified object is a power sink. The command returns a 1 if the object is a power sink and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjPowerSink $objPtr
```

Returns:

0

The object is not a power sink.

dbIsObjPtnCut

`dbIsObjPtnCut objPtr`

Reports whether the specified object is a partition cut. The command returns a 1 if the object is a partition cut and a 0 if it is not.

Parameters

<code><i>objPtr</i></code>	Address of the object.
----------------------------	------------------------

Command Order

Use this command after importing the design.

Example

`dbIsObjPtnCut $objPtr`

Returns:

0

The object is a partition cut.

dbIsObjPtnFeed

`dbIsObjPtnFeed objPtr`

Reports whether the specified object is a partition feed. The command returns a 1 if the object is a partition feed and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjPtnFeed $objPtr
```

Returns:

0

The object is not a partition feed.

dbIsObjPtnPinBlk

```
dbIsObjPtnPinBlk objPtr
```

Reports whether the specified object is a partition pin blockage. The command returns a 1 if the object is a partition pin blockage and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjPtnPinBlk $objPtr
```

Returns:

0

The object is not a partition pin blockage.

dbIsObjRaShape

```
dbIsObjRaShape objPtr
```

Reports whether the specified object is a rail analysis shape. The command returns 1 if it is a rail analysis shape and 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjRaShape $objPtr
```

Returns:

0

The object is not a rail analysis shape.

dbIsObjRouteBlk

`dbIsObjRouteBlk objPtr`

Reports whether the specified object is a routing blockage. The command returns a `1` if the object is a routing blockage and a `0` if it is not.

Parameters

<code><i>objPtr</i></code>	Address of the object.
----------------------------	------------------------

Command Order

Use this command after importing the design.

Example

`dbIsObjRouteBlk $objPtr`

Returns:

`1`

The object is a routing blockage.

dbIsObjRow

`dbIsObjRow objPtr`

Reports whether the specified object is a row. The command returns a `1` if the object is a row and a `0` if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjRow $objPtr
```

Returns:

0

The object is not a row.

dbIsObjScreen

```
dbIsObjScreen objPtr
```

Reports whether the specified object is a density screen. The command returns a 1 if the object is a density screen and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjScreen $objPtr
```

Returns:

0

The object is not a screen.

dbIsObjSelected

```
dbIsObjSelected objPtr
```

Reports whether the specified object is selected. The command returns a 1 if the object is selected and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjSelected $objPtr
```

Returns:

0

The object is not selected.

dbIsObjSNet

dbIsObjSNet *objPtr*

Returns a 1 if the specified object is a special net, else a 0.

Parameters

<i>objPtr</i>	Address of object.
---------------	--------------------

Command Order

Use this command after design import.

Example

dbIsObjSNet \$objPtr

Returns:

0

The object is not a special net.

dbIsObjStdRow

dbIsObjStdRow *objPtr*

Reports whether the specified object is a standard row. The command returns a 1 if the object is a standard row and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjStdRow $objPtr
```

Returns:

0

The object is not a standard row.

dbIsObjStrip

```
dbIsObjStrip objPtr
```

Returns a 1 if the specified object is a power/ground strap, else a 0.

Parameters

<i>objPtr</i>	Address of object.
---------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsObjStrip $objPtr
```

Returns:

0

The object is not a strip box.

dbIsObjStripBox

`dbIsObjStripBox objPtr`

Reports whether the specified object is a strip box. The command returns a 1 if the object is a strip box and a 0 if it is not.

Parameters

<code><i>objPtr</i></code>	Address of the object.
----------------------------	------------------------

Command Order

Use this command after importing the design.

Example

`dbIsObjStripBox $objPtr`

Returns:

0

The object is not a strip box.

dbIsObjTerm

`dbIsObjTerm objPtr`

Reports whether the specified object is a terminal. The command returns a 1 if the object is a terminal and a 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjTerm $objPtr
```

Returns:

0

The object is not a term.

dbIsObjWire

```
dbIsObjWire objPtr
```

Reports whether the specified object is a wire. The command returns 1 if the object is a wire and 0 if it is not.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObjWire $objPtr
```

Returns:

0

The object is not a wire.

dbIsObjXPin

```
dbIsObjXPin objPtr
```

Returns a 1 if the specified object is a duplicate pin, else a 0.

Parameters

<i>objPtr</i>	Address of object.
---------------	--------------------

Command Order

Use this command after design import.

Example

```
dbIsObjXPin $objPtr
```

Returns:

0

The object is not a duplicate pin.

dbIsObstructHilite

```
dbIsObstructHilite obstructPtr
```

Reports whether the specified obstruction is highlighted. The command returns `1` if the obstruction is highlighted and `0` if it is not.

Parameters

<i>obstructPtr</i>	Address of the obstruction.
--------------------	-----------------------------

Command Order

Use this command after importing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96ddcb0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0x96d5b08
```

```
dbIsObstructHilite $objPtr
```

Returns:

```
1
```

The obstruction is highlighted.

dbIsObstructSel

```
dbIsObstructSel obstructPtr
```

Reports whether the specified obstruction is selected. The command returns a 1 if the obstruction is selected and a 0 if it is not.

Parameters

<i>obstructPtr</i>	Address of the obstruction.
--------------------	-----------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsObstructSel $objPtr
```

Returns:

1

The object is selected.

dbIsPathFalse

```
dbIsPathFalse pathPtr
```

Returns a 1 if the specified path is false, else a 0.

Parameters

<i>pathPtr</i>	Address of path.
----------------	------------------

Command Order

Use this command after timing graph has been built.

Example

```
dbIsPathFalse $pathPtr
```

Returns:

1

The path is false.

dbIsPathHilite

```
dbIsPathHilite pathPtr
```

Reports whether the specified path is highlighted. The command returns 1 if the path is highlighted and 0 if it is not.

Parameters

<i>pathPtr</i>	Address of path.
----------------	------------------

Command Order

Use this command after building the timing graph.

Example

```
dbIsPathHilite $pathPtr
```

Returns:

0

The path is not highlighted.

dbIsPathSel

`dbIsPathSel pathPtr`

Returns a 1 if the specified path is selected, else a 0.

Parameters

<code><i>pathPtr</i></code>	Address of path.
-----------------------------	------------------

Command Order

Use this command after timing graph has been built.

Example

`dbIsPathSel $pathPtr`

Returns:

0

The path is not selected.

dbIsPerimViewable

`dbIsPerimViewable perimPtr`

Returns a 1 if the specified perimeter is viewable, else a 0.

Parameters

<i>perimPtr</i>	Address of perimeter.
-----------------	-----------------------

Command Order

Use this command after design import.

Example

```
dbIsPerimViewable $objPtr
```

Returns:

1

The perimeter is viewable.

dbIsPowerDomainDefault

dbIsPowerDomainDefault

```
dbIsPowerDomainDefault powerDomainPtr
```

Reports whether the specified power domain is the default power domain. The command returns 1 if the power domain is the default and 0 if it is not.

Parameters

<i>powerDomainPtr</i>	Address of hierarchical instance.
-----------------------	-----------------------------------

Command Order

Use this command after importing the design.

dbIsPowerSinkEastCut

`dbIsPowerSinkEastCut powersinkPtr`

Returns a 1 if the specified power sink is an east cut, else a 0.

Parameters

<code>powersinkPtr</code>	Address of power sink.
---------------------------	------------------------

Command Order

Use this command after specifying power pad locations.

Example

`dbIsPowerSinkEastCut $objPtr`

Returns:

1

The power sink is an east cut.

dbIsPowerSinkHilite

`dbIsPowerSinkHilite powersinkPtr`

Reports whether the specified power sink is highlighted. The command returns 1 if the power sink is highlighted and 0 if it is not.

Parameters

<i>powersinkPtr</i>	Address of power sink.
---------------------	------------------------

Command Order

Use this command after defining power sinks.

Example

```
dbIsPowerSinkHilite $objPtr
```

Returns:

1

The power sink is highlighted.

dbIsPowerSinkMarked

```
dbIsPowerSinkMarked powersinkPtr
```

Returns a 1 if the specified power sink is marked, else a 0.

Parameters

<i>powersinkPtr</i>	Address of power sink.
---------------------	------------------------

Command Order

Use this command after defining power sinks.

Example

```
dbIsPowerSinkMarked $objPtr
```

Returns:

1

The power sink is marked.

dbIsPowerSinkNorthCut

```
dbIsPowerSinkNorthCut powersinkPtr
```

Returns a 1 if the specified power sink is a north cut, else a 0.

Parameters

<i>powersinkPtr</i>	Address of power sink.
---------------------	------------------------

Command Order

Use this command after specifying power pad locations.

Example

```
dbIsPowerSinkNorthCut $objPtr
```

Returns:

1

The power sink is a north cut.

dbIsPowerSinkSel

`dbIsPowerSinkSel powersinkPtr`

Returns a 1 if the specified power sink is selected, else a 0.

Parameters

<code>powersinkPtr</code>	Address of power sink.
---------------------------	------------------------

Command Order

Use this command after defining power sinks.

Example

`dbIsPowerSinkSel $objPtr`

Returns:

0

The power sink is not selected.

dbIsPowerSinkSouthCut

`dbIsPowerSinkSouthCut powersinkPtr`

Returns a 1 if the specified power sink is a south cut, else a 0.

Parameters

<code>powersinkPtr</code>	Address of power sink.
---------------------------	------------------------

Command Order

Use this command after specifying power pad locations.

Example

```
dbIsPowerSinkSouthCut $objPtr
```

Returns:

0

The power sink is not a south cut.

dbIsPowerSinkWestCut

```
dbIsPowerSinkWestCut powersinkPtr
```

Returns a 1 if the specified power sink is a west cut, else a 0.

Parameters

<i>powersinkPtr</i>	Address of power sink.
---------------------	------------------------

Command Order

Use this command after specifying power pad locations.

Example

```
dbIsPowerSinkWestCut $objPtr
```

Returns:

0

The power sink is not a west cut.

dbIsPropTypeRegistered

dbIsPropTypeRegistered *propTypePtr*

Returns a 1 if the specified property type is registered, else a 0.

Parameters

<i>propTypePtr</i>	Address of property.
--------------------	----------------------

Command Order

Use this command after design import and at least one property has been applied to one object.

Example

```
dbIsPropTypeRegistered WIRE_COUNT
```

Returns:

1

The property type, WIRE_COUNT, is registered.

dbIsPtnBlackBox

dbIsPtnBlackBox *ptnPtr*

Returns a 1 if the specified partition is a blackbox, else a 0.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after design import.

Example

```
dbIsPtnBlackBox [dbGetPtnByName sheet7]
```

Returns:

0

The partition is not a blackbox.

dbIsPtnCommit

```
dbIsPtnCommit ptnPtr
```

Returns a 1 if the specified partition is committed, else a 0.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after specifying partitions.

Example

```
dbIsPtnCommit [dbGetPtnByName sheet7]
```

Returns:

1

The specified partition is committed.

dbIsPtnCutHilite

`dbIsPtnCutHilite ptnCutPtr`

Reports whether the specified partition cut is highlighted. The command returns 1 if the partition cut is highlighted and 0 if it is not.

Parameters

<code><i>ptnCutPtr</i></code>	Address of the partition cut.
-------------------------------	-------------------------------

Command Order

Use this command after importing the design, specifying partitions, and creating at least one partition cut.

Example

`dbIsPtnCutHilite $objPtr`

Returns:

1

The partition cut is highlighted.

dbIsPtnCutSel

`dbIsPtnCutSel ptnCutPtr`

Reports whether the specified partition cut is selected. The command returns a 1 if the partition cut

is selected and a 0 if it is not.

Parameters

<i>ptnCutPtr</i>	Address of the partition cut.
------------------	-------------------------------

Command Order

Use this command after importing and partitioning the design and creating at least one partition cut.

Example

```
dbIsPtnCutSel $objPtr
```

Returns:

1

The partition cut is selected.

dbIsPtnFeedHilite

```
dbIsPtnFeedHilite ptnFeedPtr
```

Reports whether the specified partition feedthrough is highlighted. The command returns 1 if the partition feedthrough is highlighted and 0 if it is not.

Parameters

<i>ptnFeedPtr</i>	Address of partition feed.
-------------------	----------------------------

Command Order

Use this command after partitioning the design, and creating at least one partition feedthrough.

Example

```
set s [dbHeadSelList]
```

Returns:

0x96e09e0

```
set objPtr [dbSelPtr $s]
```

Returns:

0x9d2ff00

```
dbIsPtnFeedHilite $objPtr
```

Returns:

1

The partition feedthrough is highlighted.

dbIsPtnFeedOnLayer

```
dbIsPtnFeedOnLayer ptnFeedPtr Int
```

Returns a 1 if the specified partition feed is on specified layer, else a 0.

Parameters

<i>ptnFeedPtr</i>	Address of a partition.
<i>Int</i>	Metal layer.

Command Order

Use this command after partitioning and creating at least one partition feed through.

Example

```
dbIsPtnFeedOnLayer $objPtr 1
```

Returns:

0

The partition feed is not on a specific layer.

dbIsPtnFeedSel

```
dbIsPtnFeedSel ptnFeedPtr
```

Returns a 1 if the specified partition feed through is selected, else a 0.

Parameters

<i>ptnFeedPtr</i>	Address of partition feed through.
-------------------	------------------------------------

Command Order

Use this command after specifying partitions and creating at least one partition feed through.

Example

```
dbIsPtnFeedSel $objPtr
```

Returns:

0

The partition feed is not selected.

dbIsPtnInTimeShell

`dbIsPtnInTimeShell ptnPtr`

Returns a 1 if the specified partition has been budgeted, else a 0.

Parameters

<code><i>ptnPtr</i></code>	Address of partition.
----------------------------	-----------------------

Command Order

Use this command after build timing graph.

Example

`dbIsPtnInTimeShell [dbGetPtnByName SH17]`

Returns:

1

The partition `SH17` has been budgeted.

dbIsPtnLayerBlockedOnLayer

`dbIsPtnLayerBlockedOnLayer ptnPtr Int`

Reports whether the specified partition is blocked on the specified layer. The command returns a 1 if the partition is blocked and a 0 if it is not.

Parameters

<i>ptnPtr</i>	Address of the partition.
<i>Int</i>	Metal layer.

Command Order

Use this command after importing and partitioning the design.

Example

```
dbIsPtnLayerBlockedOnLayer [dbGetPtnByName sheet7] 3
```

Returns:

1

The partition `sheet7` is blocked on metal 3 layer.

dbIsPtnM1Blocked

```
dbIsPtnM1Blocked ptnPtr
```

Reports whether the specified partition has a *meta/1* blockage. The command returns a `1` if the partition has a blockage and a `0` if it does not.

Parameters

<i>ptnPtr</i>	Address of the partition.
---------------	---------------------------

Command Order

Use this command after importing and partitioning the design.

Example

```
dbIsPtnM1Blocked [dbGetPtnByName sheet7]
```

Returns:

1

The partition `sheet7` is blocked on metal 1.

dbIsPtnM2Blocked

```
dbIsPtnM2Blocked ptnPtr
```

Reports whether the specified partition has a *meta/2* blockage. The command returns a `1` if the partition has a blockage and a `0` if it does not.

Parameters

<code>ptnPtr</code>	Address of the partition.
---------------------	---------------------------

Command Order

Use this command after importing and partitioning the design.

Example

```
dbIsPtnM2Blocked [dbGetPtnByName sheet7]
```

Returns:

1

The partition `sheet7` is blocked on metal 2

dbIsPtnM3Blocked

dbIsPtnM3Blocked *ptnPtr*

Reports whether the specified partition has a *meta/3* blockage. The command returns a 1 if the partition has a blockage and a 0 if it does not.

Parameters

ptnPtr

Address of the partition.

Command Order

Use this command after importing and partitioning the design.

Example

```
dbIsPtnM3Blocked [dbGetPtnByName sheet7]
```

Returns:

1

The partition sheet7 is blocked on metal 3.

dbIsPtnM4Blocked

dbIsPtnM4Blocked *ptnPtr*

Reports whether the specified partition has a *meta/4* blockage. The command returns a 1 if the partition has a blockage and a 0 if it does not.

Parameters

ptnPtr

Address of the partition.

Command Order

Use this command after importing and partitioning the design.

Example

```
dbIsPtnM4Blocked [dbGetPtnByName sheet7]
```

Returns:

0

The partition `sheet7` is not blocked on metal 4.

dbIsPtnM5Blocked

```
dbIsPtnM5Blocked ptnPtr
```

Reports whether the specified partition has a *meta/5* blockage. The command returns a `1` if the partition has a blockage and a `0` if it does not.

Parameters

<code>ptnPtr</code>	Address of the partition.
---------------------	---------------------------

Command Order

Use this command after importing and partitioning the design.

Example

```
dbIsPtnM5Blocked [dbGetPtnByName sheet7]
```

Returns:

0

The partition `sheet7` is not blocked on metal 7.

dbIsPtnM6Blocked

`dbIsPtnM6Blocked ptnPtr`

Reports whether the specified partition has a *meta/6* blockage. The command returns a `1` if the partition has a blockage and a `0` if it does not.

Parameters

<code>ptnPtr</code>	Address of the partition.
---------------------	---------------------------

Command Order

Use this command after importing and partitioning the design.

Example

`dbIsPtnM6Blocked [dbGetPtnByName sheet7]`

Returns:

`0`

The partition `sheet7` is not blocked on metal 6.

dbIsPtnM7Blocked

`dbIsPtnM7Blocked ptnPtr`

Reports whether the specified partition has a *meta/7* blockage. The command returns a `1` if the partition has a blockage and a `0` if it does not.

Parameters

ptnPtr

Address of the partition.

Command Order

Use this command after importing and partitioning the design.

Example

```
dbIsPtnM7Blocked [dbGetPtnByName sheet7]
```

Returns:

0

The partition `sheet7` is not blocked on metal 7.

dbIsPtnM8Blocked

```
dbIsPtnM8Blocked ptnPtr
```

Reports whether the specified partition has a *meta/8* blockage. The command returns a `1` if the partition has a blockage and a `0` if it does not.

Parameters

ptnPtr

Address of the partition.

Command Order

Use this command after importing and partitioning the design.

Example

```
dbIsPtnM8Blocked [dbGetPtnByName sheet7]
```

Returns:

0

The partition sheet7 is not blocked on metal 8.

dbIsPtnM9Blocked

```
dbIsPtnM9Blocked ptnPtr
```

Reports whether the specified partition has a *meta/9* blockage. The command returns a 1 if the partition has a blockage and a 0 if it does not.

Parameters

<i>ptnPtr</i>	Address of the partition.
---------------	---------------------------

Command Order

Use this command after importing and partitioning the design.

Example

```
dbIsPtnM9Blocked [dbGetPtnByName sheet7]
```

Returns:

0

The partition sheet7 is not blocked on metal 9.

dbIsPtnPinBlkHilite

`dbIsPtnPinBlkHilite ptnPinBlkPtr`

Reports whether the specified partition pin blockage is highlighted. The command returns `1` if the partition pin blockage is highlighted and `0` if it is not.

Parameters

<code>ptnPinBlkPtr</code>	Address of the partition pin blockage.
---------------------------	--

Command Order

Use this command after importing and partitioning the design, and creating at least one partition pin blockage.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96e09e0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0xa063fe0
```

```
dbIsPtnPinBlkHilite $objPtr
```

Returns:

```
1
```

The partition pin blockage is highlighted.

dbIsPtnPinBlkOnLayer

`dbIsPtnPinBlkOnLayer ptnPinBlkPtr Int`

Reports whether the specified partition pin blockage is on the specified metal layer. The command returns a `1` if the partition pin blockage is on the layer and a `0` if it is not.

Parameters

<code>ptnPinBlkPtr</code>	Address of the partition pin blockage.
<code>Int</code>	Metal layer.

Command Order

Use this command after importing and partitioning the design and creating at least one partition pin blockage.

Example

`dbIsPtnPinBlkOnLayer $objPtr 3`

Returns:

`1`

The partition pin blockage is on metal 3.

dbIsPtnPinBlkSel

`dbIsPtnPinBlkSel ptnPinBlkPtr`

Reports whether the specified partition pin blockage is selected. The command returns a `1` if the

partition pin blockage is selected and a 0 if it is not.

Parameters

<i>ptnPinBlkPtr</i>	Address of the partition pin blockage.
---------------------	--

Command Order

Use this command after importing and partitioning the design and creating at least one partition pin blockage.

Example

```
dbIsPtnPinBlkSel $objPtr
```

Returns:

1

The partition pin blockage is selected.

dbIsPtnPinLayerFreeOnSideOnLayer

```
dbIsPtnPinLayerFreeOnSideOnLayer ptnPinBlkPtr Int Int
```

Returns a 1 if partition has layer free on the side specified by the first Int for the metal layer specified by the second Int, else returns a 0.

Parameters

<i>ptnPinBlkPtr</i>	Address of partition.
<i>Int</i>	

Command Order

Use this command after partitioning.

Example

```
dbIsPtnPinLayerFreeOnSideOnLayer [dbGetPtnByName sheet7] 3 3
```

Returns:

1

The partition is free on the south side for metal 3 for pin assignment.

dbIsRCDCoupling

```
dbIsRCDCoupling
```

Returns 1 if there is an RC database, and returns 0 if there is not.

Command Order

Use this command after extraction.

Example

```
dbIsRCDCoupling
```

Returns:

rc data are not available

0

There is no RC database.

dbIsRCDBFromFE

```
dbIsRCDBFromFE
```

Returns 1 if RC extraction is from Innovus, and 0 if it is not.

Command Order

Use this command after extraction.

Example

dbIsRCDBFromFE

Returns:

1

RC extraction is from Innovus.

dbIsRCDBSpcf

dbIsRCDBSpcf

Returns 1 if RC extraction is from a SPEF file and 0 if it is not.

Command Order

Use this command after extraction.

Example

dbIsRCDBSpcf

Returns:

0

RC extraction is not from SPEF.

dbIsRCDBSpcfUsingStarN

dbIsRCDBSpcfUsingStarN

Reports whether the RC extraction is from a Star SPEF file. The command returns `1` if the RC extraction is from a Star SPEF file and `0` if it is not.

Parameters

None

Command Order

Use this command after extraction.

Example

dbIsRCDBSpcfUsingStarN

Returns:

`0`

The RC extraction is not from a SPEF Star extraction.

dbIsResFromRCTbl

dbIsResFromRCTbl

Reports whether the resistance is from the RC table. Returns `1` if the resistance is from the RC table and `0` if it is not.

Parameters

None

Command Order

Use this command after importing the design.

Example

```
dbIsResFromRCTbl
```

Returns:

0

The resistance is not from the RC table.

dbIsRouteBlkHilite

```
dbIsRouteBlkHilite routeBlkPtr
```

Reports whether the specified routing layer blockage is highlighted. The command returns 1 if the routing layer blockage is highlighted and a 0 if it is not.

Parameters

<i>routeBlkPtr</i>	Address of the routing layer blockage.
--------------------	--

Command Order

Use this command after importing the design and creating at least one routing layer blockage.

Example

```
dbIsRouteBlkHilited $objPtr
```

Returns:

1

The routing blockage is highlighted.

dbIsRouteBlkSel

```
dbIsRouteBlkSel routeBlkPtr
```

Reports whether the specified routing layer blockage is selected. The command returns 1 if the routing layer blockage is selected and a 0 if it is not.

Parameters

<i>routeBlkPtr</i>	Address of the routing layer blockage.
--------------------	--

Command Order

Use this command after importing the design and creating at least one routing layer blockage.

Example

```
dbIsRouteBlkSel $objPtr
```

Returns:

1

The routing blockage is selected.

dbIsRouteBoxFeedthrough

```
dbIsRouteBoxFeedthrough routeBoxPtr
```

Reports whether the specified route box is a feedthrough. The command returns a 1 if the route box is a feedthrough and a 0 if it is not.

Parameters

<i>routeBoxPtr</i>	Address of the route box.
--------------------	---------------------------

Command Order

Use this command after importing and partitioning the design and creating at least one route box.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96e09e0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0x96d6890
```

```
set obj_type [dbObjType $objPtr]
```

```
innovus 231> dbIsRouteBoxFeedthrough $objPtr
```

Returns:

```
0
```

The route box is not a feedthrough.

dbIsRouteBoxHilite

```
dbIsRouteBoxHilite routeBoxPtr
```

Reports whether the specified route box is highlighted. The command returns `1` if the route box is highlighted and `0` if it is not.

Parameters

<code>routeBoxPtr</code>	Address of the route box.
--------------------------	---------------------------

Command Order

Use this command after importing the design and creating a route box.

Example

```
dbIsRouteBoxHilite $objPtr
```

Returns:

`1`

The route box is highlighted.

dbIsRouteBoxSel

```
dbIsRouteBoxSel routeBoxPtr
```

Reports whether the specified route box is selected. The command returns a `1` if the route box is selected and a `0` if it is not.

Parameters

<i>routeBoxPtr</i>	Address of the route box.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one route box.

Example

```
dbIsRouteBoxSel $objPtr
```

Returns:

1

The route box is selected.

dbIsRouteGuideHilite

```
dbIsRouteGuideHilite routeGuidePtr
```

Reports whether the specified route guide is highlighted. The command returns `1` if the route guide is highlighted and `0` if it is not.

Parameters

<i>routeGuidePtr</i>	Address of the route guide.
----------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
dbIsRouteGuideHilite $routeGuidePtr
```

Returns:

0

The route guide is not highlighted.

dbIsRouteGuideOptimizePin

```
dbIsRouteGuideOptimizePin routeGuidePtr
```

Route guides by default assign pins in the route box according to its order in the database "as is," when dbIsRouteGuideOptimizePin is turned on, ordering of the pins will be performed.

Parameters

<i>routeGuidePtr</i>	Address of the route guide.
----------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
dbIsRouteGuideOptimizePin $routeGuidePtr
```

Returns:

1

The route guide has been pin optimized.

dbIsRouteGuidePinBased

```
dbIsRouteGuidePinBased routeGuidePtr
```

Reports whether the specified route guide is pin based. The command returns a `1` if the route guide is pin based and a `0` if it is not.

Parameters

<i>routeGuidePtr</i>	Address of the route guide.
----------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
dbIsRouteGuidePinBased $rotueGuidePtr
```

Returns:

`1`

The route guide is pin based.

dbIsRouteGuideSel

```
dbIsRouteGuideSel routeGuidePtr
```

Reports whether the specified route guide is selected. The command returns a `1` if the route guide is selected and a `0` if it is not.

Parameters

<i>routeGuidePtr</i>	Address of the route guide.
----------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
dbIsRouteGuideSel $routeGuidePtr
```

Returns:

0

The route guide is selected.

dbIsRouteVia

```
dbIsRouteVia route
```

Reports whether the specified route is a via. The command returns a 1 if the route is a via and a 0 if it is not.

Parameters

<i>route</i>	Address of the route.
--------------	-----------------------

Command Order

Use this command after importing the design and creating at least one route.

Example

```
proc iter_routes_delete_vias {} {  
  
    set fout [open "IterRoutesRmVias.out" w]  
  
    puts $fout "Routes Info:  
"  
    puts $fout ""  
    puts $fout ""  
  
    dbForAllCellNet [dbgTopCell] net {  
  
        set nName [dbNetName $net]  
  
        dbIterAllRoutes  
  
        set iter [dbIterRoutes $net]  
  
        while {[set route [dbRouteNext $iter]] != {0x0}} {  
  
            puts $fout "Processing net -> $nName\n"  
  
            if { [dbIsRouteVia $route] } {  
  
                puts $fout " viaPtr -> $route"  
  
                set via_info [dbInfoVia $route]  
  
                puts $fout " via -> $via_info"  
  
                if { [ dbIsObjStripBox $route ] } {  
  
                    set rc [ dbDeleteStripBox $route ]  
  
                    puts $fout " Deleted the (strip box) via, $route, rc = $rc"  
                } else {  
  
                    set rc [ dbDeleteRouteBox $route ]  
  
                    puts $fout " Deleted the (route box) via, $route, rc = $rc"  
                }  
            }  
        }  
    }  
}
```

dbIsRouteWire

```
dbIsRouteWire route
```

Reports whether the specified route is a wire. The command returns a 1 if the route is a wire and a 0 if it is not.

Parameters

<i>route</i>	Route.
--------------	--------

Command Order

Use this command after importing the design and creating at least one route.

Example

```
proc iter_routes_delete_vias {} {  
    set fout [open "IterRoutesRmVias.out" w]  
  
    puts $fout "Routes Info:  
    puts $fout ""  
    puts $fout ""  
    dbForAllCellNet [dbgTopCell] net {  
        set nName [dbNetName $net]  
        dbIterAllRoutes  
        set iter [dbIterRoutes $net]  
        while {[set route [dbRouteNext $iter]] != {0x0}} {  
            puts $fout "Processing net -> $nName\n"  
            if { [dbIsRouteWire $route] } {  
                puts $fout " wirePtr -> $route"  
                set wire_info [dbInfoWire $route]  
                puts $fout " wire -> $wire_info"  
                if { [ dbIsObjStripBox $route ] } {
```

```
set rc [ dbDeleteStripBox $route ]  
puts $fout " Deleted the (strip box) wire, $route, rc = $rc"  
} else {  
set rc [ dbDeleteRouteBox $route ]  
puts $fout " Deleted the (route box) wire, $route, rc = $rc"  
}
```

dbIsRowAreaIo

dbIsRowAreaIo *rowPtr*

Returns a 1 if the specified row is an area I/O row, else a 0.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after design import.

Example

dbIsRowAreaIo \$rowPtr

Returns:

0

The row is not an area I/O row.

dbIsRowClusterHilite

```
dbIsRowClusterHilite ioslotPtr
```

Reports whether the specified row cluster is highlighted. The command returns `1` if the row cluster is highlighted and `0` if it is not.

Parameters

<i>ioslotPtr</i>	Address of row cluster.
------------------	-------------------------

Command Order

Use this command after design import.

Example

```
dbIsRowClusterHilite $ioslotPtr
```

Returns:

0

The row cluster is not highlighted.

dbIsRowClusterSel

```
dbIsRowClusterSel ioslotPtr
```

Returns a `1` if the specified row cluster is selected, else a `0`.

Parameters

<i>ioslotPtr</i>	Address of row cluster.
------------------	-------------------------

Command Order

Use this command after design import.

Example

```
dbIsRowClusterSel $ioslotPtr
```

Returns:

0

The row cluster is not selected.

dbIsRowFlipped

```
dbIsRowFlipped rowPtr
```

Returns a 1 if the specified row is flipped else a 0.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after design import.

Example

```
dbIsRowFlipped $rowPtr
```

Returns:

0

The row is not flipped.

dbIsRowHilite

`dbIsRowHilite rowPtr`

Reports whether the specified row is highlighted. The command returns 1 if the row is highlighted and 0 if it is not.

Parameters

<code>rowPtr</code>	Address of row.
---------------------	-----------------

Command Order

Use this command after design import.

Example

`dbIsRowHilite $rowPtr`

Returns:

0

The row is not highlighted.

dbIsRowMarked

`dbIsRowMarked rowPtr`

Returns a 1 if the specified row is marked, else a 0.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after design import.

Example

```
dbIsRowMarked $rowPtr
```

Returns:

0

The row is not marked.

dbIsRowSel

```
dbIsRowSel rowPtr
```

Returns a 1 if the specified row is selected, else a 0.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after design import.

Example

```
dbIsRowSel $rowPtr
```

Returns:

0

The row is not selected.

dbIsRulerFlip

```
dbIsRulerFlip rulerPtr
```

Returns a 1 if the specified ruler is flipped, else a 0.

Parameters

<i>rulerPtr</i>	Address of ruler.
-----------------	-------------------

Command Order

Use this command after design import and creating a ruler.

Example

```
set rulerPtr [dbCreateRuler 100 100]
```

Returns:

0x7305040

```
dbIsRulerFlip $rulerPtr
```

Returns:

0

The ruler is not flipped.

dbIsRulerHilite

```
dbIsRulerHilite rulerPtr
```

Reports whether the specified ruler is highlighted. The command returns `1` if the ruler is highlighted and `0` if it is not.

Parameters

<i>rulerPtr</i>	Address of the ruler.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
set rulerPtr [dbCreateRuler 100 100]
```

Returns:

```
0x7305040
```

```
dbIsRulerHilite $rulerPtr
```

Returns:

```
0
```

The ruler is not highlighted.

dbIsScreenSel

```
dbIsScreenSel screenPtr
```

Reports whether the specified screen is selected. The command returns a `1` if the screen is selected and a `0` if it is not.

Parameters

<i>screenPtr</i>	Address of the screen.
------------------	------------------------

Command Order

Use this command after Innovus is invoked.

Example

```
dbIsScreenSel $objPtr
```

Returns:

`1`

The screen is selected.

dbIsShapeTypePath

```
dbIsShapeTypePath shapePtr
```

Reports whether the specified shape is a path. The command returns `1` if the shape is a path and `0` if it is not.

Parameters

shapePtr

Address of the shape.

Command Order

Use this command after importing the design.

Example

```
proc printOneShape {s} {
    if {[dbIsShapeTypeRect $s]} {
        puts "shape type is rect [dbShapeRect $s]"
    } elseif {[dbIsShapeTypePath $s]} {
        puts "shape type is path"
    } elseif {[dbIsShapeTypePoly $s]} {
        puts "shape type is poly"
    } elseif {[dbIsShapeTypeUndefined $s]} {
        puts "shape type is undefined"
    } else {
        puts "*** error *** shape type"
    }
}

proc printOneLayerShape {ls} {
    if {[dbIsLayerShapeShapeList $ls]} {
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
        dbForEachLayerShapeShape $ls s {
            printOneShape $s
        }
    } elseif {[dbIsLayerShapeVia $ls]} {
        puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
[dbLayerShapeViaViaLoc $ls]"
    } else {

```

```
puts "*** error *** layerShape"  
} }
```

dbIsShapeTypePoly

`dbIsShapeTypePoly shapePtr`

Reports whether the specified shape is a polygon. The command returns 1 if the shape is a polygon and 0 if it is not.

Parameters

<code>shapePtr</code>	Address of the shape.
-----------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
proc printOneShape {s} {  
    if {[dbIsShapeTypeRect $s]} {  
        puts "shape type is rect [dbShapeRect $s]"  
    } elseif {[dbIsShapeTypePath $s]} {  
        puts "shape type is path"  
    } elseif {[dbIsShapeTypePoly $s]} {  
        puts "shape type is poly"  
    } elseif {[dbIsShapeTypeUndefined $s]} {  
        puts "shape type is undefined"  
    } else {
```

```
puts "*** error *** shape type"
}

proc printOneLayerShape {ls} {
if {[dbIsLayerShapeShapeList $ls]} {
puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
dbForEachLayerShapeShape $ls s {
printOneShape $s
}
} elseif {[dbIsLayerShapeVia $ls]} {
puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
[dbLayerShapeViaViaLoc $ls]"
} else {
puts "*** error *** layerShape"
}
```

dbIsShapeTypeRect

`dbIsShapeTypeRect shapePtr`

Reports whether the specified shape is a rectangle. The command returns `1` if the shape is a rectangle and `0` if it is not.

Parameters

<code><i>shapePtr</i></code>	Address of the shape.
------------------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
proc printOneShape {s} {
    if {[dbIsShapeTypeRect $s]} {
        puts "shape type is rect [dbShapeRect $s]"
    } elseif {[dbIsShapeTypePath $s]} {
        puts "shape type is path"
    } elseif {[dbIsShapeTypePoly $s]} {
        puts "shape type is poly"
    } elseif {[dbIsShapeTypeUndefined $s]} {
        puts "shape type is undefined"
    } else {
        puts "*** error *** shape type"
    }
}

proc printOneLayerShape {ls} {
    if {[dbIsLayerShapeShapeList $ls]} {
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
        dbForEachLayerShapeShape $ls s {
            printOneShape $s
        }
    } elseif {[dbIsLayerShapeVia $ls]} {
        puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
[dbLayerShapeViaViaLoc $ls]"
    } else {
        puts "*** error *** layerShape"
    }
}
```

dbIsShapeTypeUndefined

`dbIsShapeTypeUndefined shapePtr`

Reports whether the specified shape type is undefined. The command returns 1 if the shape type is undefined and 0 if it is defined.

Parameters

<i>shapePtr</i>	Address of the shape.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
proc printOneShape {s} {
    if {[dbIsShapeTypeRect $s]} {
        puts "shape type is rect [dbShapeRect $s]"
    } elseif {[dbIsShapeTypePath $s]} {
        puts "shape type is path"
    } elseif {[dbIsShapeTypePoly $s]} {
        puts "shape type is poly"
    } elseif {[dbIsShapeTypeUndefined $s]} {
        puts "shape type is undefined"
    } else {
        puts "*** error *** shape type"
    }
}

proc printOneLayerShape {ls} {
    if {[dbIsLayerShapeShapeList $ls]} {
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
        dbForEachLayerShapeShape $ls s {

```

```
printOneShape $s
}
} elseif {[dbIsLayerShapeVia $ls]} {
puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
[dbLayerShapeViaViaLoc $ls]"
} else {
puts "*** error *** layerShape"
```

dbIsSiteCore

`dbIsSiteCore sitePtr`

Reports whether the specified site is a core site. The command returns a 1 if the site is a core site and a 0 if it is not.

Parameters

<code>sitePtr</code>	Address of the site.
----------------------	----------------------

Command Order

Use this command after Innovus is invoked.

Example

```
dbForEachHeadSite [dbgHead] sitePtr {
set name [dbSiteName $sitePtr]
if { [dbIsSiteCore $sitePtr] } {
puts "Site $name is a core site"
}
}
```

Reports core sites.

dbIsStdRowArealo

`dbIsStdRowAreaIo stdRowPtr`

Reports whether the specified standard row is an area I/O. The command returns a `1` if the standard row is an area I/O and a `0` if it is not.

Parameters

<code>stdRowPtr</code>	Address of the standard row.
------------------------	------------------------------

Command Order

Use this command after importing the design.

Example

`dbIsStdRowAreaIo $stdRowPtr`

Returns:

`1`

The standard row is an area I/O.

dbIsStdRowFlipped

`dbIsStdRowFlipped stdRowPtr`

Reports whether the specified standard row is flipped. The command returns a `1` if the standard row is flipped and a `0` if it is not,

Parameters

<i>stdRowPtr</i>	Address of the standard row.
------------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsStdRowFlipped $stdRowPtr
```

Returns:

1

The standard row is flipped.

dbIsStdRowHilite

```
dbIsStdRowHilite stdRowPtr
```

Reports whether the specified standard row is highlighted. The command returns 1 if the standard row is highlighted and 0 if it is not.

Parameters

<i>stdRowPtr</i>	Address of the standard row.
------------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsStdRowHilite $stdRowPtr
```

Returns:

0

The standard row is highlighted.

dbIsStdRowMarked

```
dbIsStdRowMarked stdRowPtr
```

Reports whether the specified standard row is marked. The command returns a 1 if the standard row is marked and a 0 if it is not.

Parameters

<i>stdRowPtr</i>	Address of the standard row.
------------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsStdRowMarked $stdRowPtr
```

Returns:

1

The standard row is marked.

dbIsStdRowSel

`dbIsStdRowSel stdRowPtr`

Reports whether the specified standard row is selected. The command returns a `1` if the standard row is selected and a `0` if it is not.

Parameters

<code>stdRowPtr</code>	Address of the standard row.
------------------------	------------------------------

Command Order

Use this command after importing the design.

Example

`dbIsStdRowSel $stdRowPtr`

Returns:

`0`

The standard row is not selected.

dbIsStripBoxDrcObj

`dbIsStripBoxDrcObj stripBoxPtr`

Reports whether the specified strip box is a drc object. The command returns a `1` if the strip box is drc object and a `0` if it is not.

Parameters

<i>stripBoxPtr</i>	Address of the strip box.
--------------------	---------------------------

Command Order

Use this command after importing the design and running verify.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96ddcb0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0x986599c
```

```
dbIsStripBoxDrcObj $objPtr
```

Returns:

```
1
```

The strip box is a drc object.

dbIsStripBoxHilite

```
dbIsStripBoxHilite stripBoxPtr
```

Reports whether the specified strip box is highlighted. The command returns 1 if the strip box is highlighted and 0 if it is not.

Parameters

<i>stripBoxPtr</i>	Address of the strip box.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one power and ground strip.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96ddcb0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0x986599c
```

```
dbIsStripBoxHilite $objPtr
```

Returns:

```
1
```

The strip box is highlighted.

dbIsStripBoxHookup

```
dbIsStripBoxHookup stripBoxPtr
```

Reports whether the specified strip box is hooked up. The command returns a `1` if the strip box is hooked up and a `0` if it is not.

Parameters

<i>stripBoxPtr</i>	Address of the strip box.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one power and ground strip.

Example

```
dbIsStripBoxHookup $objPtr
```

Returns:

0

The strip box is not hooked up.

dbIsStripBoxHor

```
dbIsStripBoxHor stripBoxPtr
```

Reports whether the specified strip box is horizontal. The command returns a 1 if the strip box is horizontal and a 0 if it is not.

Parameters

<i>stripBoxPtr</i>	Address of the strip box.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one power and ground strip.

Example

```
dbIsStripBoxHor $objPtr
```

Returns:

0

The strip box is not horizontal.

dbIsStripBoxRail

```
dbIsStripBoxRail stripBoxPtr
```

Reports whether the specified strip box is a rail. The command returns a 1 if the strip box is a rail and a 0 if it is not.

Parameters

<i>stripBoxPtr</i>	Address of the strip box.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one power and ground strip.

Example

```
dbIsStripBoxRail $objPtr
```

Returns:

0

The strip box is a rail.

dbIsStripBoxSel

```
dbIsStripBoxSel stripBoxPtr
```

Reports whether the specified strip box is selected. The command returns a 1 if the strip box is selected and a 0 if it is not.

Parameters

<i>stripBoxPtr</i>	Address of the strip box.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one power and ground strip.

Example

```
dbIsStripBoxSel $objPtr
```

Returns:

1

The strip box is selected.

dbIsStripBoxSignal

```
dbIsStripBoxSignal stripBoxPtr
```

Returns a 1 if stripbox is signal, else a 0.

Parameters

<i>stripBoxPtr</i>	Address of strip box.
--------------------	-----------------------

Command Order

Use this command after design import and creation of at least one power/ground strap.

Example

```
dbIsStripBoxSignal $objPtr
```

Returns:

0

The strip box is not a signal.

dbIsStripBoxVer

```
dbIsStripBoxVer stripBoxPtr
```

Reports whether the specified strip box is vertical. The command returns a 1 if the strip box is vertical and a 0 if it is not.

Parameters

<i>stripBoxPtr</i>	Address of the strip box.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one power and ground strip.

Example

```
dbIsStripBoxVer $objPtr
```

Returns:

1

The strip box is vertical.

dbIsStripBoxVia

`dbIsStripBoxVia stripBoxPtr`

Reports whether the specified stripbox is a via. The command returns 1 if the stripbox is a via and 0 if it is not.

Parameters

<code><i>stripBoxPtr</i></code>	Address of stripbox.
---------------------------------	----------------------

Command Order

Use this command after importing the design and creating at least one power or ground strap.

Example

`dbIsStripBoxVia $objPtr`

Returns:

0

The strip box is not a via.

dbIsStripBoxViaCell

dbIsStripBoxViaCell *stripBoxPtr*

Returns a 1 if the specified stripbox is a via cell, else a 0.

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
--------------------	----------------------

Command Order

Use this command after design import and creation of at least one power/ground strap.

Example

dbIsStripBoxViaCell \$objPtr

Returns:

0

The strip box is not a via cell.

dbIsTechSiteArealo

dbIsTechSiteArealo

dbIsTechSiteArealo *techSitePtr*

Reports whether the specified tech site is an area I/O site. The command returns 1 if the tech site is an area I/O site and 0 if it is not.

Parameters

<i>techSitePtr</i>	Address of tech site.
--------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTechSiteAreaIo [dbGetTechSiteByName tsm3site]
```

Returns:

0

The tech site `tsm3site` is not an area I/O site.

dbIsTechSiteCore

dbIsTechSiteCore

```
dbIsTechSiteCore techSitePtr
```

Reports whether the specified tech site is a core site. The command returns `1` if the tech site is a core site and `0` if it is not.

Parameters

<code>techSitePtr</code>	Address of tech site.
--------------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTechSiteCore [dbGetTechSiteByName tsm3site]
```

Returns:

1

The tech site `tsm3site` is a core site.

dbIsTechSiteGNDOnBottom

dbIsTechSiteGNDOnBottom

`dbIsTechSiteGNDOnBottom techSitePtr`

Reports whether the specified tech site has ground on the bottom. The command returns `1` if the tech site has ground on the bottom and `0` if it does not.

Parameters

<code>techSitePtr</code>	Address of tech site.
--------------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTechSiteGNDOnBottom [dbGetTechSiteByName tsm3site]
```

Returns:

1

The tech site `tsm3site` has ground on the bottom.

dbIsTechSiteMarked

dbIsTechSiteMarked

dbIsTechSiteMarked *techSitePtr*

Reports whether the specified tech site is marked. The command returns 1 if the tech site is marked and 0 if it is not.

Parameters

<i>techSitePtr</i>	Address of tech site.
--------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTechSiteMarked [dbGetTechSiteByName tsm3site]
```

Returns:

0

The tech site `tsm3site` is not marked.

dbIsTechSiteSymmetryR90

dbIsTechSiteSymmetryR90

dbIsTechSiteSymmetryR90 *techSitePtr*

Reports whether the specified tech site allows a symmetry of R90. The command returns 1 if the

Reports whether the specified tech site allows a symmetry of R90 about the command's vertical axis. The tech site allows a symmetry of R90 and 0 if it does not.

Parameters

<i>techSitePtr</i>	Address of tech site.
--------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTechSiteSymmetryR90 [dbGetTechSiteByName tsm3site]
```

Returns:

0

The tech site `tsm3site` does not allow a symmetry of R90.

dbIsTechSiteSymmetryX

dbIsTechSiteSymmetryX

```
dbIsTechSiteSymmetryX techSitePtr
```

Reports whether the specified tech site allows a symmetry of flipping about the X axis. The command returns 1 if the tech site allows a symmetry of flipping about the X axis and 0 if it does not.

Parameters

<i>techSitePtr</i>	Address of tech site.
--------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTechSiteSymmetryX [dbGetTechSiteByName tsm3site]
```

Returns:

0

The tech site `tsm3site` does not allow a symmetry of flipping about the X axis.

dbIsTechSiteSymmetryY

dbIsTechSiteSymmetryY

```
dbIsTechSiteSymmetryY techSitePtr
```

Reports whether the specified tech site allows a symmetry of flipping about the Y axis. The command returns `1` if the tech site allows a symmetry of flipping about the Y axis and `0` if it does not.

Parameters

<code>techSitePtr</code>	Address of tech site.
--------------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTechSiteSymmetryY [dbGetTechSiteByName tsm3site]
```

Returns:

0

The tech site `tsm3site` does not allow a symmetry of flipping about the Y axis.

dbIsTechSiteVDDOnBottom

dbIsTechSiteVDDOnBottom

`dbIsTechSiteVDDOnBottom techSitePtr`

Reports whether the specified tech site has VDD on the bottom. The command returns `1` if the tech site has VDD on the bottom and `0` if it does not.

Parameters

<code>techSitePtr</code>	Address of tech site.
--------------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTechSiteVDDOnBottom [dbGetTechSiteByName tsm3site]
```

Returns:

0

The tech site `tsm3site` does not have VDD on the bottom.

dbIsTermAsyncCtrl

dbIsTermAsyncCtrl

`dbIsTermAsyncCtrl termPtr`

Reports whether the specified terminal is asynchronously controlled. The command returns `1` if the terminal is asynchronously controlled and `0` if it is not.

Parameters

<code><i>termPtr</i></code>	Address of the terminal.
-----------------------------	--------------------------

Command Order

Use this command after importing the design.

Example

`dbIsTermAsyncCtrl [dbGetTermByName vcom]`

Returns:

`1`

The term is asynchronously controlled.

dbIsTermBidi

`dbIsTermBidi termPtr`

Reports whether the specified terminal is bidirectional. The command returns a `1` if the terminal is bidirectional and a `0` if it is not.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTermBidi $termPtr
```

Returns:

1

The term is bidirectional.

dbIsTermBrkLoop

```
dbIsTermBrkLoop termPtr
```

Check if the combinational logic loop is broken on the specified term (for timing analysis).

Parameters

<i>termPtr</i>	
----------------	--

Command Order

Use this command after

Example

```
dbIsTermBrkLoop $termPtr
```

Returns:

1

The term has a combinational timing loop.

dbIsTermClk

```
dbIsTermClk termPtr
```

Reports whether the specified terminal is a clock. The command returns a 1 if the terminal is a clock and a 0 if it is not.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set termPtr [dbGetTermByName [dbGetInstByName SH28/I12/I36/MINIFY1/CAPINTLC1/U3] Z2]
```

Returns:

0x60a19e8

```
dbIsTermClk $termPtr
```

Returns:

0

The term is not a clock.

dbIsTermD

`dbIsTermD termPtr`

Reports whether the specified terminal is a D terminal. The command returns a `1` if the terminal is a D terminal and a `0` if it is not.

Parameters

<code><i>termPtr</i></code>	Address of the terminal.
-----------------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set termPtr [dbGetTermByName [dbGetInstByName SH28/I12/I36/MINIFY1/CAPINTLC1/U3] Z2]
```

Returns:

`0x60a19e8`

```
dbIsTermD $termPtr
```

Returns:

`0`

The term is not the D input.

dbIsTermFeed

`dbIsTermFeed termPtr`

Reports whether the specified terminal is a feed. The command returns a `1` if the terminal is a feed and a `0` if it is not.

Parameters

<code>termPtr</code>	Address of the terminal.
----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set termPtr [dbGetTermByName [dbGetInstByName SH28/I12/I36/MINIFY1/CAPINTLC1/U3] Z2]
```

Returns:

`0x60a19e8`

`dbIsTermFeed $termPtr`

Returns:

`0`

The ter is not a feedthru.

dbIsTermFTerm

`dbIsTermFTerm termPtr`

Reports whether the specified terminal is an f-term. The command returns a `1` if the terminal is an f-term and a `0` if it is not

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachNetInputTerm $netPtr termPtr {  
    set ioTerm [dbIsTermFTerm $termPtr]  
    #puts $termPtr  
    if { $ioTerm == 0 } {  
        set termName [dbTermName $termPtr]  
        set loadInst [dbTermInstName $termPtr]  
        set cellName [dbInstCellName $loadInst]  
        set cellPtr [dbGetCellByName $cellName]  
        set ftermPtr [dbGetFTermByName $cellPtr $termName]  
        set pinCap [expr [dbFTermPinCap $ftermPtr]*[dbHeadPicoFPerDBU  
        ]]  
        set maxPinCap [expr [dbFTermMaxCap $ftermPtr]*[dbHeadPicoFPer  
        DBU]]  
    } else {  
        set termName [dbObjName $termPtr]  
        set srcInst "Primary"  
        set pinCap [expr [dbFTermPinCap $termPtr]*[dbHeadPicoFPerDBU]  
        ]  
        set maxPinCap [expr [dbFTermMaxCap $termPtr]*[dbHeadPicoFPerD
```

```
BU] ]  
}
```

dbIsTermHilite

```
dbIsTermHilite termPtr
```

Reports whether the specified terminal is highlighted. The command returns 1 if the terminal is highlighted and 0 if it is not.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set termPtr [dbGetTermByName [dbGetInstByName SH28/I12/I36/MINIFY1/CAPINTLC1/U3] Z2]
```

Returns:

```
0x60a19e8
```

```
dbIsTermHilite $termPtr
```

Returns:

```
0
```

The terminal is not highlighted.

dbIsTermIgnored

dbIsTermIgnored *termPtr*

Returns a 1 if terminal is ignored, else a 0.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
set termPtr [dbGetTermByName [dbGetInstByName SH28/I12/I36/MINIFY1/CAPINTLC1/U3] Z2]
```

Returns:

0x60a19e8

```
dbIsTermIgnored $termPtr
```

Returns:

0

The term is not ignored.

dbIsTermInCluInst

dbIsTermInCluInst *termPtr*

Returns a 1 if the specified term is in the cluster instance, else a 0.

Parameters

<i>termPtr</i>	Address of term.
----------------	------------------

Command Order

Use this command after the timing graph has been built.

Example

```
dbIsTermInInCluInst [dbGetTermByName BLU]
```

Returns:

0

The term is not in the cluster.

dbIsTermInPath

```
dbIsTermInPath termPtr pathPtr
```

Returns a 1 if the specified term is in the specified path, else a 0.

Parameters

<i>termPtr</i>	Address of term.
<i>pathPtr</i>	Address of path

Command Order

Use this command after the timing graph has been built.

Example

```
dbIsTermInPath [dbGetTermByName BLU] $pathPtr
```

Returns:

0

The term is not in the path.

dbIsTermInput

```
dbIsTermInput termPtr
```

Reports whether the specified terminal is an input terminal. The command returns a 1 if the terminal is an input and a 0 if it is not.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design

Example

```
set termPtr [dbGetTermByName [dbGetInstByName SH28/I12/I36/MINIFY1/CAPINTLC1/U3] Z2]
```

Returns:

0x60a19e8

```
dbIsTermInput $termPtr
```

Returns:

0

The term is not an input.

dbIsTermMarked

`dbIsTermMarked termPtr`

Reports whether the specified terminal is marked. The command returns a `1` if the terminal is marked and a `0` if it is not.

Parameters

<code><i>termPtr</i></code>	Address of the terminal.
-----------------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set termPtr [dbGetTermByName [dbGetInstByName SH28/I12/I36/MINIFY1/CAPINTLC1/U3] Z2]
```

Returns:

`0x60a19e8`

```
dbIsTermMarked $termPtr
```

Returns:

`0`

The term is not marked.

dbIsTermMarked2

`dbIsTermMarked2 termPtr`

Returns a 1 if the specified terminal is marked 2, else a 0.

Parameters

<code>termPtr</code>	Address of terminal.
----------------------	----------------------

Command Order

Use this command after design import.

Example

`dbIsTermMarked2 $termPtr`

Returns:

0

The term is not marked 2.

dbIsTermMarked3

`dbIsTermMarked3 termPtr`

Returns a 1 if the specified term is marked 3, else a 0.

Parameters

<code>termPtr</code>	Address of terminal.
----------------------	----------------------

Command Order

Use this command after design import.

Example

```
dbIsTermMarked3 $termPtr
```

Returns:

0

The term is not marked 3.

dbIsTermMPW

dbIsTermMPW

```
dbIsTermMPW termPtr
```

Reports whether the specified terminal is minimum pulse width. The command returns 1 if the terminal is minimum pulse width and 0 if it is not.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTermMPW [dbGetTermByName vcom]
```

Returns:

1

The terminal is minimum pulse width.

dbIsTermMultiHInst

dbIsTermMultiHInst *termPtr*

Returns a 1 if the specified terminal is on multiple hierarchical instances, else a 0.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after design import.

Example

dbIsTermMultiHInst \$termPtr

Returns:

0

The terminal is not on multiple hierarchical instances.

dbIsTermMultiInst

dbIsTermMultiInst *termPtr*

Returns a 1 if specified terminal is on multiple instances, else a 0.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after design import.

Example

```
dbIsTermMultiInst $termPtr
```

Returns:

0

The term is not on multiple instances.

dbIsTermOutput

```
dbIsTermOutput termPtr
```

Reports whether the specified terminal is an output terminal. The command returns a 1 if the terminal is an output and a 0 if it is not.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set termPtr [dbGetTermByName [dbGetInstByName SH28/I12/I36/MINIFY1/CAPINTLC1/U3] Z2]
```

Returns:

0x60a19e8

```
dbIsTermOutput $termPtr
```

Returns:

1

The term is an output.

dbIsTermQ

```
dbIsTermQ termPtr
```

Reports whether the specified terminal is a Q terminal. The command returns a 1 if the terminal is a Q terminal and a 0 if it is not.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
set termPtr [dbGetTermByName [dbGetInstByName SH28/I12/I36/MINIFY1/CAPINTLC1/U3] Z2]
```

Returns:

0x60a19e8

`dbIsTermQ $termPtr`

Returns:

0

The term is not a Q.

dbIsTermSel

`dbIsTermSel termPtr`

Reports whether the specified terminal is selected. The command returns a 1 if the terminal is selected and a 0 if it is not.

Parameters

<code><i>termPtr</i></code>	Address of the terminal.
-----------------------------	--------------------------

Command Order

Use this command after importing the design.

Example

`dbIsTermSel $termPtr`

Returns:

0

The term is not selected.

dbIsTermSpecial

`dbIsTermSpecial termPtr`

Reports whether the specified terminal is a special net. The command returns a `1` if the terminal is a special net and a `0` if it is not.

Parameters

<code>termPtr</code>	Address of the terminal.
----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTermSpecial $termPtr
```

Returns:

`0`

The term is not special.

dbIsTermTAClkSrc

`dbIsTermTAClkSrc termPtr`

Reports whether the specified terminal is a clock source. The command returns a `1` if the terminal is a clock source and a `0` if it is not.

Parameters

<code>termPtr</code>	Address of the terminal.
----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTermTAClkSrc $termPtr
```

Returns:

0

The term is not a clock source for timing analysis.

dbIsTermTAIgnored

```
dbIsTermTAIgnored termPtr
```

Returns a 1 if the specified terminal is ignored by timing analysis, else a 0.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after timing analysis.

Example

```
dbIsTermTAIgnored $termPtr
```

Returns:

0

The term should not be ignored during timing analysis.

dbIsTermTieHi

```
dbIsTermTieHi termPtr
```

Reports whether the specified terminal is a tie-hi terminal. The command returns a `1` if the terminal is a tie-hi terminal and a `0` if it is not.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTermTieHi $termPtr
```

Returns:

`0`

The term is not a tie-hi.

dbIsTermTieHiOrLo

```
dbIsTermTieHiOrLo termPtr
```

Reports whether the specified terminal is a tie-high or a tie-low terminal. The command returns a `1` if the terminal is a tie-high or a tie-low terminal and a `0` if it is neither.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTermTieHiOrLo $termPtr
```

Returns:

0

The term is not a tie-hi or a tie-lo.

dbIsTermTieLo

```
dbIsTermTieLo termPtr
```

Reports whether the specified terminal is a tie-lo terminal. The command returns a `1` if the terminal is a tie-lo terminal and a `0` if it is not.

Parameters

<i>termPtr</i>	Address of the terminal
----------------	-------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTermTieLo $termPtr
```

Returns:

0

The term is not a tie-lo.

dbIsTermTriCtl

```
dbIsTermTriCtl termPtr
```

Reports whether the specified terminal is a tristate terminal. The command returns a 1 if the terminal is a tristate terminal and a 0 if it is not.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsTermTriCtl $termPtr
```

Returns:

0

The term is not a tristate.

dbIsTermUnused

`dbIsTermUnused termPtr`

Reports whether the specified terminal is unused. The command returns a 1 if the terminal is unused and a 0 if it is not.

Parameters

<code>termPtr</code>	Address of the terminal.
----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

`dbIsTermUnused $termPtr`

Returns:

0

The term is used.

dbIsViaCellDefault

`dbIsViaCellDefault viacellPtr`

Returns a 1 if the specified via cell is the default (as defined in the technology file), else a 0.

Parameters

viacellPtr	Address of via cell.
------------	----------------------

Command Order

Use this command after design import.

Example

```
dbIsViaCellDefault $viacellPtr
```

Returns:

1

The via cell is the default.

dbIsViaCellFromDef

```
dbIsViaCellFromDef viacellPtr
```

Returns a 1 if the specified via cell is from DEF, else a 0.

Parameters

viacellPtr	Address of via cell.
------------	----------------------

Command Order

Use this command after design import.

Example

```
dbIsViaCellFromDef $viacellPtr
```

Returns:

1

The via cell came from DEF.

dbIsViaCellNonDefault

dbIsViaCellNonDefault *viacellPtr*

Returns a 1 if the via cell is NonDefault, else a 0.

Parameters

<i>viacellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after design import.

Example

dbIsViaCellNonDefault \$viaCellPtr

Returns:

1

The via cell is a non-default via cell.

dbIsViaCellRegular

dbIsViaCellRegular *viacellPtr*

Returns a 1 if the via cell is regular, else a 0. Each via is an instance of a via cell; creating a via requires previously created via cell. A via cell contains a lower metal layer, an upper metal layer, and a "cut" layer (for the via holes) that connects the two metal layers. Each layer can contain any

number, zero included, of arbitrary rectangles. Given a via cell, if its via holes are spaced evenly horizontally and vertically to form a full X by Y 2-D array (the X spacing may be different from the Y spacing), the via cell is "regular." A regular via cell is thus 2-D regular on the cut layer, but may not be regular on the metal layers.

Parameters

<i>viacellPtr</i>	Address of Verilog instance.
-------------------	------------------------------

Command Order

Use this command after design import.

Example

```
dbIsViaCellRegular $viacellPtr
```

Returns:

1

The via cell is a regular via cell.

dbIsVInstHilite

```
dbIsVInstHilite vinstPtr
```

Reports whether the Verilog instance is highlighted. The command returns `1` if Verilog instance is highlighted and `0` if it is not.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after design import.

Example

```
dbIsVInstHilite $vinstPtr
```

Returns:

1

The Verilog instance is highlighted.

dbIsVInstMarked

```
dbIsVInstMarked vinstPtr
```

Returns a 1 if the specified Verilog instance is marked, else a 0.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after design import.

Example

```
dbIsVInstMarked $vinstPtr
```

Returns:

0

The Verilog instance is not marked.

dbIsVInstMarked2

dbIsVInstMarked2 *vinstPtr*

Returns a 1 if Verilog instance is marked 2, else a 0.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after design import.

Example

```
dbIsVInstMarked2 $vinstPtr
```

Returns:

1

The Verilog instance is not marked 2.

dbIsVInstMarked3

dbIsVInstMarked3 *vinstPtr*

Returns a 1 if the specified Verilog instance is marked 3, else a 0.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after design import.

Example

```
dbIsVInstMarked3 $vinstPtr
```

Returns:

1

The Verilog instance is not marked 3.

dbIsVInstMarked4

```
dbIsVInstMarked4 vinstPtr
```

Reports whether the specified Verilog instance is marked 4. The command returns 1 if the Verilog instance is marked 4 and 0 if it is not.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsVInstMarked4 $vinstPtr
```

Returns:

1

The Verilog instance is not marked ⁴.

dbIsVInstSel

`dbIsVInstSel vinstPtr`

Returns a 1 if the specified Verilog instance is selected, else a 0.

Parameters

<code>vinstPtr</code>	Address of Verilog instance.
-----------------------	------------------------------

Command Order

Use this command after design import.

Example

`dbIsVInstSel $vinstPtr`

Returns:

1

The Verilog instance is selected.

dbIsVInstUnused

`dbIsVInstUnused vinstPtr`

Reports whether the specified Verilog instance is unused. The command returns `1` if the Verilog instance is unused and `0` if it is not.

Parameters

<code>vinstPtr</code>	Address of verilog instance.
-----------------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbIsVInstUnused $vinstPtr
```

Returns:

`1`

The Verilog instance is unused.

dbIsVNetBNet

`dbIsVNetBNet vnetPtr`

Returns the address of the associated bnet for the specified Verilog net.

Parameters

<code>vnetPtr</code>	Address of Verilog net.
----------------------	-------------------------

Command Order

Use this command after design import.

Example

```
dbIsVNetBNet $vnetPtr
```

Returns:

0x80947320

The address of the Verilog bnet is 0x80947320.

dbIsVNetHilite

```
dbIsVNetHilite vnetPtr
```

Reports whether the Verilog net is highlighted. The command returns 1 if the Verilog net is highlighted and 0 if it is not.

Parameters

<i>vnetPtr</i>	Address of Verilog net.
----------------	-------------------------

Command Order

Use this command after design import.

Example

```
dbIsVNetHilite $vnetPtr
```

Returns:

1

The Verilog net is highlighted.

dbIsVNetMarked

`dbIsVNetMarked vnetPtr`

Returns a 1 if the specified Verilog net is marked, else a 0.

Parameters

<code>vnetPtr</code>	Address of Verilog net.
----------------------	-------------------------

Command Order

Use this command after design import.

Example

`dbIsVNetMarked $vnetPtr`

Returns:

1

The Verilog net is marked.

dbIsVNetMarked2

`dbIsVNetMarked2 vnetPtr`

Returns a 1 if the specified Verilog net is marked 2, else a 0.

Parameters

<i>vnetPtr</i>	Address of Verilog net.
----------------	-------------------------

Command Order

Use this command after design import.

Example

```
dbIsVNetMarked2 $vnetPtr
```

Returns:

0

The Verilog net is not marked 2.

dbIsVNetSel

```
dbIsVNetSel vnetPtr
```

Returns a 1 if the specified Verilog net is selected, else a 0.

Parameters

<i>vnetPtr</i>	Address of Verilog net.
----------------	-------------------------

Command Order

Use this command after design import.

Example

```
dbIsVNetSel $vnetPtr
```

Returns:

1

The Verilog net is selected.

dbIsVTermFTerm

```
dbIsVTermFTerm vtermPtr
```

Returns the address of the specified f-term.

Parameters

<i>vtermPtr</i>	Address of Verilog terminal.
-----------------	------------------------------

Command Order

Use this command after design import.

Example

```
dbForEachVNetVTerm $vnetPtr vtermPtr {  
    set fterm [dbVTermFTerm $vtermPtr]  
    set name [dbFTermName $fterm]  
    puts "name = $name"  
}
```

The above script returns the f-term name for all of the f-terms on the specified Verilog net.

dbIsVTermTieHi

`dbIsVTermTieHi vtermPtr`

Returns a 1 if the specified Verilog terminal is a tie-hi, else a 0.

Parameters

<code>vtermPtr</code>	Address of Verilog terminal.
-----------------------	------------------------------

Command Order

Use this command after design import.

Example

```
dbForEachVNetVTerm $vnetPtr vtermPtr {  
    set fterm [dbVTermFTerm $vterPtr]  
    set name [dbFTermName $fterm]  
    puts "name = $name"  
    set tiehi [dbIsVTermTieHi $vtermPtr]  
    puts "tiehi = $tiehi"  
}
```

The above script f-term name and the tie-hi status for the specified Verilog net.

dbIsVTermTieLo

`dbIsVTermTieLo vtermPtr`

Returns a 1 if the specified Verilog terminal is a tie-lo, else a 0.

Parameters

<i>vtermPtr</i>	Address of Verilog terminal.
-----------------	------------------------------

Command Order

Use this command after design import.

Example

```
dbForEachVNetVTerm $vnetPtr vtermPtr {  
    set fterm [dbVTermFTerm $vterPtr]  
    set name [dbFTermName $fterm]  
    puts "name = $name"  
    set tielo [dbIsVTermTieLo $vtermPtr]  
    puts "tielo = $tielo"]  
}
```

The above script returns the f-term name and the tie-lo status for all Verilog terminals in the specified Verilog net.

dbIsWireDel

```
dbIsWireDel wirePtr
```

Reports whether the specified wire was deleted. The command returns a 1 if the wire was deleted and a 0 if it was not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96ddcb0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0xb129ad4
```

```
dbIsWireDel $objPtr
```

Returns:

```
0
```

The wire has not been deleted.

dbIsWireDup

```
dbIsWireDup wirePtr
```

Reports whether the specified wire is a duplicate. The command returns a 1 if the wire is a duplicate and a 0 if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96ddcb0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0xb129ad4
```

```
dbIsWireDup $objPtr
```

Returns:

```
0
```

The wire is not a duplicate.

dbIsWireGlobal

```
dbIsWireGlobal wirePtr
```

Reports whether the specified the wire has been global routed. The command returns a 1 if the wire is global routed and a 0 if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96ddcb0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0xb129ad4
```

```
dbIsWireGlobal $objPtr
```

Returns:

```
1
```

The wire has been globally routed.

dbIsWireHilite

```
dbIsWireHilite wirePtr
```

Reports whether the specified wire is highlighted. The command returns `1` if the wire is highlighted and `0` if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after routing the design.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96ddcb0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0xb129ad4
```

```
dbIsWireHilite $objPtr
```

Returns:

```
1
```

The wire is highlighted.

dbIsWireHor

```
dbIsWireHor wirePtr
```

Reports whether the specified wire is horizontal. The command returns a `1` if the wire is horizontal and a `0` if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
set s [dbHeadSelList]
```

Returns:

```
0x96ddcb0
```

```
set objPtr [dbSelPtr $s]
```

Returns:

```
0xb129ad4
```

```
dbIsWireHor $objPtr
```

Returns:

```
1
```

The wire is horizontal.

dbIsWireMarked

```
dbIsWireMarked wirePtr
```

Reports whether the specified wire is marked. The command returns a 1 if the wire is marked and a 0 if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
dbIsWireMarked $objPtr
```

Returns:

0

The wire is not marked.

dbIsWireMarked2

```
dbIsWireMarked2 wirePtr
```

Returns a 1 if the specified wire is marked 2, else a 0.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after route.

Example

```
dbIsWireMarked2 $wirePtr
```

Returns:

1

The wire is marked 2.

dbIsWireMaster

```
dbIsWireMaster wirePtr
```

Reports whether the specified wire is a master. The command returns a `1` if the wire is a master and a `0` if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
dbIsWireMaster $objPtr
```

Returns:

1

The wire is the master.

dbIsWireSel

```
dbIsWireSel wirePtr
```

Reports whether the specified wire is selected. The command returns a 1 if the wire is selected and a 0 if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbIsWireSel $objPtr
```

Returns:

1

The wire is selected.

dbIsWireSlave

```
dbIsWireSlave wirePtr
```

Reports whether the specified wire is a slave. The command returns a 1 if the wire is a slave and a 0 if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
dbIsWireSlave $objPtr
```

Returns:

0

The wire is not a slave.

dbIsWireTerm

```
dbIsWireTerm wirePtr
```

Reports whether the specified wire is a terminal. The command returns a 1 if the wire is a terminal and a 0 if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
dbIsWireTerm $objPtr
```

Returns:

0

The wire is not a terminal.

dbIsWireVer

```
dbIsWireVer wirePtr
```

Reports whether the specified wire is vertical. The command returns a 1 if the wire is vertical and a 0 if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
dbIsWireVer $objPtr
```

Returns:

0

The wire is not vertical.

dbIsWireViaHilite

```
dbIsWireViaHilite wirePtr
```

Reports whether the via is highlighted on the specified wire. The command returns `1` if the via is highlighted on the wire and `0` if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after routing the design.

Example

```
dbIsWireViaHilite $objPtr
```

Returns:

0

The via is not highlighted.

dbIsWireViaSel

```
dbIsWireViaSel wirePtr
```

Reports whether the via is selected on the specified wire. The command returns `1` if the via is selected on the wire and `0` if it is not.

Parameters

<i>wirePtr</i>	Address of the wire.
----------------	----------------------

Command Order

Use this command after route.

Example

```
dbIsWireViaSel $objPtr
```

Returns:

0

The via is not selected.

dbIsWireX

```
dbIsWireX wirePtr
```

Returns a 1 if wire is a X(duplicate from design import for distinguishing between hierarchical and flat), else a 0.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after route.

Example

```
dbIsWireX $objPtr
```

Returns:

0

The wire is not a duplicate.

dbIterAllRoutes

```
dbIterAllRoutes
```

Iterates all routes.

Command Order

Use this command after route.

Note: The command should always be used (as a pair) with the [dbEndIterAllRoutes](#) command to ensure proper functioning with other applications.

Example

```
proc iter_routes_delete_vias {} {  
    set fout [open "IterRoutesRmVias.out" w]  
    puts $fout "Routes Info:  
"  
    puts $fout ""  
    puts $fout ""  
    dbForAllCellNet [dbgTopCell] net {  
        set nName [dbNetName $net]  
        dbIterAllRoutes  
        set iter [dbIterRoutes $net]  
        while {[set route [dbRouteNext $iter]] != {0x0}} {  
            puts $fout "Processing net -> $nName\n"
```

```
if { [dbIsRouteWire $route] } {  
    puts $fout " wirePtr -> $route"  
  
    set wire_info [dbInfoWire $route]  
  
    puts $fout " wire -> $wire_info"  
  
    if { [ dbIsObjStripBox $route ] } {  
  
        set rc [ dbDeleteStripBox $route ]  
  
        puts $fout " Deleted the (strip box) wire, $route, rc = $rc"  
    } else {  
  
        set rc [ dbDeleteRouteBox $route ]  
  
        puts $fout " Deleted the (route box) wire, $route, rc = $rc"  
    }  
  
} elseif { [dbIsRouteVia $route] } {  
  
    puts $fout " viaPtr -> $route"  
  
    set via_info [dbInfoVia $route]  
  
    puts $fout " via -> $via_info"  
  
    if { [ dbIsObjStripBox $route ] } {  
  
        set rc [ dbDeleteStripBox $route ]  
  
        puts $fout " Deleted the (strip box) via, $route, rc = $rc"  
    } else {  
  
        set rc [ dbDeleteRouteBox $route ]  
  
        puts $fout " Deleted the (route box) via, $route, rc = $rc"  
    }  
  
} else {  
  
    puts $fout "ROUTE NOT A WIRE OR A VIA"  
}  
  
puts $fout "\n"  
}  
  
dbEndIterRoutes $iter
```

```
dbEndIterAllRoutes  
}  
  
close $fout  
  
return  
}
```

Deletes all power/ground structures.

dbIterRoutes

dbIterRoutes *netPtr*

Iterates routes for the specified net

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after route.

Example

```
proc iter_routes_delete_vias {} {  
  
    set fout [open "IterRoutesRmVias.out" w]  
  
    puts $fout "Routes Info:"  
  
    puts $fout ""  
  
    puts $fout ""  
  
    dbForAllCellNet [dbgTopCell] net {
```

```
set nName [dbNetName $net]
dbIterAllRoutes
set iter [dbIterRoutes $net]
while {[set route [dbRouteNext $iter]] != {0x0}} {
    puts $fout "Processing net -> $nName\n"
    if { [dbIsRouteWire $route] } {
        puts $fout " wirePtr -> $route"
        set wire_info [dbInfoWire $route]
        puts $fout " wire -> $wire_info"
        if { [ dbIsObjStripBox $route ] } {
            set rc [ dbDeleteStripBox $route ]
            puts $fout " Deleted the (strip box) wire, $route, rc = $rc"
        } else {
            set rc [ dbDeleteRouteBox $route ]
            puts $fout " Deleted the (route box) wire, $route, rc = $rc"
        }
    } elseif { [dbIsRouteVia $route] } {
```

dbIterViaCellRectangles

dbIterViaCellRectangles *layer viacell*

Returns all the rectangles for the specified via cell.

Parameters

<i>layer</i>	0 for lower metal, 1 for cut layer, 2 for upper metal
--------------	---

viacell

Address of via cell.

Command Order

Use this command after route.

Example

```
set iter [dbIterViaCellRectangles LAYER $vc]
while {[set rect [dbViaCellRectangleNext $iter]] != ""} {
    # $rect is list of four coordinates: lx ly hx hy
    ...
    if (want to break out of the iterator loop)
        dbEndIterViaCellRectangles $iter
    ...
}
```

The above script is pseudo code for listing all the rectangle coordinates for every via.

Database Commands L - R

- [L](#)
- [M](#)
- [N](#)
- [O](#)
- [P](#)
- [Q](#)
- [R](#)

L

dbLayerArrNext

`dbLayerArrNext layerPtr`

Returns the address of the next layer.

Parameters

<code>layerPtr</code>	Address of the layer.
-----------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 138>dbLayerArrNext $objPtr  
0xdb8dd24
```

The address of the next layer is 0xdb8dd24.

dbLayerBlkBox

`dbLayerBlkBox layerBlkPtr`

Returns the box that is being blocked for the specified layer.

Parameters

<code><i>layerBlkPtr</i></code>	Address of the routing blockage.
---------------------------------	----------------------------------

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
innovus 139> dbLayerBlkBox $layerBlkPtr  
4000 4000 16000 16000
```

The lower left coordinate for the routing blockage is (4.0 4.0), and the upper right coordinate is (16.0 16.0).

dbLayerBlkCutLayer

dbLayerBlkCutLayer *layerBlkPtr*

Returns the layers that are being blocked for routing on cut layers.

Parameters

<i>layerBlkPtr</i>	Address of the routing blockage.
--------------------	----------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 139> dbLayerBlkCutLayer $objPtr
```

4

The layers being blocked on cut layers is 4.

dbLayerBlkLayer

dbLayerBlkLayer *layerBlkPtr*

Returns the bit layer that is being blocked for routing for the specified layer blockage.

Note: There can be more than one layer being blocked for routing. The bit map only gives one layer. A "for loop" must be used to get all layers.

Parameters

<i>layerBlkPtr</i>	Address of the routing blockage.
--------------------	----------------------------------

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
dbLayerBlkLayer $objPtr
```

Returns:

4

The layer being blocked from routing is 3 (100).

The following example does not use `dbLayerBlkLayer`, but is useful for typical scripting:

```
dbForEachFPlanLayerBlk [dbCellFPlan [dbgTopCell]] blk {  
    puts [dbLayerLefName [ dbLayerShapeLayer $blk ]]  
}
```

dbLayerBlkName

```
dbLayerBlkName layerBlkPtr
```

Returns the name of the routing blockage for the specified layer blockage.

Parameters

<i>layerBlkPtr</i>	Address of the routing blockage.
--------------------	----------------------------------

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
encounter 140> dbLayerBlkName $objPtr  
defLayerBlkName
```

The name of the layer blockage is `defLayerBlkName`.

dbLayerBlkNext

```
dbLayerBlkNext layerBlkPtr
```

Returns the address of the next routing blockage. The command returns `0x0` if there is no next blockage.

Parameters

<code>layerBlkPtr</code>	Address of the routing blockage.
--------------------------	----------------------------------

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
innovus 141> dbLayerBlkNext $objPtr  
0x0
```

There is no next routing blockage.

dbLayerBlkType

dbLayerBlkType *layerBlkPtr*

Returns the type for the specified blockage. Possible values are:

dbcPtnCutLayerBlock

dbcRouteGuideLayerBlock

dbcUserLayerBlock

Parameters

<i>layerBlkPtr</i>	Address of the routing blockage.
--------------------	----------------------------------

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
innovus 142> dbLayerBlkType $objPtr  
dbcUserLayerBlock
```

The type of blockage is user.

dbLayerCap

dbLayerCap *layerPtr*

Returns a table of cap values for the specified metal layer.

Parameters

<i>layerPtr</i>	Address of the metal layer.
-----------------	-----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 9> dbLayerCap [dbGetLayerByName METAL2]  
{400 0.000204} {800 0.000154} {1200 0.000139} {1600 0.000133} {2000 0.000129} {2400  
0.000127} {2500 0.000129} {2800 0.000126} {3200 0.000125} {3600 0.000125} {4000  
0.000124} {4400 0.000124} {4800 0.000124} {5200 0.000123} {5600 0.000123} {6000  
0.000123} {6400 0.000123}
```

The cap table for layer 2 is returned.

dbLayerCapPerSQ

```
dbLayerCapPerSQ layerPtr
```

Returns the capacitance per square micron for the specified metal layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 10> dbLayerCapPerSQ [dbGetLayerByName METAL2]  
0.0
```

The capacitance per square for layer 2 is 0.0.

dbLayerDrawName

dbLayerDrawName *layerPtr*

Returns the drawn name for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 22> dbLayerDrawName [dbGetLayerByName METAL2]  
M2
```

The name of the metal layer is *M2*.

dbLayerExtension

dbLayerExtension *layerPtr*

Returns the amount of extension past the end of the point for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 26> dbLayerExtension [dbGetLayerByName METAL2]
```

```
200
```

The extension is 0.2.

dbLayerExtIdList

```
dbLayerExtIdList layerPtr
```

Returns the address of the external ID for the specified layer. The command returns `0x0` if there is no external ID for the layer.

Parameters

<i>layerPtr</i>	Address of layer.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
innovus 25> dbLayerExtIdList [dbGetLayerByName METAL2]  
0x0
```

There is no external ID for the specified layer.

dbLayerFillCheckLength

```
dbLayerFillCheckLength layerPtr
```

Returns the fill check length value for the specified layer. The command returns 0 if no value is specified.

Parameters

<i>layerPtr</i>	Address of layer.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
innovus 27> dbLayerFillCheckLength [dbGetLayerByName METAL2]  
0
```

There is no fill check length specified.

dbLayerFillCheckStep

dbLayerFillCheckStep *layerPtr*

Returns the fill check step value for the specified layer. The command returns 0 if no value is specified. Fill check step is defined in the LEF file as DENSITYCHECKSTEP.

Parameters

<i>layerPtr</i>	Address of layer.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
innovus 47> dbLayerFillCheckStep [dbGetLayerByName metal 3]  
0
```

There is no fill check step value specified.

dbLayerFillCheckWidth

dbLayerFillCheckWidth *layerPtr*

Returns 1 if the specified layer needs to be checked for fill and 0 if it does not.

Parameters

<i>layerPtr</i>	Specifies
-----------------	-----------

Command Order

Use this command after importing the design.

Example

```
innovus 34> dbLayerFillCheckWidth [dbGetLayerByName metal 3]  
0
```

The layer does not need to be checked.

dbLayerFillMaxDensity

dbLayerFillMaxDensity *layerPtr*

Returns the maximum density of the layer fill for the specified layer.

Parameters

<i>layerPtr</i>	address of layer.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
innovus 28> dbLayerFillMaxDensity [dbGetLayerByName METAL2]  
0.0
```

The maximum density of the layer fill is 0.0.

dbLayerFillMinDensity

dbLayerFillMinDensity *layerPtr*

Returns the minimum density of the layer fill for the specified layer.

Parameters

<i>layerPtr</i>	Address of layer.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
innovus 35> dbLayerFillMinDensity [dbGetLayerByName metal 3]  
0.0
```

The minimum density of the layer fill is 0.0.

dbLayerFillMinSpacing

dbLayerFillMinSpacing *layerPtr*

Returns the minimum spacing for fill metal for the specified layer.

Parameters

<i>layerPtr</i>	Address of layer.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
innovus 35> dbLayerFillMinSpacing [dbGetLayerByName metal 3]  
1.0
```

The minimum spacing for fill metal for layer 3 is 1.0.

dbLayerId

dbLayerId *layerPtr*

Returns the layer ID for the specified layer.

Parameters

<i>layerPtr</i>	Address of layer.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
innovus 30> dbLayerId [dbGetLayerByName METAL2]  
2
```

The layer ID is 2.

dbLayerLEFId

dbLayerLEFId *layerPtr*

Returns the LEF ID for the specified layer.

Parameters

<i>layerPtr</i>	Address of layer.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
innovus 30> dbLayerLEFId [dbGetLayerByName METAL2]  
0
```

The LEF ID is 0.

dbLayerLefName

dbLayerLefName *layerPtr*

Returns the LEF layer name for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 31> dbLayerLefName [dbGetLayerByName METAL2]  
METAL2
```

The LEF layer name is METAL2.

dbLayerMaxCap

```
dbLayerMaxCap layerPtr
```

Returns the maximum capacitance defined for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 33> dbLayerMaxCap [dbGetLayerByName METAL2]  
0.0
```

The maximum capacitance for *metal2* is 0.0.

dbLayerMaxWidth

dbLayerMaxWidth *layerPtr*

Returns the maximum width defined for the specified layer. The command returns 2147483647 if a maximum width is not defined.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 33> dbLayerMaxWidth [dbGetLayerByName METAL2]  
2147483647
```

There is no maximum width defined for *metal2*.

dbLayerMinArea

`dbLayerMinArea layerPtr`

Returns the minimum area defined for the specified layer.

Note: The `dbLayerMinArea` command returns values in square database units. To convert these units into square microns, you must specify the `dbDBUToMicrons` command twice (see Examples below).

Parameters

<code>layerPtr</code>	Specifies the pointer to the layer.
-----------------------	-------------------------------------

Command Order

Use this command after importing the design.

Examples

```
innovus 4> dbLayerMinArea [dbGetLayerByName M6]
168000.0
innovus 5> dbDBUToMicrons [dbDBUToMicrons [dbLayerMinArea
[dbGetLayerByName M6] ] ]
0.042
```

dbLayerMinCap

`dbLayerMinCap layerPtr`

Returns the minimum capacitance defined for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 34> dbLayerMinCap [dbGetLayerByName METAL2]  
0.0
```

The minimum capacitance for *metal2* is 0.0.

dbLayerMinSpace

dbLayerMinSpace *layerPtr*

Returns the minimum space for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 35> dbLayerMinSpace [dbGetLayerByName METAL2]
```

400

The minimum space for *metal2* is 0.4.

dbLayerMinWidth

dbLayerMinWidth *layerPtr*

Returns the minimum width for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 36> dbLayerMinWidth [dbGetLayerByName METAL2]
```

40

The minimum layer width for *metal2* is 0.04.

dbLayerName

dbLayerName *layerPtr*

Returns the layer name for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 37> dbLayerName [dbGetLayerByName METAL2]  
M2
```

The layer name for *metal2* is *M2*.

dbLayerNextInFE

```
dbLayerNextInFE layerPtr
```

Returns the address of the layer after the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 38> dbLayerNextInFE [dbGetLayerByName METAL2]  
0x48c8fa0
```

The next layer address is 0x48c8fa0.

dbLayerNextInLEF

`dbLayerNextInLEF layerPtr`

Returns the address of the layer in the LEF file after the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 39> dbLayerNextInLEF [dbGetLayerByName METAL2]  
0x4a78478
```

The address of the next layer in the LEF file is 0x4a78478.

dbLayerOffset

`dbLayerOffset layerPtr`

Returns the layer offset for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 40> dbLayerOffset [dbGetLayerByName METAL2]  
500
```

The layer offset is 0.5.

dbLayerPrefDir

```
dbLayerPrefDir layerPtr
```

Returns the preferred direction for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 41> dbLayerPrefDir [dbGetLayerByName METAL2]  
dbcV
```

The preferred direction for *metal2* is vertical.

dbLayerRes

dbLayerRes *layerPtr*

Returns the resistance for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 42> dbLayerRes [dbGetLayerByName METAL2]  
0.53
```

The resistance for *metal2* is 0.53.

dbLayerResistanceTable

dbLayerResistanceTable {*layerName* | *layerPtr*}

Returns the resistance table values for the specified metal layer. You can specify the layer by name or by address.

Parameters

<i>layerName</i>	Name of the metal layer.
<i>layerPtr</i>	Address of the metal layer.

Command Order

Use this command after importing the design.

Examples

```
innovus 42> dbLayerResistanceTable METAL1
0.1161 0.134 0.1082 0.176 0.1067 0.188 0.1009 0.258
```

dbLayerRule2ndLowerWidthRange

```
dbLayerRule2ndLowerWidthRange layerRulePtr
```

Returns the minimum width of a second range for the specified layer rule. Defined in the LEF file as:
RANGE *minWidth maxWidth*.

Parameters

<i>layerRulePtr</i>	Address of layer rule.
---------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRule2ndUpperWidthRange

`dbLayerRule2ndUpperWidthRange layerRulePtr`

Returns the maximum width of a second range for the specified layer rule. Defined in the LEF file as: RANGE *minWidth maxWidth*.

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleArrNext

`dbLayerRuleArrNext layerRulePtr`

Returns the address of the next layer rule.

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleExtension

`dbLayerRuleExtension layerRulePtr`

Returns the extension for the specified layer rule. Defined in the LEF file as: WIREEXTENSION.

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleHasExtension

`dbLayerRuleHasExtension layerRulePtr`

Returns 1 if the specified layer rule has an extension and 0 if it does not. Defined in the LEF file as: WIREEXTENSION.

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after design import.

dbLayerRuleHasRange

`dbLayerRuleHasRange layerRulePtr`

Returns `1` if the specified layer rule has a range and `0` if it does not. Defined in the LEF file as: `RANGE minWidth maxWidth`.

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleHasSecondRange

`dbLayerRuleHasSecondRange layerRulePtr`

Returns `1` if there is a second range for the specified layer rule and `0` if there is not. Defined in the LEF file as: `RANGE minWidth maxWidth`.

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleHasThreshold

dbLayerRuleHasThreshold *layerRulePtr*

Returns a 1 if the specified layer rule has a threshold, else a 0. Defined in LEF as follows:
LENGTHTHRESHOLD *maxLength*.

Parameters

<i>layerRulePtr</i>	Specifies
---------------------	-----------

Command Order

Use this command after importing the design.

dbLayerRuleHasWidth

dbLayerRuleHasWidth *layerRulePtr*

Returns a 1 if the specified layer rule has width, else a 0. Defined in the LEF file as: WIDTH *width*.

Parameters

<i>layerRulePtr</i>	Address of layer rule.
---------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleLayer

`dbLayerRuleLayer layerRulePtr`

Returns the layer for the specified layer rule. Defined in the LEF file as: `LAYER layerName`.

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleLengthThreshold

`dbLayerRuleLengthThreshold layerRulePtr`

Returns the length of the threshold for the specified layer rule. Defined in LEF as follows:

`USELENGTHTHRESHOLD`

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleLowerWidthRange

`dbLayerRuleLowerWidthRange layerRulePtr`

Returns the lower width range for the specified layer rule. Defined in the LEF file as: `RANGE minWidth maxWidth`.

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleMinSpacing

`dbLayerRuleMinSpacing layerRulePtr`

Returns the minimum spacing for the specified layer rule. Defined in the LEF file as: `SPACING minSpacingValue`.

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbLayerRuleMinSpacing
```

dbLayerRuleMinWidth

```
dbLayerRuleMinWidth layerRulePtr
```

Returns the minimum width for the specified layer rule. Defined in the LEF file as: `WIDTH width.`

Parameters

<code>layerRulePtr</code>	Address of layer rule.
---------------------------	------------------------

Command Order

Use this command after importing the design.

dbLayerRuleNext

```
dbLayerRuleNext layerRulePtr
```

Returns the address of the next layer rule, 0x0 if there is none.

Parameters

<i>layerRulePtr</i>	Address of layer rule.
---------------------	------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 84> dbLayerRuleNext $objPtr  
0x3d6d1a4
```

The next layer rule address is 0x3d6d1a4.

dbLayerRuleUpperWidthRange

```
dbLayerRuleUpperWidthRange layerRulePtr
```

Returns the upper width (maximum width) for the specified layer rule. Defined in LEF as follows:

RANGE *minWidth maxWidth*

Parameters

<i>layerRulePtr</i>	Specifies
---------------------	-----------

Command Order

Use this command after importing the design.

dbLayerShapeLayer

dbLayerShapeLayer *layerShapePtr*

Returns the address of the layer for the specified layer shape.

Parameters

<i>layerShapePtr</i>	Address of the layer shape.
----------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one via.

Example

```
proc printOneLayerShape {ls} {
    if {[dbIsLayerShapeShapeList $ls]} {
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
        dbForEachLayerShapeShape $ls s {
            printOneShape $s
        }
    } elseif {[dbIsLayerShapeVia $ls]} {
        puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
[dbLayer
ShapeViaViaLoc $ls]"
    } else {
        puts "*** error *** layerShape"
    }
}
```

dbLayerShapeType

`dbLayerShapeType layerShapePtr`

Returns the address of the type for the specified layer shape.

Parameters

<code>layerShapePtr</code>	Address of the layer shape.
----------------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one via.

dbLayerShapeViaViaCell

`dbLayerShapeViaViaCell layerShapePtr`

Returns the address of the via cell for the specified via layer shape.

Parameters

<code>layerShapePtr</code>	Address of the layer shape.
----------------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one via.

Example

```
proc printOneLayerShape {ls} {
    if {[dbIsLayerShapeShapeList $ls]} {
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
        dbForEachLayerShapeShape $ls s {
            printOneShape $s
        }
    } elseif {[dbIsLayerShapeVia $ls]} {
        puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
        [dbLayer
        ShapeViaViaLoc $ls]"
    } else {
        puts "*** error *** layerShape"
    }
}
```

dbLayerShapeViaViaLoc

`dbLayerShapeViaViaLoc layerShapePtr`

Returns the address of the via cell location for the specified via layer shape.

Parameters

<code>layerShapePtr</code>	Address of the layer shape.
----------------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one via.

Example

```
proc printOneLayerShape {ls} {
    if {[dbIsLayerShapeShapeList $ls]} {
        puts "layerShape is shapeList on layer [dbLayerName [dbLayerShapeLayer $ls]]"
        dbForEachLayerShapeShape $ls s {
            printOneShape $s
        }
    } elseif {[dbIsLayerShapeVia $ls]} {
        puts "layerShape is via viaCell [dbViaCellName [dbLayerShapeViaViaCell $ls]] loc
[dbLayer
ShapeViaViaLoc $ls]"
    } else {
        puts "*** error *** layerShape"
    }
}
```

dbLayerSpace

dbLayerSpace *layerPtr*

Returns the space set for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 44> dbLayerSpace [dbGetLayerByName METAL2]  
400
```

The space for *metal2* is 0.4.

dbLayerThickness

dbLayerThickness *layerPtr*

Returns the thickness for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 45> dbLayerThickness [dbGetLayerByName METAL2]  
570
```

The layer thickness is 570.

dbLayerWireId

dbLayerWireId *layerPtr*

Returns the ID for the specified layer.

Parameters

<i>layerPtr</i>	Address of layer
-----------------	------------------

Command Order

Use this command after importing the design.

Example

```
innovus 70> dbLayerWireId [dbGetLayerByName M2]  
2
```

The wire ID for *metal2* is 2.

dbLayerWirePitch

dbLayerWirePitch *layerPtr*

Returns the pitch for the specified layer.

Parameters

<i>layerPtr</i>	Specifies
-----------------	-----------

Command Order

Use this command after importing the design.

Example

```
innovus 46> dbLayerWirePitch [dbGetLayerByName METAL2]  
1000
```

The wire pitch is 1.0.

dbLayerWireSpace

`dbLayerWireSpace layerPtr`

Returns the wire spacing for the specified layer.

Parameters

<code>layerPtr</code>	Address of the layer.
-----------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 47> dbLayerWireSpace [dbGetLayerByName METAL2]  
400
```

The wire space is 0.4.

dbLayerWireWidth

dbLayerWireWidth *layerPtr*

Returns the wire width for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 48> dbLayerWireWidth [dbGetLayerByName METAL2]  
400
```

The wire width is 0.4.

dbLocPtrLoc

dbLocPtrLoc *locPtr*

Returns address of master location pointer.

Parameters

<i>locPtr</i>	Address of location.
---------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 61> dbLocPtrLoc $objPtr  
0x7c8db90
```

The address of the master location is 0x7c8db90.

dbLocPtrNext

`dbLocPtrNext locPtr`

Returns the address of the next location, 0x0 if there is none.

Parameters

<code>locPtr</code>	Address of location.
---------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 93> dbLocPtrNext $objPtr  
0x7c8e098
```

The address of the next location is 0x7c8e098.

dbLocX

dbLocX *loc_coords*

Returns the x location for the specified coordinates.

Parameters

<i>loc_coords</i>	Coordinates.
-------------------	--------------

Command Order

Use this command after importing the design.

Example

```
innovus 60> dbHeadBox  
0 0 1249000 1350000  
innovus 61> dbLocX [dbHeadBox]  
0
```

The x location is 0.

dbLocY

dbLocY *loc_commands*

Returns the y coordinate for the specified coordinates.

Parameters

loc_commands

Coordinates.

Command Order

Use this command after importing the design.

Example

```
innovus 60> dbHeadBox  
0 0 1249000 1350000  
innovus 62> dbLocY [dbHeadBox]  
0
```

The y location is 0.

M

dbMarkCriticalNet

`dbMarkCriticalNet thresh (ns)`

Marks critical nets with the specified threshold.

Parameters

thresh

Floating point number

Command Order

Use this command after importing the design.

Example

```
innovus 14> dbMarkCriticalNet 0.3
```

The threshold is set to 0.3 for critical nets.

dbMaskPtnPinLayerOnSide

```
dbMaskPtnPinLayerOnSide ptnPtr side layerId
```

Blocks the layer specified for pin assignment of the specified side.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>side</i>	Side.
<i>layerId</i>	Layer ID

Command Order

Use this command after specifying partitions.

Example

```
innovus 14> dbMaskPtnPinLayerOnSide [dbGetPtnByName sheet7] W 3
```

The pins on the west side of the partition sheet7 are blocked on metal3.

dbmCellGetXNetByName

```
dbmCellGetXNetByName cellPtr name
```

Returns the address of the duplicate net of the specified cell by the name

Parameters

<i>cellPtr</i>	Address of cell.
<i>name</i>	Name of net.

Command Order

Use this command after importing the design.

Example

```
innovus 1>dbmCellGetXNetByName [dbgTopCell] BLU  
0x83b5086b
```

The duplicate net address is 0x83b5086b.

dbMergeNetRC

dbMergeNetRC *fromTermPtr* *toTermPtr*

Merges the nets specified by the *fromTermPtr* to the *toTermPtr*.

Parameters

<i>fromTermPtr</i>	Address of term.
<i>toTermPtr</i>	Address of term.

Command Order

Use this command after the design has been extracted.

Example

```
innovus 1>dbMergeNetRC $fromterm $toterms
```

The RC values for nets connected from the "from" term to the "to" term are combined.

dbMergeNetWires

```
dbMergeNetWires fromNetPtr toNetPtr
```

Merges the nets specified by the *fromNetPtr* to the *toNetPtr*.

Parameters

<i>fromNetPtr</i>	Address of net.
<i>toNetPtr</i>	Address of net.

Command Order

Use this command after importing the design.

Example

```
innovus 1>dbMergeNetWires $oldNet $newNet
```

```
innovus 2>
```

The nets specified by the old net address are merged with the nets specified by the new address.

dbMicronsToDBU

```
dbMicronsToDBU number
```

Returns the number in database units.

Parameters

<i>number</i>	Number.
---------------	---------

Command Order

Use this command after importing the design.

Example

```
innovus 1> dbMicronsToDBU 5.85  
5850
```

The number of database units for 5.85 is 5,850.

dbMinDistLocToWire

```
dbMinDistLocToWire loc_x loc_y wirePtr
```

Returns the minimum distance for the specified wire at the given location.

Parameters

<i>loc_x</i>	Number.
<i>loc_y</i>	Number.
<i>wirePtr</i>	Address of wire.

Command Order

Use this command after routing the design.

Example

```
innovus 1>dbForEachNetWire $net wirePtr {  
    set mindis [dbMinDistLocToWire 1000 1000 $wirePtr]  
    puts "min distance = $mindis"  
"  
  
min distance = 10969285  
min distance = 10969285  
min distance = 12039000  
min distance = 12039000
```

The minimum distance for wires for the specified net are returned.

dbMultiplyXCap

`dbMultiplyXCap netPtr aggressor_netPtr float`

Scales the capacitance between the net and the aggressor net.

Parameters

<code><i>netPtr</i></code>	Address of net.
<code><i>aggressor_netPtr</i></code>	Address of net.
<code><i>float</i></code>	Value to multiply to capacitance.

Command Order

Use this command after extraction.

Example

```
innovus 1>dbMultiplyXcp [dbGetNetByName BLU] [dbGetNetByName A\[1\]] 0.5
```

N

dbNameToObj

`dbnametooobj objPtr`

Returns the name of the object.

Parameters

<code><i>objPtr</i></code>	Address of the object.
----------------------------	------------------------

Command Order

Use this command after importing the design.

dbNameToType

`dbNameToType objPtr`

Returns the name of the type of the specified object

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

dbNetBaseName

dbNetBaseName *netPtr*

Returns the base name of the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 3> set s [dbHeadSelList]
```

```
0x965aa20
```

```
innovus 4> set objPtr [dbSelPtr $s]
```

```
0x54be2f8
```

```
innovus 5> dbNetBaseName $objPtr
```

```
BLU
```

The base name is BLU.

dbNetBox

dbNetBox *netPtr*

Returns the box of the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design (the result is more meaningful after route).

Example

```
innovus 6> dbNetBox $objPtr  
45600 751750 1008200 763620
```

The lower left coordinate of the net box is (45.6 751.75) and the upper right coordinate is (1008.2 763.62).

dbNetBus

dbNetBus *netPtr*

Returns the address of the bus specified by the net. If there is no bus, the command returns 0x0.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
dbNetBus $objPtr
```

```
0x0
```

The net is not a bus.

dbNetCap

```
dbNetCap netPtr
```

Returns the capacitance of the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 11> dbNetCap $objPtr
```

```
0.0
```

The capacitance is 0.0.

dbNetChangeWireGlobalStatus

`dbNetChangeWireGlobalStatus netPtr oldStatus newStatus`

Changes the global status of the specified wire. Possible values are:

`dbcNoWire`

`dbcRoutedWire`

`dbcFixedWire`

`dbcCoverWire`

`dbcShieldWire`

`dbcNoShieldWire`

`dbcGlobalWire`

Parameters

<code>netPtr</code>	Address of net.
<code>oldStatus</code>	Status.
<code>newStatus</code>	Status.

Command Order

Use this command after route.

Example

```
innovus 62>dbNetChangeWireGlobalStatus $netPtr dbcRoutedWire dbcFixedWire
innovus 63>
```

The net is changed from Routed to Fixed.

dbNetChangeWireStatus

`dbNetChangeWireStatus netPtr oldStatus newStatus`

Changes the status of the specified wire. Possible values for status are:

`dbcNoWire`

`dbcRoutedWire`

`dbcFixedWire`

`dbcCoverWire`

`dbcShieldWire`

`dbcNoShieldWire`

`dbcGlobalWire`

Parameters

<code>netPtr</code>	Address of net.
<code>oldStatus</code>	Status.
<code>newStatus</code>	Status.

Command Order

Use this command after route.

Example

```
encounter 62>dbNetChangeWireStatus $netPtr dbcGlobalWire dbcRoutedWire
encounter 63>
```

The status is changed from global route to routed.

dbNetCleanupWires

`dbNetCleanupWires netPtr`

Corrects overlapping, crossing, and cycling wires for the specified net.

Parameters

<code>netPtr</code>	Address of net.
---------------------	-----------------

Command Order

Use this command after route.

dbNetDate

`dbNetDate netPtr`

For debugging purposes only.

dbNetDefName

`dbNetDefName netPtr`

Returns the name as defined by DEF of the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 13> dbNetDefName $objPtr
```

```
BLU
```

The DEF name for the net is `BLU`.

dbNetDeleteAggrNearestWire

```
dbNetDeleteAggrNearestWire netPtr(victim) netPtr(aggressor)
```

Deletes the specified aggressor net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after signal integrity (SI) analysis.

Example

```
innovus 13> dbNetDeleteAggrNearestWire [dbGetNetByName n1] [dbGetNetByName n2]  
innovus 14>
```

The net `n2` is deleted.

dbNetExtrRCSummary

`dbNetExtrRCSummary netPtr`

Returns count of capacitance, total capacitance, count of coupling capacitance, total of coupling caps, count of resistances, total resistance.

Parameters

<code>netPtr</code>	Address of the net.
---------------------	---------------------

Command Order

Use this command after extracting the design.

dbNetExtRule

`dbNetExtRule netPtr`

Returns the address of the next extension rule.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 13> dbNetExtRule $objPtr  
0x0
```

There is no next extension rule.

dbNetFreeWires

```
dbNetFreeWires netPtr
```

Frees wires for the specified net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after route.

Example

```
innovus 2> dbNetFreeWires [dbGetNetByName BLU]  
innovus 3>
```

The wires are freed on the `BLU` net.

dbNetFreeWires2

`dbNetFreeWires2 netPtr`

Frees wires for the specified net and cleans up the net terminal references to the wires. Cadence recommends using this command instead of the `dbNetFreeWires` command.

Parameters

<code>netPtr</code>	Address of net.
---------------------	-----------------

Command Order

Use this command after routing the design.

Example

```
innovus 3> dbNetFreeWires2 [dbGetNetByName BLU]  
innovus 4>
```

The wires are freed and cleaned up for the net `BLU`.

dbNetFrequency

`dbNetFrequency netPtr`

Returns the frequency of the specified net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after performing timing analysis.

dbNetGetCritDrivingTimeArc

`dbNetGetCritDrivingTimeArc netPtr riseFall maxMin`

Returns the address of the critical rise/fall and max/min for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
<i>rise/fall</i>	1 implies rise; 0 implies fall
<i>maxMin</i>	1 implies max; 0 implies min

Command Order

Use this command after performing timing analysis.

Example

```
innovus 71>dbNetGetCritDrivingTimeArc [dbGetNetByName BLU] 1 0  
0x83ba78a2 0x78b9fd32
```

The critical rise/fall timing arc address is 0x83ba78a2, and the critical max/min address is

0x78b9fd32.

dbNetGlobalWireAddVias

dbNetGlobalWireAddVias *netPtr*

Changes the global wire specified separate wires and vias. For example, a wire that goes directly from *METAL1* to *METAL5* is split up into wires and vias that go from *METAL1* to *METAL2*, *METAL2* to *METAL3*, and so on.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 71> dbNetGlobalWireAddVias [dbGetNetByName BLU]  
1
```

Separate wires and vias are created for the net `BLU`.

dbNetGlobalWireToUsableWire

dbNetGlobalWireToUsableWire *netPtr*

Changes the specified global wire so that it can be used in the Innovus software for extraction, save, restore, and so forth. However, it can not be changed back and loaded into NanoRoute for detail routing.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 71> dbNetGlobalWireToUsableWire [dbGetNetByName BLU]
```

```
1
```

The wire is changed and can now be extracted.

dbNetGroupMsAttr

```
dbNetGroupMsAttr netGroupPtr
```

Returns the attributes for the specified mixed signal net group.

Parameters

<i>netGroupPtr</i>	Address of the net group.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbNetGroupMsAttr $netGroupPtr
```

dbNetGroupMsCoaxialLayers

```
dbNetGroupMsCoaxialLayers netGroupPtr
```

Returns the coaxial layers for the specified mixed signal net group.

Parameters

<i>netGroupPtr</i>	Address of the net group.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbNetGroupMsCoaxialLayers $netGroupPtr
```

dbNetGroupMsMatchOrNot

```
dbNetGroupMsMatchOrNot netGroupPtr
```

Returns a 1 if there is a match or not for the specified mixed signal net group.

Parameters

<i>netGroupPtr</i>	Address of the net group.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbNetGroupMsMatchOrNot $netGroupPtr
```

dbNetGroupMsOverlap

```
dbNetGroupMsOverlap netGroupPtr
```

Returns a 1 if there is an overlap for the specified mixed signal net group.

Parameters

<i>netGroupPtr</i>	Address of the net group.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbNetGroupMsOverlap $netGroupPtr
```

dbNetGroupMsResistance

dbNetGroupMsResistance *netGroupPtr*

Returns the resistance for the specified mixed signal net group.

Parameters

<i>netGroupPtr</i>	Address of the net group.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbNetGroupMsResistance $netGroupPtr
```

dbNetGroupMsShieldNet

dbNetGroupMsShieldNet *netGroupPtr*

Returns the shield net for the specified mixed signal net group.

Parameters

<i>netGroupPtr</i>	Address of the net group.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbNetGroupMsShieldNet $netGroupPtr
```

dbNetGroupMsShieldWidth

```
dbNetGroupMsShieldWidth netGroupPtr
```

Returns the shield width for the specified mixed signal net group.

Parameters

<i>netGroupPtr</i>	Address of the net group.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbNetGroupMsShieldWidth $netGroupPtr
```

dbNetGroupMsSpacing

```
dbNetGroupMsSpacing netGroupPtr
```

Returns the spacing for the specified mixed signal net group.

Parameters

netGroupPtr

Address of the net group.

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
dbNetGroupMsSpacing $netGroupPtr
```

dbNetGroupMsThreshold

`dbNetGroupMsThreshold netGroupPtr`

Returns the threshold for the specified mixed signal net group.

Parameters

netGroupPtr

Address of the net group.

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
innovus 71> dbNetGroupMsThreshold $netGroupPtr
```

dbNetGroupMsTolerance

`dbNetGroupMsTolerance netGroupPtr`

Returns the tolerance for the specified mixed signal net group.

Parameters

<i>netGroupPtr</i>	Address of the net group.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
innovus 71> dbNetGroupMsTolerance $netGroupPtr
```

dbNetGroupMsWidth

`dbNetGroupMsWidth netGroupPtr`

Returns the width for the specified mixed signal net group.

Parameters

<i>netGroupPtr</i>	Address of the net group.
--------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one net group.

Example

```
innovus 71> dbNetGroupMsWidth $netGroupPtr
```

dbNetGroupName

`dbNetGroupName netGroupPtr`

Returns the name for the specified net group.

Parameters

<code><i>netGroupPtr</i></code>	Address of the net group.
---------------------------------	---------------------------

Command Order

Use this command after importing the design and creation of at least one net group.

Example

```
innovus 71> dbNetGroupName $netGroupPtr
```

```
myNetGroup1
```

The net group name is `myNetGroup1`.

dbNetHorWt

`dbNetHorWt netPtr`

Returns the horizontal net weight for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 71> dbNetHorWt [dbGetNetByName BLU]  
1
```

The horizontal net weight is 1.

dbNetLenX

```
dbNetLenX netPtr
```

Returns the horizontal net length for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 15> dbNetLenX $objPtr  
962600
```

The horizontal net length is 962,600.

dbNetLenY

`dbNetLenY netPtr`

Returns the y net length for the specified net.

Parameters

<code><i>netPtr</i></code>	Address of the net.
----------------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 16> dbNetLenY $objPtr  
11870
```

The y net length is 11,870.

dbNetLogicValue

`dbNetLogicValue netPtr`

Returns the logic value for the net specified.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> dbNetLogicValue $objPtr  
0
```

The logic value is 0.

dbNetMaxCapSlack

```
dbNetMaxCapSlack netPtr
```

Returns the maximum capacitance slack for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after timing analysis.

Example

```
innovus 18> dbNetName [dbGetNetByName n_7268  
0.0
```

The maximum capacitance slack for net n_7268 is 0.0.

dbNetMaxFanoutSlack

`dbNetMaxFanoutSlack netPtr`

Returns the maximum fanout slack for the specified net.

Parameters

<code><i>netPtr</i></code>	Address of the net.
----------------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 18> dbNetName [dbGetNetByName n_7268]  
0.0
```

The maximum fanout slack for the net n_7268 is 0.0.

dbNetMaxTranSlack

`dbNetMaxTranSlack netPtr`

Returns the maximum transition for the net specified.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 18> dbNetMaxTranTranSlack [dbGetNetByName n_7268]
```

```
0.5
```

The maximum transition slack for the net n_7268 is 0.5.

dbNetName

```
dbNetName netPtr
```

Returns the net name for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 18> dbNetName $objPtr  
BLU
```

The name of the net is BLU.

dbNetNext

`dbNetNext netPtr`

Returns the address of the next net.

Parameters

<code><i>netPtr</i></code>	Address of the net.
----------------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 19> dbNetNext $objPtr  
0x54be34c
```

The next net address is 0x54be34c.

dbNetNrFanIn

`dbNetNrFanIn netPtr`

Returns the number of fanins for the specified net.

Parameters

<i>netPtr</i>	Address of specified net.
---------------	---------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 20> dbNetNrFanIn $objPtr
```

```
2
```

There are two fanins.

dbNetNrFanOut

```
dbNetNrFanOut netPtr
```

Returns the number of fanouts.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 21> dbNetNrFanOut $objPtr  
1
```

There is one fanout.

dbNetNrOutTerm

`dbNetNrOutTerm netPtr`

Returns the number of output terminals for the specified net.

Parameters

<code><i>netPtr</i></code>	Address of the net.
----------------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 22> dbNetNrOutTerm $objPtr  
2
```

There are two output terminals.

dbNetNrTerm

`dbNetNrTerm netPtr`

Returns the total number of terminals for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 23> dbNetNrTerm $objPtr
```

```
3
```

There are three terminals on the net.

dbNetPCell

```
dbNetPCell netPtr
```

Returns the address of the physical cell for the specified net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 24> dbNetPCell $objPtr  
0x72b0a28
```

The address of the physical cell is 0x72b0a28.

dbNetPdefName

```
dbNetPdefName netPtr
```

Returns the pdef name of the specified net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 72> dbNetPdefName [dbGetNetByName BLU]  
BLU
```

The pdef name is BLU.

dbNetPower

```
dbNetPower netPtr
```

Returns the power of the specified net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 72> dbNetPower [dbGetNetByName int]  
0.0
```

The power of the net `int` is 0.0.

dbNetPrefExtraSpace

```
dbNetPrefExtraSpace netPtr
```

Returns a 1 if the specified net has extra space, else a 0.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 72>dbNetPrefExtraSpace [dbGetNetByName BLU]  
0
```

The net does not have extra space.

dbNetPrefixId

`dbNetPrefixId netPtr`

Returns the prefix ID for the specified net.

Parameters

<code><i>netPtr</i></code>	Address of net.
----------------------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 72> dbNetPrefixId [dbGetNetByName BLU]  
0
```

The prefix ID is 0.

dbNetPrefixName

`dbNetPrefixName netPtr`

Returns the prefix for the specified net. If there is no prefix, the command returns no result.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 73> dbNetPrefixName [dbGetNetByName BLU]  
innovus 74>
```

There is no prefix name.

dbNetPrev

dbNetPrev *netPtr*

Returns the address of the previous net. If there is no previous net, the command returns 0x0.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 26> dbNetPrev $objPtr  
0x0
```

There is no previous net.

dbNetPrintWires

```
dbNetPrintWires netPtr
```

Returns all of the information for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design, but is more meaningful after routing.

Example

```
innovus 28> dbNetPrintWires $objPtr
```

Wires for net BLU

```
Wire0: 0xa69d198: (45800 763220 2) net BLU:  
Dir=H head=0xa6421b8 len=1000 term=1 0x0 s=dbcNoWire Adj=ED(11+1) rid=0 vid=4  
Wire1: 0xa69d204: (45800 763220 1) net BLU:  
Dir=H head=0xa8f9714 len=600 term=1 0x0 s=dbcNoWire Adj=EU(9+0) rid=0 vid=0
```

```
Term0: 0x7c8f168: (46400 763220 0)BLU
Term1: 0x8b4c5f8: (46400 763220 1)SH15/I194:BLU
Term2: 0x7fc4350: (1007300 751950 1)SH22/I40:DACBLU
```

The information for net BLU is returned.

dbNetSwitchingPower

dbNetSwitchingPower *netPtr*

Returns the switching power of the specified net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 72> dbNetSwitchingPower [dbGetNetByName int]
0.0
```

The switching power of the net int is 0.0.

dbNetTermList

dbNetTermList *netPtr*

Returns the address of the list of terminals for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 37> dbNetTermList $objPtr  
0x7c8f168
```

The address for the list of terms is 0x7c8f168.

dbNetToggleDensity

dbNetToggleDensity *netPtr*

Returns the toggle density for the specified net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after routing the design.

Examples

```
innovus 4> dbNetToggleDensity [dbGetNetByName ibias]  
0.0
```

The toggle density is 0.0.

dbNetUserWt

```
dbNetUserWt netPtr
```

Returns the net weight for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 38> dbNetUserWt $objPtr  
2
```

The net weight is 2.

dbNetVerWt

dbNetVerWt *netPtr*

Returns the vertical net weight for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 39> dbNetVerWt $objPtr  
1
```

The vertical net weight is 1.

dbNetWidthX

dbNetWidthX *netPtr*

Returns the width for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 40> dbNetWidthX $objPtr  
1
```

The width is 1.

dbNetWireLenX

```
dbNetWireLenX netPtr
```

Returns the x wire length for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after routing.

Examples

```
innovus 40> dbNetWireLenX [dbGetNetByName ibias]  
1
```

The wire length is 1.

dbNetWireLenY

dbNetWireLenY *netPtr*

Returns the y wire length for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Examples

```
innovus 40> dbNetWireLengthY [dbGetNetByName ibias]
```

```
1
```

The width is 1.0.

dbNetWireList

dbNetWireList *netPtr*

Returns the address of the list of wires for the specified net.

Parameters

<i>netPtr</i>	Address of the net.
---------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 41> dbNetWireList $objPtr  
0xa69d198
```

The address of the list of wires is 0xa69d198.

dbNewFPin

```
dbNewFPin ftermPtr
```

Creates a new FPin for the specified f-term.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

Command Order

Use this command after importing the design.

Example

```
innovus 22> dbNewFPin [dbGetFTermByName BLU]  
0x9c7d840
```

A new FPin is created.

dbNewHInst

dbNewHInst *hinstName*

Creates a new hierarchical instance for the specified hierarchical instance.

Parameters

<i>hinstName</i>	Address of hierarchical instance.
------------------	-----------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 24> dbNewHInst [dbGetHInstByName SH17]
```

```
0x8a652a0
```

A new hierarchical instance is created.

O

dbObjBox

dbObjBox *objPtr*

Returns the object box. If the box is not inside the core area, the command returns 2147483647 2147483647 -2147483648 -2147483648.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 59> dbObjBox $objPtr  
2147483647 2147483647 -2147483648 -2147483648
```

The box is not inside the core.

dbObjFPlanBox

```
dbObjFPlanBox objPtr
```

Returns the box of the specified object.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 61> dbObjFPlanBox $objPtr  
4586300 2914300 5557300 3886300
```

The floorplan box's lower left coordinate is (4586.3 2914.3) while the upper right coordinate is (5557.3 3886.3).

dbObjName

`dbObjName objPtr`

Returns the name of the specified object.

Parameters

<code><i>objPtr</i></code>	Address of the object.
----------------------------	------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 62> dbObjName $objPtr  
SH1
```

The name of the object is `SH1`.

dbObjPrint

dbObjPrint *objPtr*

Returns the object type and address of the object.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 63> dbObjPrint $objPtr
dbcObjHInst (dbType=10) : ptr= 0x9091b60
```

The object type and address is returned.

dbObjPropList

dbObjPropList *objPtr*

Returns the address of the list of properties for the specified object.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 64> dbObjPropList $objPtr  
0x0
```

There is no list of properties.

dbObjPtr0

```
dbObjPtr0 objPtr
```

Returns address of the specified object.

Parameters

<i>objPtr</i>	Address of the object.
---------------	------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 65> dbObjPtr0 $objPtr  
0x3880000a
```

The address is 0x3880000a.

dbObjPtrPlus1

dbObjPtrPlus1 *objPtr*

Returns ((PTR*)p)+1 or ((PTR*)p)+N for the specified object. Useful for getting array element.

Parameters

<i>objPtr</i>	Address of object.
---------------	--------------------

Command Order

Use this command after importing the design.

Example

```
innovus 66> dbObjPtrPlus1 $objPtr
0x9091b64
```

The address is 0x9091b64.

dbObjToName

dbObjToName *objPtr*

Returns the name of the specified object.

Parameters

<i>objPtr</i>	Address of object.
---------------	--------------------

Command Order

Use this command after importing the design.

Example

```
innovus 67>dbObjToName $netPtr  
reset
```

The name of the object is `reset`.

dbObstructBox

`dbObstructBox obstructPtr`

Returns the box for the specified obstruction.

Parameters

<code>obstructPtr</code>	Address of the obstruction.
--------------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one obstruction.

Example

```
innovus 71> dbObstructBox $objPtr  
1983300 3670300 2221300 3913300
```

The lower left coordinate of the obstruction box is (1983.3 3670.3) while the upper right coordinate is (2221.3 3913.3).

dbObstructName

dbObstructName *obstructPtr*

Returns the name of the specified obstruction.

Parameters

<i>obstructPtr</i>	Address of the obstruction.
--------------------	-----------------------------

Command Order

Use this command after importing the design and creating at least one obstruction.

Example

```
innovus 72> dbObstructName $objPtr
defObstructName
```

The name of the obstruction is `defObstructName`.

dbObstructNext

dbObstructNext *obstructPtr*

Returns the address of the next obstruction.

Parameters

obstructPtr

Address of the obstruction.

Command Order

Use this command after importing the design and creating at least one obstruction.

Example

```
innovus 73> dbObstructNext $objPtr  
0x96d81f0
```

The next obstruction is at address 0x96d81f0.

dbObstructType

dbObstructType *obstructPtr*

Returns the type of obstruction for the specified obstruction. Possible values are:

```
dbcUserObstruct 0  
dbcBlockObstruct 1  
(dbcDynamicObstruct 2)  
(dbcPtnCutObstruct 3)
```

Parameters

obstructPtr

Address of the obstruction.

Command Order

Use this command after importing the design and creating at least one obstruction.

Example

```
innovus 74> dbObstructType $objPtr  
dbcUserObstruct
```

The type of obstruct is user.

dbOpCondName

dbOpCondName *opCondPtr*

Returns the operating condition name for the specified operating condition.

Parameters

<i>opCondPtr</i>	Address of the operating condition.
------------------	-------------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> dbOpCondName opCondPtr  
BEST
```

The name of the operating condition is BEST.

dbOpCondNext

dbOpCondNext *opCondPtr*

Returns the address of the next operating condition.

Parameters

<i>opCondPtr</i>	Address of the operating condition.
------------------	-------------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> dbOpCondNext $opCondPtr
0x96d81d0
```

The next operating condition address is 0x96d81d0.

dbOpCondProc

```
dbOpCondProc opCondPtr
```

Returns the process for the specified operating condition.

Parameters

<i>opCondPtr</i>	Address of the operating condition.
------------------	-------------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> dbOpCondProc $opCondPtr  
1.0
```

The process is 1.0.

dbOpCondTemp

```
dbOpCondTemp opCondPtr
```

Returns the temperature for the specified operating condition.

Parameters

<i>opCondPtr</i>	Address of the operating condition.
------------------	-------------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 1> dbOpCondTemp $opCondPtr  
100.0
```

The operating condition temperature is 100.0.

dbOpCondVolt

```
dbOpCondVolt opCondPtr
```

Returns the voltage for the specified operating condition.

Parameters

<i>opCondPtr</i>	Address of the operating condition.
------------------	-------------------------------------

Command Order

Use this command after importing the design.

P

dbPathFallDelay

`dbPathFallDelay pathPtr`

Returns the path fall delay for the specified path.

Parameters

<i>pathPtr</i>	Address of path.
----------------	------------------

Command Order

Use this command after timing analysis.

Example

```
set s [dbHeadSelList]
```

```
0x9c2d730
innovus 29> set objPtr [dbSelPtr $s]
0x9b8f970
innovus 30> set obj_type [dbObjType $objPtr]
dbcObjPath
innovus 31> dbPathFallDelay $objPtr
138833
```

The path fall delay is 138.833.

dbPathId

`dbPathId pathPtr`

Returns the ID for the specified path.

Parameters

<code><i>pathPtr</i></code>	Address of path.
-----------------------------	------------------

Command Order

Use this command after timing analysis.

Example

```
innovus 32> dbPathId $objPtr
1
```

The path ID is 1.

dbPathLastTerm

dbPathLastTerm *pathPtr*

Returns the address of the last term for the specified path.

Parameters

<i>pathPtr</i>	Address of path.
----------------	------------------

Command Order

Use this command after timing analysis.

Example

```
innovus 33> dbPathLastTerm $objPtr
0x9b7bf00
```

The address of the last term is 0x9b7bf00.

dbPathNext

dbPathNext *pathPtr*

Returns the address of the next path, 0x0 if there is none.

Parameters

<i>pathPtr</i>	Address of path.
----------------	------------------

Command Order

Use this command after timing analysis.

Example

```
innovus 34> dbPathNext $objPtr  
0x0
```

There is no next address.

dbPathRiseDelay

`dbPathRiseDelay pathPtr`

Returns the rise delay for the specified path.

Parameters

<code><i>pathPtr</i></code>	Address of path.
-----------------------------	------------------

Command Order

Use this command after timing analysis.

Example

```
innovus 35> dbPathRiseDelay $objPtr  
141362
```

The rise delay is 141,362.

dbPathTermList

dbPathTermList *pathPtr*

Returns the address of the list of terms for the specified path.

Parameters

<i>pathPtr</i>	Address of path.
----------------	------------------

Command Order

Use this command after timing analysis.

Example

```
innovus 36> dbPathTermList $objPtr
0x9b7bc90
```

The address of the term list is 0x9b7bc90.

dbPerimBox

dbPerimBox *perimPtr*

Returns the box specified by the perimeter.

Parameters

<i>perimPtr</i>	Address of perimeter.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 74> dbPerimBox $perimPtr  
3056000 2072000 4096000 3646000
```

The lower left coordinate is (3056.0 2072.0) while the upper right coordinate is (4096.0 3646.0).

dbPerimNext

```
dbPerimNext perimPtr
```

Returns the address of the next perimeter, 0x0 if there is none.

Parameters

<i>perimPtr</i>	Address of perimeter.
-----------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 173> dbPerimNext $perimPtr  
0x7f88f90
```

The next address is 0x7f88f90.

dbPerimPtList

dbPerimPtList *perimPtr*

Returns address of partition list for the specified perimeter.

Parameters

<i>perimPtr</i>	Address of perimeter.
-----------------	-----------------------

Command Order

Use this command after partitioning.

Example

```
innovus 66> dbPerimPtList $perimPtr
0x7f88f90
```

The address of the partition list is 0x7f88f90.

dbPGFTermPGTerm

dbPGFTermPGTerm *pgFTermPtr instPtr*

Returns address of term associated with the specified PG f-term.

Parameters

<i>pgFTermPtr</i>	Address of PG F-term
-------------------	----------------------

<i>instPtr</i>	Address of instance.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 66> dbPGTermPGTerm [dbGetPGTermByName [dbGetCellByName YFD1] VDD]
[dbGetInstanceByName SH17/I340]
0x7f88f90
```

The address of the term is 0x7f88f90.

dbPGTermName

```
dbPGTermName pgTermPtr instPtr
```

Returns the name of the PG term specified by the PG address of the instance.

Parameters

<i>pgTermPtr</i>	Address of PG term
<i>instPtr</i>	Address of instance

Command Order

Use this command after importing the design.

Example

```
innovus 66> dbPGTermName [dbGetPGTermByName [dbGetInstByName SH17/I340] VDD]  
[dbGetInstByName SH17/I340]  
  
VDD  
  
innovus 67>
```

The name of the term specified is VDD.

dbPGTermNet

dbPGTermNet *pgTermPtr*

Returns address of the net specified by the PG term.

Parameters

<i>pgTermPtr</i>	Address of PG term.
------------------	---------------------

Command Order

Use this command after importing the design.

Example

```
innovus 66> dbPGTermNet [dbGetPGTermByName [dbGetInstByName SH17/I340] vdd]  
0x7f88f90
```

The address of the net is 0x7f88f90.

dbPGTermPGFTerm

dbPGTermPGFTerm *pgTermPtr instPtr*

Returns address of the f-term specified by the term.

Parameters

<i>pgTermPtr</i>	Address of PG term.
<i>instPtr</i>	Address of instance.

Command Order

Use this command after importing the design.

Example

```
innovus 66> dbPGTermPGFTerm [dbGetPGTermByName [dbGetInstByName SH17/I340] vdd]  
[dbGetInstByName SH17/I340]]
```

0x7f88f90

The address of the PG f-term is 0x7f88f90.

dbPinGroupCell

dbPinGroupCell *pinGroupPtr*

Returns the address of the cell for the specified pin group.

Parameters

pinGroupPtr

Address of the pin group.

Command Order

Use this command after importing the design and creating at least one pin group.

Example

```
innovus 95> dbPinGroupCell $pinGrpPtr  
0x965aa20
```

The address of the cell containing the pin group is 0x965aa20.

dbPinGroupName

dbPinGroupName *pinGroupPtr*

Returns the name of the pin group for the specified pin group.

Parameters

pinGroupPtr

Address of the pin group.

Command Order

Use this command after importing the design and creating at least one pin group.

Example

```
innovus 1> dbPinGroupName $pinGroup  
pins
```

The name of the pin group is `pins`.

dbPinGroupNext

`dbPinGroupNext pinGroupPtr`

Returns the address of the next pin group. If there is no other pin group, the command returns 0x0.

Parameters

<code>pinGroupPtr</code>	Address of the pin group.
--------------------------	---------------------------

Command Order

Use this command after importing the design and creating at least one pin group.

dbPinGroupPinList

`dbPinGroupPinList pinGroupPtr`

Returns address of pin list for the specified pin group.

Parameters

<code>pinGroupPtr</code>	Address of pin group.
--------------------------	-----------------------

Command Order

Use this command after importing the design and creating at least one pin group.

dbPinGroupPinListTail

`dbPinGroupPinListTail pinGroupPtr`

Returns the address of the last pin in the group for the specified pin group.

Parameters

<code><i>pinGroupPtr</i></code>	Address of pin group.
---------------------------------	-----------------------

Command Order

Use this command after importing the design and creating at least one pin group.

dbPlaceDummyIoPad

`dbPlaceDummyIoPad ioPtr dbLocX dbLocY orient`

Places the dummy pad at the specified location with the specified orientation.

Parameters

<code><i>ioPtr</i></code>	Address of the I/O pad.
<code><i>dbLocX</i></code>	X coordinate.

<i>dbLocY</i>	Y coordinate.
<i>orient</i>	Orientation.

Command Order

Use this command after importing the design.

Example

```
innovus 212> dbPlaceDummyIoPad $ioPtr 1000 1000 R0  
innovus 213>
```

The dummy I/O pad is placed at (1.0 1.0) with an orientation of 0.

dbPlaceFTerm

dbPlaceFTerm ftermPtr dbLocX dbLocY layer

Places the specified f-term and the specified location.

Parameters

<i>ftermPtr</i>	Address of f-term
<i>dbLocX</i>	X coordinate.
<i>dbLocY</i>	Y coordinate.
<i>layer</i>	Metal layer.

Command Order

Use this command after importing the design.

Example

```
innovus 5> set ftermPtr [dbGetFTermByName BLU]  
0x7cb5f30  
innovus 6> dbPlaceFTerm $ftermPtr 500.0 500.0 3  
innovus 7>
```

The f-term `BLU` is placed at (500.0 500.0) on *meta13*.

dbPlaceInst

`dbPlaceInst instPtr dbLocX dbLocY orient`

Parameters

<code>instPtr</code>	Address of the instance.
<code>dbLocX</code>	X coordinate.
<code>dbLocY</code>	Y coordinate.
<code>orient</code>	Orientation.

Command Order

Use this command after importing the design.

Example

```
innovus 243> dbPlaceInst $objPtr 500.0 500.0 R90  
innovus 244>
```

The instance is placed at (500 500) with an orientation of 90.

dbPlaceInstAtOrigin

```
dbPlaceInstAtOrigin instPtr dbLocX dbLocY orient
```

Parameters

<i>instPtr</i>	Address of the instance.
<i>dbLocX</i>	X coordinate.
<i>dbLocY</i>	Y coordinate.
<i>orient</i>	Orientation.

Command Order

Use this command after importing the design.

Example

```
innovus 244> dbPlaceInstAtOrigin $objPtr 500.0 500.0 R90
innovus 245>
```

dbPowerDomainCore2Bot

dbPowerDomainCore2Bot

```
dbPowerDomainCore2Bot powerDomainPtr
```

Returns the distance from the core to the bottom of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 73> dbPowerDomainCore2Bot [dbGetPowerDomainByName VDD1]  
0
```

The distance from the core to the bottom of power domain `VDD1` is 0.

dbPowerDomainCore2Left

dbPowerDomainCore2Left

`dbPowerDomainCore2Left` *powerDomainPtr*

Returns the distance from the core to the left side of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 73> dbPowerDomainCore2Left [dbGetPowerDomainByName VDD1]  
0
```

The distance from the core to the left side of the power domain `VDD1` is 0.

dbPowerDomainCore2Right

dbPowerDomainCore2Right

`dbPowerDomainCore2Right powerDomainPtr`

Returns the distance from the core to the right side of the specified power domain.

Parameters

<code>powerDomainPtr</code>	Address of power domain.
-----------------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 73> dbPowerDomainCore2Right [dbGetPowerDomainByName VDD1]  
0
```

The distance from the core to the right side of the power domain `VDD1` is 0.

dbPowerDomainCore2Top

dbPowerDomainCore2Top

dbPowerDomainCore2Top *powerDomainPtr*

Returns the distance from the core to the top side of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 73> dbPowerDomainCore2Top [dbGetPowerDomainByName VDD1]  
0
```

The distance from the core to the top side of the power domain `VDD1` is 0.

dbPowerDomainExtGNet

dbPowerDomainExtGNet *powerDomainPtr*

Returns the address of the external ground net associated with the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachPowerDomain [dbgHead] powerDomainPtr {  
    set x [dbPowerDomainExtGNet $powerDomainPtr]  
    puts "G net = $x"  
}
```

Reports the external ground nets for the design.

dbPowerDomainExtIntGNet

```
dbPowerDomainExtIntGNet powerDomainPtr
```

Returns the address of the external interior ground net associated with the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachPowerDomain [dbgHead] powerDomainPtr {  
    set x [dbPowerDomainExtIntGNet $powerDomainPtr]  
    puts "G net = $x"  
}
```

Reports the ground nets for the design.

dbPowerDomainExtIntPNet

dbPowerDomainExtIntPNet powerDomainPtr

Returns the address of the external interior power net associated with the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachPowerDomain [dbgHead] powerDomainPtr {  
    set x [dbPowerDomainExtIntPNet $powerDomainPtr]  
    puts "G net = $x"  
}
```

Reports the power nets for the design.

dbPowerDomainExtPNet

`dbPowerDomainExtPNet powerDomainPtr`

Returns the address of the external power net associated with the specified power domain.

Parameters

<code><i>powerDomainPtr</i></code>	Address of power domain.
------------------------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachPowerDomain [dbgHead] powerDomainPtr {  
    set x [dbPowerDomainExtPNet $powerDomainPtr]  
    puts "G net = $x"  
}
```

Reports the external power nets for the design.

dbPowerDomainGNet

`dbPowerDomainGNet powerDomainPtr`

Returns the address of the ground net associated with the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachPowerDomain [dbgHead] powerDomainPtr {  
    set x [dbPowerDomainGNet $powerDomainPtr]  
    puts "G net = $x"  
}
```

Reports the ground nets for the design.

dbPowerDomainGroup

dbPowerDomainGroup

```
dbPowerDomainGroup powerDomainPtr
```

Returns the address of the group associated with the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 75> dbPowerDomainGroup [dbGetPowerDomainByName VDD1]  
0xa7d2670
```

The address for the group associated with power domain VDD1 is 0xa7d2670.

dbPowerDomainMinGapBot

dbPowerDomainMinGapBot

dbPowerDomainMinGapBot *powerDomainPtr*

Returns the minimum gap at the bottom of the specified power domain, in database units (DBU).

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 75>dbPowerDomainMinGapBot [dbGetPowerDomainByName VDD1]  
560
```

The minimum gap at the bottom of power domain `VDD1` is 560 database units.

dbPowerDomainMinGapLeft

dbPowerDomainMinGapLeft

`dbPowerDomainMinGapLeft powerDomainPtr`

Returns the minimum gap at the left side of the specified power domain, in database units (DBU).

Parameters

<code>powerDomainPtr</code>	Address of power domain.
-----------------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 75> dbPowerDomainMinGapLeft [dbGetPowerDomainByName VDD1]
```

```
560
```

The minimum gap at the left side of power domain `VDD1` is 560 database units.

dbPowerDomainMinGapRight

dbPowerDomainMinGapRight

`dbPowerDomainMinGapRight powerDomainPtr`

Returns the minimum gap at the right side of the specified power domain, in database units (DBU).

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 75> dbPowerDomainMinGapRight [dbGetPowerDomainByName VDD1]  
560
```

The minimum gap at the right side of power domain `VDD1` is 560 database units.

dbPowerDomainMinGapTop

dbPowerDomainMinGapTop

`dbPowerDomainMinGapTop` *powerDomainPtr*

Returns the minimum gap at the top side of the specified power domain, in database units (DBU).

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 75> dbPowerDomainMinGapTop [dbGetPowerDomainByName VDD1]  
560
```

The minimum gap at the top of power domain `VDD1` is 560 database units.

dbPowerDomainName

`dbPowerDomainName` *powerDomainPtr*

Returns the name of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 57> dbForEachPowerDomain [dbgHead] powerDomainPtr {  
set name [dbPowerDomainName $powerDomainPtr]  
puts "name = $name"  
}
```

GND1

The power domain is GND1.

dbPowerDomainPNet

dbPowerDomainPNet *powerDomainPtr*

Returns the address of the associated power net specified by the power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 58> dbForEachPowerDomain [dbgHead] powerDomainPtr {  
set x [dbPowerDomainPNet $powerDomainPtr]  
puts "P net = $x"  
}
```

Reports all of the power nets for the design.

dbPowerDomainRowFlip

dbPowerDomainRowFlip

dbPowerDomainRowFlip *powerDomainPtr*

Returns the orientation for the bottom row of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 75>dbPowerDomainRowFlip [dbGetPowerDomainByName VDD1]  
n
```

The rows are double-backed for the power domain VDD1.

dbPowerDomainRowSpaceType

dbPowerDomainRowSpaceType

dbPowerDomainRowSpaceType *powerDomainPtr*

Returns the type of spacing for the specified power domain. The command returns 1 if the spacing is every row and 2 if the spacing is every other row.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
dbPowerDomainRowSpaceType [dbGetPowerDomainByName VDD1]
```

Returns:

2

The row spacing for power domain `VDD1` is every other row.

dbPowerDomainRowSpacing

dbPowerDomainRowSpacing

```
dbPowerDomainRowSpacing powerDomainPtr
```

Returns the row spacing for the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
dbPowerDomainRowSpacing [dbGetPowerDomainByName VDD1]
```

Returns:

0.0

The row spacing for power domain VDD1 is 0.0.

dbPowerDomainRSExtBot

dbPowerDomainRSExtBot

```
dbPowerDomainRSExtBot powerDomainPtr
```

Returns the route search extent for the bottom of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
dbPowerDomainRSExtBot [dbGetPowerDomainByName VDD1]
```

Returns:

0

The route search extent for the bottom of power domain VDD1 is 0.

dbPowerDomainRSExtLeft

dbPowerDomainRSExtLeft

dbPowerDomainRSExtLeft *powerDomainPtr*

Returns the route search extent for the left side of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
dbPowerDomainRSExtLeft [dbGetPowerDomainByName VDD1]
```

Returns:

0

The route search extent for the left side of power domain VDD1 is 0.

dbPowerDomainRSExtRight

dbPowerDomainRSExtRight

dbPowerDomainRSExtRight *powerDomainPtr*

Returns the route search extent for the right side of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
dbPowerDomainRSExtRight [dbGetPowerDomainByName VDD1]
```

Returns:

0

The route search extent for the right side of power domain VDD1 is 0.

dbPowerDomainRSExtTop

dbPowerDomainRSExtTop

```
dbPowerDomainRSExtTop powerDomainPtr
```

Returns the route search extent for the top of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
dbPowerDomainRSExtTop [dbGetPowerDomainByName VDD1]
```

Returns:

0

The route search extent for the top of power domain VDD1 is 0.

dbPowerDomainTechSite

dbPowerDomainTechSite

```
dbPowerDomainTechSite powerDomainPtr
```

Returns the address of the tech site for the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 75>dbPowerDomainTechSite [dbGetPowerDomainByName VDD1]
```

0x73925620

The address of the tech site for power domain VDD1 is 0x73925620.

dbPowerDomainVoltage

dbPowerDomainVoltage

dbPowerDomainVoltage *powerDomainPtr*

Returns the voltage of the specified power domain.

Parameters

<i>powerDomainPtr</i>	Address of power domain.
-----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one power domain.

Example

```
innovus 75>dbPowerDomainVoltage [dbGetPowerDomainByName VDD1]  
1.62
```

The voltage for power domain VDD1 is 1.62V.

dbPowerSinkNet

dbPowerSinkNet *powersinkPtr*

Returns the address of the net associated with the specified power sink.

Parameters

<i>powersinkPtr</i>	Address of power sink.
---------------------	------------------------

Command Order

Use this command after specifying power pad locations.

Example

```
innovus 34> dbPowerSinkNet $objPtr  
0xaec1f10
```

The address of the power sink net is 0xaec1f10.

dbPowerSinkNext

```
dbPowerSinkNext powersinkPtr
```

Returns the address of the next power sink, 0x0 if there is none.

Parameters

<i>powersinkPtr</i>	Address of power sink.
---------------------	------------------------

Command Order

Use this command after specifying power pad locations.

Example

```
innovus 33> dbPowerSinkNext $objPtr  
0x9673388
```

The address of the next power sink 0x9673388.

dbPowerSinkPrev

dbPowerSinkPrev *powersinkPtr*

Returns the address of the previous power sink, 0x0 if there is none.

Parameters

<i>powersinkPtr</i>	Address of power sink.
---------------------	------------------------

Command Order

Use this command after specifying power pad locations.

Example

```
innovus 35> dbPowerSinkPrev $objPtr
0x9673388
```

The previous power sink is 0x9673388.

dbPrintCellSym

dbPrintCellSym *cellPtr*

Returns the symbol table information for the specified cell.

Parameters

<i>cellPtr</i>	Address of the cell.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
set cellPtr [dbInstCell $objPtr]
```

Returns:

0x4a09678

Then:

```
dbPrintCellSym $cellPtr
```

Returns:

Symbol table of cell RAMRGB_DR (nrSym=1) :

0: [] suffixLen=0 next=1

The symbol table is returned.

dbPrintFTermMaxCapTbl

```
dbPrintFTermMaxCapTbl ftermPtr
```

Returns the address of the maximum capacitance table for the specified f-term.

Parameters

<i>ftermPtr</i>	Address of the f-term.
-----------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbPrintFTermMaxCapTbl [dbGetFTermByName BUFX2 A]
```

Returns:

No max capacitance table is found.

There is no capacitance table for the f-term A for cell BUFX2.

dbPrintFTermMaxTranTbl

```
dbPrintFTermMaxTranTbl ftermPtr
```

Returns the address of the maximum transition table for the specified f-term.

Parameters

<i>ftermPtr</i>	Address of the f-term.
-----------------	------------------------

Command Order

Use this command after importing the design.

Example

```
dbPrintFTermMaxTranTbl [dbGetFTermByName BUFX2 A]
```

Returns:

No max transisiton table is found.

There is no transition table for the f-term A for cell BUFX2.

dbPrintHInstStat

```
dbPrintHInstStat hinstPtr
```

Prints the status for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of the hierarchical instance.
-----------------	---------------------------------------

Command Order

Use this command after importing the design.

Example

```
dbPrintHInstStat $objPtr
```

Returns the status:

```
*** Stats for hinst (std= 2228 a= 30209 io= 0 blk= 0) SH1: ***
sub 1 (std= 2207 a= 29820 i= 2460 m= 9 x= 323): SH1/I129
glue0 (std= 21 a= 389 i= 23 m= 9 x= 11)
** Tot (std= 2228 a= 30209 i= 2483 m= 9 x= 328) **
** Connectivity matrix **

i\i: tot x 0 1
1: 332 320 12 0
*** End stats for hinst SH1 ***
```

dbPrintInstByFlag

`dbPrintInstByFlag flagName`

Returns the number of instances for the specified flag.

Parameters

<i>flagName</i>	Available values: Critical, IPOed, PostRouteSiFix, TrialRouted, RouteDirty, Marked, Marked2, Marked3, Marked4, Hilite
-----------------	--

Command Order

Use this command after importing the design.

Example

`dbPrintInstByFlag marked`

Returns:

0

There are no instances that are marked.

dbPrintNetByFlag

`dbPrintNetByFlag flagName`

Returns the number of nets for the specified flag.

Parameters

flagName

Available values: Critical, IPOed, PostRouteSiFix, TrialRouted, RouteDirty, Marked, Marked2, Marked3, Marked4, Hilite

Command Order

Use this command after importing the design.

Example

```
innovus 19> dbPrintNetByFlag marked
```

Returns:

0

There are no nets that are marked.

dbPrintTermByFlag

```
dbPrintTermByFlag flagName
```

Returns the number of terms for the specified flag.

Parameters

flagName

Available values: Critical, IPOed, PostRouteSiFix, TrialRouted, RouteDirty, Marked, Marked2, Marked3, Marked4, Hilite

Command Order

Use this command after importing the design.

Example

```
dbPrintTermByFlag marked
```

Returns:

0

There are no terms that are marked.

dbPropDataType

```
dbPropDataType propPtr
```

Returns the data type for the specified property. Possible values are

dbcPropInt

dbcPropBits

dbcPropDbl

dbcPropLoc

dbcPropBox

dbcPropPtr

dbcPropStr

Parameters

<i>propPtr</i>	Address of the property.
----------------	--------------------------

Command Order

Use this command after importing the design.

dbPropListDataType

`dbPropListDataType propPtr`

Returns property list data type for the specified property.

Parameters

<i>propPtr</i>	Address of the property.
----------------	--------------------------

Command Order

Use this command after importing the design and creating at least one property on one object.

dbPropListName

`dbPropListName propPtr`

Returns property list name for the specified property.

Parameters

<i>propPtr</i>	Address of the property.
----------------	--------------------------

Command Order

Use this command after importing the design and creating at least one property on one object.

dbPropListParent

`dbPropListParent propPtr`

Returns the address of the property list for the specified property.

Parameters

<code>propPtr</code>	Address of the property.
----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one property on one object.

dbPropListValue

`dbPropListValue propPtr`

Returns property list value for the specified property.

Parameters

<code>propPtr</code>	Address of the property.
----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one property on one object.

dbPropName

dbPropName *propPtr*

Returns property name for the specified property.

Parameters

<i>propPtr</i>	Address of the property.
----------------	--------------------------

Command Order

Use this command after importing the design and creating at least one property on one object.

Example

```
dbForEachObjProp $route prop {  
    set propName [dbPropName $prop ]  
    set propVal [ dbPropertyValue $prop ]  
    puts $fout " PropName = $propName"  
    puts $fout " PropertyValue = $propVal\n"  
}
```

Stores to a file the property name and value.

dbPropNext

dbPropNext *propPtr*

Returns the address of the next property.

Parameters

<i>propPtr</i>	Address of the property.
----------------	--------------------------

Command Order

Use this command after importing the design and creating at least one property.

Example

```
dbPropNext $propPtr
```

Returns:

0x96e9280

The address of the next property is 0x96e9280.

dbPropParent

```
dbPropParent propPtr
```

Returns the address of the parent for the specified property.

Parameters

<i>propPtr</i>	Address of the property.
----------------	--------------------------

Command Order

Use this command after importing the design and creating at least one property.

Example

```
dbPropParent $propPtr
```

Returns:

0x4a09678

The address of the parent is 0x4a09678.

dbPropTypeDataType

```
dbPropTypeDataType propTypePtr
```

Returns the property data type for the specified property type. Possible values are

dbcPropInt

dbcPropBits

dbcPropDbl

dbcPropLoc

dbcPropBox

dbcPropPtr

dbcPropStr

Parameters

<i>propTypePtr</i>	Address of the property type.
--------------------	-------------------------------

Command Order

Use this command after importing the design and creating at least one property.

dbPropTypeName

`dbPropTypeName propTypePtr`

Returns the property type name for the specified property type.

Parameters

<code>propTypePtr</code>	Address of the property type.
--------------------------	-------------------------------

Command Order

Use this command after importing the design and creating at least one property.

Example

```
dbPropTypeName $propType
```

Returns:

`WIRE_COUNT`

The property type name is `WIRE_COUNT`.

dbPropTypeNext

`dbPropTypeNext propTypePtr`

Returns the address of the next property type.

Parameters

<i>propTypePtr</i>	Address of the property type.
--------------------	-------------------------------

Command Order

Use this command after importing the design and creating at least one property type.

Example

```
dbPropTypeNext $propTypePtr
```

Returns:

0x4a09678

The next property type is at address 0x4a09678.

dbPropTypeTypeld

```
dbPropTypeTypeld propTypePtr
```

Returns the ID for the specified property type.

Parameters

<i>propTypePtr</i>	Address of property.
--------------------	----------------------

Command Order

Use this command after importing the design and creating at least one property.

Example

```
dbPropTypeTypeld $objPtr
```

Returns:

1234

The ID is 1234.

dbPropTypeUsage

`dbPropTypeUsage propTypePtr`

Returns the usage type for the specified property type.

Parameters

<code>propTypePtr</code>	Address of the property.
--------------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one property.

dbPropUsage

`dbPropUsage propPtr`

Returns the usage for the specified property type.

Parameters

<code>propPtr</code>	Address of property.
----------------------	----------------------

Command Order

Use this command after importing the design and creating at least one property.

dbPropValue

`dbPropValue propPtr`

Returns the value of the specified property.

Parameters

<code>propPtr</code>	Address of the property.
----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one property.

Example

```
dbForEachObjProp $route prop {  
    set propName [ dbPropertyName $prop ]  
    set propVal [ dbPropValue $prop ]  
    puts $fout " PropName = $propName"  
    puts $fout " PropValue = $propVal\n"  
}
```

Writes to a file the property name and value.

dbPt

dbPt *ptPtr*

Returns the point.

Parameters

<i>ptPtr</i>	Address of object.
--------------	--------------------

Command Order

Use this command after importing the design.

Example

dbPt [dbWireX \$objPtr]

Returns:

0 3084989

The coordinate of the point is (0.0 3048.989).

dbPtnBindFPlanPtnCutToPtnList

dbPtnBindFPlanPtnCutToPtnList *ptPtr*

Binds the partition cut to the specified partition.

Parameters

<i>ptPtr</i>	Address of partition.
--------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbPtnBindFPlanPtnCutToPtnList $ptPtr]
```

dbPtnBox

```
dbPtnBox ptnPtr
```

Returns the box for the specified partition.

Parameters

<i>ptnPtr</i>	Address of the partition.
---------------	---------------------------

Command Order

Use this command after importing the design and creating at least one partition.

Example

```
dbPtnBox $objPtr
```

Returns:

```
430300 1942300 1679300 3292300
```

The lower left coordinate for the partition box is (430.3 1942.3) while the upper right is (1679.3 3292.3).

dbPtnCell

dbPtnCell *ptnPtr*

Returns the address of the cell for the specified partition.

Parameters

ptnPtr Address of the partition.

Command Order

Use this command after importing the design and specifying partitions.

Example

dbPtnCell \$objPtr

Returns:

0x64ddbf0

The cell address is 0x64ddbf0.

dbPtnCellRailWidth

dbPtnCellRailWidth *ptnPtr*

Returns the rail width for the specified partition. A result of 0.0 indicates the default.

Parameters

ptnPtr

Address of the partition.

Command Order

Use this command after importing the design and specifying at least one partition.

Example

```
dbPtnCellRailWidth $ptnPtr
```

Returns:

0.0

The rail width is 0.0.

dbPtnCoreToBottom

dbPtnCoreToBottom ptnPtr

Returns the distance from the core to the bottom for the specified partition.

Parameters

ptnPtr

Address of the partition.

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnCoreToBottom $objPtr
```

Returns:

27000

The distance from the core to the bottom of the partition is 27.0 microns.

dbPtnCoreToLeft

dbPtnCoreToLeft *ptnPtr*

Returns the distance from the core to the left for the specified partition.

Parameters

<i>ptnPtr</i>	Address of the partition.
---------------	---------------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

dbPtnCoreToLeft \$objPtr

Returns:

1000

The distance from the core to the left for the partition is 1.0 microns.

dbPtnCoreToRight

dbPtnCoreToRight *ptnPtr*

Returns the distance from core to the right for the specified partition.

Parameters

ptnPtr

Address of the partition.

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnCoreToRight $objPtr
```

Returns:

1000

The distance from the core to the right is 1.0 microns.

dbPtnCoreToTop

```
dbPtnCoreToTop ptnPtr
```

Returns the distance from the core to the top for the specified partition.

Parameters

ptnPtr

Address of the partition.

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnCoreToTop $objPtr
```

Returns:

27000

The distance from the core to the top is 27.0 microns.

dbPtnCutBox

```
dbPtnCutBox ptnCutPtr
```

Returns the box of the specified partition cut.

Parameters

<i>ptnCutPtr</i>	Address of the partition cut.
------------------	-------------------------------

Command Order

Use this command after importing the design, specifying partitions, and creating at least one partition cut.

Example

```
dbPtnCutBox $objPtr
```

Returns:

6367232 4048300 6802300 4398203

The lower left coordinate of the partition cut is (6367.232 4048.3); the upper right coordinate is (6802.3 4398.2).

dbPtnCutName

dbPtnCutName *ptnName*

Returns the name of the specified partition.

Parameters

<i>ptnName</i>	Address of the partition cut.
----------------	-------------------------------

Command Order

Use this command after importing the design, specifying partitions, and creating at least one partition cut.

Example

dbPtnCutName \$objPtr

Returns:

sheet17

The partition name is sheet17.

dbPtnCutNextInFPlan

dbPtnCutNextInFPlan *ptnCutPtr*

Returns the address of the next partition cut. If there is no other partition cut, the command returns 0x0.

Parameters

<i>ptnCutPtr</i>	Address of the partition cut.
------------------	-------------------------------

Command Order

Use this command after importing the design, specifying partitions and creation of at least one partition cut.

Example

```
dbPtnCutNextInFPlan $objPtr
```

Returns:

0x0

There is no next partition.

dbPtNext

```
dbPtNext ptPtr
```

Returns the address of the next point, 0x0 if there is none.

Parameters

<i>ptPtr</i>	Address of point.
--------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbPtNext [dbWireX $objPtr]
```

Returns:

0x1c100009

The address of the next point is 0x1c100009.

dbPtnFeedBox

```
dbPtnFeedBox ptnFeedPtr
```

Returns the box for the specified partition feed.

Parameters

<i>ptnFeedPtr</i>	Address of the partition feedthrough.
-------------------	---------------------------------------

Command Order

Use this command after importing the design, specifying partitions, and creating at least one feedthrough.

Example

```
dbPtnFeedBox $ptnFeedPtr
```

Returns:

6367232 4048300 6802300 4398203

The lower left coordinate for the partition feed is (6367.232 4048.3); the upper right coordinate is (6802.3 4398.203).

dbPtnFeedLayer

dbPtnFeedLayer *ptnFeedPtr*

Returns the feedthrough layer for the specified partition feed.

Parameters

<i>ptnFeedPtr</i>	Address of partition feedthrough.
-------------------	-----------------------------------

Command Order

Use this command after partitioning and creating at least one feedthrough.

Example

```
dbPtnFeedLayer $objPtr
```

Returns:

2

The partition feed is on `metal2`.

dbPtnFeedName

dbPtnFeedName *ptnFeedPtr*

Returns the name of the partition feed.

Parameters

ptnFeedPtr

Address of the partition feed.

Command Order

Use this command after importing the design, specifying partitions, and creating at least one partition feedthrough.

Example

```
dbPtnFeedName $ptnFeedPtr
```

Returns:

sheet27

The name of the partition feed is sheet27.

dbPtnFeedNext

```
dbPtnFeedNext ptnFeedPtr
```

Returns the address of the next partition feedthrough. If there is no other partition feedthrough, the command returns 0x0.

Parameters

ptnFeedPtr

Address of the partition feedthrough.

Command Order

Use this command after importing the design, specifying partitions, and creating at least one partition feedthrough.

Example

```
dbPtnFeedNext $ptnFeedPtr
```

Returns:

0x0

There is no next feedthrough.

dbPtnFreePinLayerOnSide

```
dbPtnFreePinLayerOnSide ptnPtr side
```

Returns the number of free pins for the specified partition on the specified side.

Parameters

<i>ptnPtr</i>	Address of partition
<i>side</i>	Side.

Command Order

Use this command after partitioning.

dbPtnHInst

```
dbPtnHInst ptnPtr
```

Returns the address of the hierarchical instance for the specified partition.

Parameters

<i>ptnPtr</i>	Address of the partition
---------------	--------------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
set objPtr [dbGetPtnByName sheet7]
```

Returns:

0x4de9750

Then:

```
dbPtnHInst $objPtr
```

Returns:

0x8a94648

The hierarchical instance address is 0x8a94648.

dbPtnHorTrackOffsetOnLayer

```
dbPtnHorTrackOffsetOnLayer ptnPtr Int
```

Returns the horizontal track offset for the specified partition.

Parameters

<i>ptnPtr</i>	Address of the partition
<i>Int</i>	Metal layer.

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnHorTrackOffsetOnLayer $objPtr 3
```

Returns:

0

The horizontal track offset is 0 for *meta3*.

dbPtnInst

```
dbPtnInst ptnPtr
```

Returns the address of the instance for the specified partition. If there is no instance, the command returns 0x0.

Parameters

<i>ptnPtr</i>	Address of the partition.
---------------	---------------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnInst $objPtr
```

Returns:

0x0

There is no instance.

dbPtnLayerBlocked

dbPtnLayerBlocked *ptnPtr*

Returns the bit code for layers that are blocked for the specified partition.

Parameters

ptnPtr

Address of the partition.

Command Order

Use this command after importing the design and specifying partitions.

Example

dbPtnLayerBlocked \$objPtr

Returns:

0x7

The partition is blocked on metal layers 1 through 3.

dbPtnLayerHaloOnLayer

dbPtnLayerHaloOnLayer *ptnPtr*

Returns the halo layer for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbForEachHeadPtn [dbgHead] p{dbPtnLayerHaloOnLayer $objPtr}
```

Returns:

0

The halo is on layer 0.

dbPtnMinPinSpace

```
dbPtnMinPinSpace ptnPtr
```

Returns the minimum space between pins for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnMinPinSpace $objPtr
```

Returns:

1

The partition pin space is 1.

dbPtnMinPinSpaceOnSide

```
dbPtnMinPinSpaceOnSide ptnPtr side
```

Returns the minimum space between pins on the specified side (function actually returns pitch -1).

Parameters

<i>ptnPtr</i>	Address of partition.
<i>side</i>	Side.

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnMinPinSpaceOnSide $ptnPtr dbcW
```

Returns:

600

The minimum space on the west side is 0.6.

dbPtnName

dbPtnName *ptnPtr*

Returns the name of the specified partition.

Parameters

<i>ptnPtr</i>	Address of the partition.
---------------	---------------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

dbPtnName \$objPtr

Returns:

sheet7

The partition name is sheet7.

dbPtnNext

dbPtnNext *ptnPtr*

Returns the address of next partition. If there is no other partition, the command returns 0x0.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnNext $objPtr
```

Returns:

0x0

There is no next partition.

dbPtnPCell

```
dbPtnPCell ptnPtr
```

Returns the address of the physical cell for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after commit partition.

Example

```
dbPtnPCell [dbGetPtnByName sheet7]
```

Returns:

0x72b2258

The address of the physical cell is 0x72b2258.

dbPtnPinBlkBox

dbPtnPinBlkBox *ptnPinBlkPtr*

Returns the box for the specified partition pin blockage.

Parameters

<i>ptnPinBlkPtr</i>	Address of the partition bin blockage.
---------------------	--

Command Order

Use this command after importing the design, specifying partitions, and creating at least one partition pin blockage.

Example

dbPtnPinBlkBox \$objPtr

Returns:

1308557 3959863 2043072 4137569

The partition pin blockage's lower left coordinate is (1308.557 3959.863); the upper right coordinate is (2043.072 4137.569).

dbPtnPinBlkLayer

dbPtnPinBlkLayer *ptnPinBlkPtr*

Returns the bit code for layers that are blocked in the specified partition pin blockage.

Parameters

<i>ptnPinBlkPtr</i>	Address of the partition pin blockage.
---------------------	--

Command Order

Use this command after importing the design, specifying partitions, and creating at least one partition pin blockage.

Example

```
dbPtnPinBlkLayer $objPtr
```

Returns:

7

Metal layers 1 through 3 are blocked.

dbPtnPinBlkName

```
dbPtnPinBlkName ptnPinBlkPtr
```

Returns the partition pin blockage name for the specified partition pin blockage.

Parameters

<i>ptnPinBlkPtr</i>	Address of the partition pin blockage.
---------------------	--

Command Order

Use this command after importing the design, specifying partitions, and creating at least one partition pin blockage.

Example

```
dbPtnPinBlkName $objPtr
```

Returns:

```
defPtnPinBlkName
```

The name of the partition pin blockage is `defPtnPinBlkName`.

dbPtnPinBlkNext

```
dbPtnPinBlkNext ptnPinBlkPtr
```

Returns the address of the next partition pin blockage. If there is no other partition pin blockage, the command returns 0x0.

Parameters

<code>ptnPinBlkPtr</code>	Address of the partition pin blockage.
---------------------------	--

Command Order

Use this command after importing the design, specifying partitions, and creating at least one partition pin blockage.

Example

```
dbPtnPinBlkNext $objPtr
```

Returns:

0x0

There is no next partition pin blockage.

dbPtnPlacementHaloBottomSideSize

```
dbPtnPlacementHaloBottomSideSize ptnPtr
```

Returns the size of the placement halo on the bottom of the specified partition. The command returns 0 if no halo exists.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnPlacementHaloBottomSideSize [dbGetPtnByName tdsp_core]
```

Returns:

0

There is no routing halo for the bottom of the partition.

dbPtnPlacementHaloLeftSideSize

```
dbPtnPlacementHaloLeftSideSize ptnPtr
```

Returns the size of the placement halo on the left side of the specified partition. The command returns 0 if no halo exists.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnPlacementHaloLeftSideSize [dbGetPtnByName tdsp_core]
```

Returns:

0

There is no halo on the left side of the partition.

dbPtnPlacementHaloRightSideSize

```
dbPtnPlacementHaloRightSideSize ptnPtr
```

Returns the size of the placement halo on the right side of the specified partition. The command returns 0 if no halo exists.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnPlacementHaloRightSideSize [dbGetPtnByName tdsp_core]
```

Returns:

0

There is no halo on the right side of the partition.

dbPtnPlacementHaloTopSideSize

```
dbPtnPlacementHaloTopSideSize ptnPtr
```

Returns the size of the placement halo on the top side of the specified partition. The command returns 0 if no halo exists.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnPlacementHaloTopSideSize [dbGetPtnByName tdsp_core]
```

Returns:

0

There is no halo on the top side of the partition.

dbPtnRoutingHaloBottomLayer

```
dbPtnRoutingHaloBottomLayer ptnPtr
```

Returns the bottom layer of the routing halo for the specified partition. The command returns 0 if no routing halo exists.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnRoutingHaloBottomLayer [dbGetPtnByName tdsp_core]
```

Returns:

0

There is no routing halo for the partition tdsp_core.

dbPtnRoutingHaloSideSize

```
dbPtnRoutingHaloSideSize ptnPtr
```

Returns the size of the routing halo for the specified partition. The command returns 0 if no routing halo exists.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnRoutingHaloSideSize [dbGetPtnByName tdsp_core]
```

Returns:

0

There is no routing halo for the partition `tdsp_core`.

dbPtnRoutingHaloTopLayer

```
dbPtnRoutingHaloTopLayer ptnPtr
```

Returns the top layer of the routing halo for the specified partition. The command returns 0 if no routing halo exists.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnRoutingHaloTopLayer [dbGetPtnByName tdsp_core]
```

Returns:

0

There is no routing halo for the partition `tdsp_core`.

dbPtnStdCellHgt

```
dbPtnStdCellHgt ptnPtr
```

Returns the standard row height for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
---------------	-----------------------

Command Order

Use this command after importing the design and specifying partitions.

Example

```
dbPtnStdCellHgt $objPtr
```

Returns:

27000

The standard cell height is 27.0 microns.

dbPtnVerTrackOffsetOnLayer

`dbPtnVerTrackOffsetOnLayer ptnPtr Int`

Returns the vertical track offset for the specified layer partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>Int</i>	Metal layer.

Command Order

Use this command after partitioning.

Example

`dbPtnVerTrackOffsetOnLayer [dbGetPtnByName sheet7] 2`

Returns:

500

The vertical track offset for partition `sheet7` for *meta2* is 0.5.

Q

dbQueryCongestAtLoc

`dbQueryCongestAtLoc locx locy`

Returns the congestion at the specified location.

Parameters

<i>locx</i>	x coordinate.
<i>locy</i>	y coordinate.

Command Order

Use this command after place.

Example

```
dbQueryCongestAtLoc 1000 1000
```

Returns:

<null>

There is no congestion at (1.0 1.0).

dbQueryInstInBox

```
dbQueryInstInBox llx lly urx ury
```

Returns whether the instance is in the specified box.

Parameters

<i>llx</i>	Lower left x coordinate of box.
<i>lly</i>	Lower left y coordinate of box.
<i>urx</i>	Upper right x coordinate of box.
<i>ury</i>	Upper right y coordinate of box.

Command Order

Use this command after partitioning.

R

dbRebindPtnCutToPtn

`dbRebindPtnCutToPtn ptnCutPtr ptnPtr`

Binds the partition cut from the old partition to the specified partition.

Parameters

<code><i>ptnCutPtr</i></code>	Address of partition.
<code><i>ptnPtr</i></code>	Address of partition.

Command Order

Use this command after importing the design and creating at least one partition cut.

dbReCalculateDefaultTechSite

dbReCalculateDefaultTechSite

`dbReCalculateDefaultTechSite`

Returns the address of the minimum tech site.

Command Order

Use this command after importing the design.

Example

```
dbReCalculateDefaultTechSite
```

Returns:

0x7f753c8

The address of the minimum tech site is 0x7f753c8.

dbReclaimAreaDownsize

```
dbReclaimAreaDownsize
```

Reclaims area by downsizing instances that are not on timing critical paths

Command Order

Use this command after timing analysis.

dbRegisterPropType

```
dbRegisterPropType propName propDescription propDataType
```

Registers the property type.

Parameters

<i>propName</i>	Userdefined string.
-----------------	---------------------

<i>propDescription</i>	User-defined string.
<i>propDataType</i>	Specifies the data type (str, int, dbl)

Command Order

Use this command after importing the design.

Example

```
dbRegisterPropType WIRE_COUNT "Assign a number to each wire" int
```

```
dbRegisterPropType VIA_COUNT "Assign a number to each via" dbl
```

Two properties are registered: `WIRE_COUNT` and `VIA_COUNT`.

dbReportMsmvInstVoltage

```
dbReportMsmvInstVoltage outFileName
```

Prints instance voltages to the specified output file.

Parameters

<i>outFileName</i>	Name of output file.
--------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
dbReportMsmvInstVoltage my.rpt
```

dbResetIsNetGnd

dbResetIsNetGnd *netPtr*

Resets the specified net to be *NOT* ground.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
dbResetIsNetGnd [dbGetNetByName BLU]
```

The net is no longer ground.

dbResetIsNetPwr

dbResetIsNetPwr *netPtr*

Resets the specified net to be *NOT* power.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
selectNet VDD  
set s [dbHeadSelList]
```

Returns:

0xd7e7220

Then:

```
set objPtr [dbSelPtr $s]
```

Returns:

0xd623cdc

Then:

```
dbResetIsNetPwr $objPtr
```

The net is no longer power.

dbResizeInst

```
dbResizeInst instPtr cellPtr
```

Resizes an instance to that of cell master.

Parameters

<i>instPtr</i>	Address of instance.
<i>cellPtr</i>	Address of cell.

Command Order

Use this command after importing the design.

Example

```
dbResizeInst $objPtr $cellPtr
```

The instance is resized.

dbRotateLocWROrigin

```
dbRotateLocWROrigin dbLocX dbLocY orient dbLocX dbLocY
```

Returns the legal location for the rotated/mirror location.

Parameters

<i>dbLocX</i>	Location of x, in database units.
<i>dbLocY</i>	Location of y, in database units.
<i>orient</i>	Orientation. Available values: dbcR0, dbcR90, dbcR180, dbcR270, dbcMX, dbcMY, dbcMX90, dbcMY90.
<i>dbLocX</i>	X coordinate, in database units, for center of rotation or mirror defined by <i>orient</i> .
<i>dbLocX</i>	Y coordinate, in database units, for center of rotation or mirror defined by <i>orient</i> .

Command Order

Use this command after importing the design.

Examples

```
dbRotateLocWROrigin 1 1 dbcR0 10 10
```

Returns:

1 1

dbRotateLocWROrigin 7 6 dbcMX 15 10

Returns:

7 14

dbRotateLocWROrigin 0 0 dbcMX 10 10

Returns:

0 20

dbRotateLocWROrigin 0 0 dbcMX90 10 10

Returns:

0 0

dbRouteBlkBox

dbRouteBlkBox *routeBlkPtr*

Returns the box that is being blocked for routing for the specified routing blockage.

Parameters

<i>routeBlkPtr</i>	Address of the routing blockage.
--------------------	----------------------------------

Command Order

Use this command after importing the design and creating at least one routing blockage.

dbRouteBlkLayer

`dbRouteBlkLayer routeBlkPtr`

Returns the bit layer that is being blocked for routing for the specified routing blockage.

Note: There can be more than one layer being blocked for routing. The bit map gives only one layer. A "for loop" must be used to get all layers.

Parameters

<code>routeBlkPtr</code>	Address of the routing blockage.
--------------------------	----------------------------------

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

`dbRouteBlkLayer $objPtr`

Returns:

4

The layer being blocked from routing is 3 (100).

The following example does not use `dbRouteBlkLayer`, but is useful for typical scripting:

```
dbForEachFPlanRouteBlk [dbCellFPlan [dbgTopCell]] blk {  
    puts [dbLayerLefName [ dbLayerShapeLayer $blk]]  
}
```

dbRouteBlkName

dbRouteBlkName *routeBlkPtr*

Returns the name of the specified routing blockage.

Parameters

<i>routeBlkPtr</i>	Address of the routing blockage.
--------------------	----------------------------------

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
dbRouteBlkName $objPtr
```

Returns:

```
defLayerBlkName
```

The name of the routing blockage is `defLayerBlkName`.

dbRouteBlkType

dbRouteBlkType *routeBlkPtr*

Returns the type of routing blockage. Possible values are:

```
dbcPtnCutLayerBlock
```

```
dbcRouteGuideLayerBlock
```

```
dbcUserLayerBlock
```

Parameters

<i>routeBlkPtr</i>	Address of the routing blockage.
--------------------	----------------------------------

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
dbRouteBlkType $objPtr
```

Returns:

```
dbcUserLayerBlock
```

The type of blockage is `user`.

dbRouteBoxBox

```
dbRouteBoxBox rbPtr
```

Returns the box for the specified route box.

Parameters

<i>rbPtr</i>	Address of route box.
--------------	-----------------------

Command Order

Use this command after importing the design and creating at least one route box.

Example

```
dbRouteBoxBox $objPtr
```

Returns:

```
665315 3171108 978738 4231091
```

The lower left coordinate of the route box is (665.315 3171.108); the upper right coordinate is (978.738 4231.091).

dbRouteBoxNext

```
dbRouteBoxNext rbPtr
```

Returns address of the next route box, 0x0 if there is none.

Parameters

<i>rbPtr</i>	Address of route box.
--------------	-----------------------

Command Order

Use this command after importing the design and creating at least one route box.

Example

```
dbRouteBoxNext $objPtr
```

Returns:

```
0x0
```

There is no next route box.

dbRouteBoxRouteGuide

dbRouteBoxRouteGuide *rbPtr*

Returns the address of the route guide for the specified route box.

Parameters

<i>rbPtr</i>	Address of the route box.
--------------	---------------------------

Command Order

Use this command after importing the design.

Example

dbRouteBoxRouteGuide \$objPtr

Returns:

0x115be198

The address of the route guide is 0x115be198.

dbRouteBoxZ

dbRouteBoxZ *rbPtr*

Returns the layer of the specified route box.

Parameters

<i>rbPtr</i>	Address of route box.
--------------	-----------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
dbRouteBoxZ $objPtr
```

```
2
```

The metal layer is 2.

dbRouteGuideBoxList

```
dbRouteGuideBoxList routeGuidePtr
```

Returns the address of the list of route guides for the specified route guide.

Parameters

<i>routeGuidePtr</i>	Address of route guide.
----------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
innovus 32> set rgPtr [dbRouteBoxRouteGuide $objPtr]
```

```
0x115be198
```

```
innovus 33> dbRouteGuideBoxList $rgPtr  
0x115c0148
```

The route guide list address is 0x115c0148.

dbRouteGuideHorLayer

`dbRouteGuideHorLayer routeGuidePtr`

Returns the metal layer used for the specified route guide in the horizontal direction.

Parameters

<code>routeGuidePtr</code>	Address of route guide.
----------------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
innovus 34> dbRouteGuideHorLayer $rgPtr  
3
```

Metal 3 is used for the route guide.

dbRouteGuideMinSpace

`dbRouteGuideMinSpace routeGuidePtr`

Returns the minimum space for the specified route guide.

Parameters

<i>routeGuidePtr</i>	SAddress of route guide.
----------------------	--------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
innovus 37> dbRouteGuideMinSpace $rgPtr  
2
```

The minimum space is 2.

dbRouteGuideName

dbRouteGuideName *routeGuidePtr*

Returns the name of the route guide for the specified route guide.

Parameters

<i>routeGuidePtr</i>	Address of route guide.
----------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
innovus 38> dbRouteGuideName $rgPtr  
group_1
```

The route guide name is `group_1`.

dbRouteGuideNetGroup

`dbRouteGuideNetGroup` *routeGuidePtr*

Returns the address of the net group associated with the specific route guide.

Parameters

<i>routeGuidePtr</i>	Address of route guide.
----------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
innovus 39> dbRouteGuideNetGroup $rgPtr  
0x11583600
```

The next route guide address is `0x11583600`.

dbRouteGuideNetList

`dbRouteGuideNetList` *routeGuidePtr*

Address of netlist for the specified route guide.

Parameters

<i>routeGuidePtr</i>	Address of route guide.
----------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
innovus 40> dbRouteGuideNetList $rgPtr  
0x115bf170
```

The netlist address is 0x115bf170.

dbRouteGuideNext

dbRouteGuideNext *routeGuidePtr*

Returns the address of the next route guide.

Parameters

<i>routeGuidePtr</i>	Address of route guide.
----------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
innovus 41> dbRouteGuideNext $rgPtr  
0x115be1c0
```

The next route guide address is 0x115be1c0.

dbRouteGuidePinGroup

dbRouteGuidePinGroup *routeGuidePtr*

Returns the address of the pin group for the specified route guide.

Parameters

<i>routeGuidePtr</i>	Address of route guide.
----------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
innovus 42> dbRouteGuidePinGroup $rgPtr  
0x11583600
```

The address of the pin group is 0x11583600.

dbRouteGuideVerLayer

dbRouteGuideVerLayer *routeGuidePtr*

Returns the metal layer for the vertical direction for the specified route guide.

Parameters

<i>routeGuidePtr</i>	Address of route guide.
----------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
innovus 43> dbRouteGuideVerLayer $rgPtr
```

```
2
```

The metal layer is 2.

dbRouteNext

```
dbRouteNext iter
```

Iterates the next route.

Parameters

<i>iter</i>	Iteration.
-------------	------------

Command Order

Use this command after importing the design and creating at least one route guide.

Example

```
proc iter_routes_delete_vias {} {  
  
    set fout [open "IterRoutesRmVias.out" w]  
  
    puts $fout "Routes Info:  
"  
    puts $fout ""  
    puts $fout ""  
  
    dbForAllCellNet [dbgTopCell] net {  
  
        set nName [dbNetName $net]  
  
        dbIterAllRoutes  
  
        set iter [dbIterRoutes $net]  
  
        while {[set route [dbRouteNext $iter]] != {0x0}} {  
  
            puts $fout "Processing net -> $nName\n"  
            if { [dbIsRouteWire $route] } {  
  
                puts $fout " wirePtr -> $route"  
  
                set wire_info [dbInfoWire $route]  
  
                puts $fout " wire -> $wire_info"  
                if { [ dbIsObjStripBox $route ] } {  
  
                    set rc [ dbDeleteStripBox $route ]  
  
                    puts $fout " Deleted the (strip box) wire, $route, rc = $rc"  
                } else {  
  
                    set rc [ dbDeleteRouteBox $route ]  
  
                    puts $fout " Deleted the (route box) wire, $route, rc = $rc"  
                }  
            } elseif { [dbIsRouteVia $route] } {  
        }
```

dbRowBox

`dbRowBox rowPtr`

Returns the box for the specified row.

Parameters

<code>rowPtr</code>	Address of row.
---------------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 43>dbRowBox $rowPtr  
2000 2000 18000 18000
```

The lower left coordinate for the row is (2.0 2.0) while the upper right coordinate is (18.0 18.0).

dbRowClusterBox

`dbRowClusterBox ioslotPtr`

Returns the box for the specified row cluster.

Parameters

ioslotPtr

Address of row cluster.

Command Order

Use this command after importing the design.

Example

```
innovus 43>dbRowClusterBox $ioslotPtr  
2000 2000 18000 18000
```

The lower left coordinate for the row cluster is (2.0 2.0), and the upper right coordinate is (18.0 18.0).

dbRowClusterName

dbRowClusterName *ioslotPtr*

Returns the name of the row cluster.

Parameters

ioslotPtr

Address of row cluster.

Command Order

Use this command after importing the design.

Example

```
innovus 43>dbRowClusterName $rgPtr  
AREAIOROW2
```

The name of the row cluster is AREAIOROW2.

dbRowClusterNext

dbRowClusterNext *ioslotPtr*

Returns the address of the specified row cluster.

Parameters

<i>ioslotPtr</i>	Address of row cluster.
------------------	-------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 43>dbRowClusterNext $rgPtr  
0xbd83b402
```

The address is 0xbd83b402.

dbRowClusterNrRow

dbRowClusterNrRow *ioslotPtr*

Returns the number of rows for the specified row cluster.

Parameters

<i>ioslotPtr</i>	Address of row cluster.
------------------	-------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 43> dbRowClusterNrRow $rgPtr  
2
```

There are two rows.

dbRowClusterRowList

```
dbRowClusterRowList ioslotPtr
```

Returns the address of the row cluster list for the specified row cluster.

Parameters

<i>ioslotPtr</i>	Address of row cluster.
------------------	-------------------------

Command Order

Use this command after importing the design and creating at least one row cluster.

Example

```
innovus 43>dbRowClusterRowList $rgPtr  
0x83b60084
```

The address of the row cluster list is 0x83b60084.

dbRowId

```
dbRowId rowPtr
```

Returns the ID for the specified row.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 43>dbRowId $rowPtr  
0
```

The ID is 0.

dbRowNext

```
dbRowNext rowPtr
```

Returns the next row address.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 43>dbRowNext $rowPtr  
0x6c29bf44
```

The next row address is 0x6c29bf44.

dbRowNextInReg

```
dbRowNextInReg rowPtr
```

Returns the next regular row address.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 43>dbRowNextInReg $rowPtr  
0xd62e9923
```

The next address is 0xd62e9923.

dbRowOrient

```
dbRowOrient rowPtr
```

Returns the orientation for the specified row.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 43>dbRowOrient $rowPtr  
R0
```

The orientation is R0.

dbRowSite

dbRowSite *rowPtr*

Returns the address of the site for the specified row.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after importing the design.

dbRowTechSite

dbRowTechSite *rowPtr*

Returns the address of the tech site for the specified row.

Parameters

<i>rowPtr</i>	Address of row.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 43>dbRowTechSite $rowPtr
```

```
0x1a49b362
```

The address of the tech site is 0x1a4b362.

dbRulerElbow

dbRulerElbow *rulerPtr*

Returns the x and y starting location for the specified ruler.

Parameters

<i>rulerPtr</i>	Address of ruler.
-----------------	-------------------

Command Order

Use this command after creating at least one ruler.

Example

```
innovus 8> set rulerPtr [dbCreateRuler 300 300]  
0x5bdedd8  
innovus 9> dbRulerElbow $rulerPtr  
300 300
```

The x starting location is 0.3, and the y starting location is 0.3.

dbRulerGrab

dbRulerGrab *rulerPtr*

Returns the grab type for the specified ruler. Possible values are:

dbcRulerNoGrab

dbcRulerGrabXY
dbcRulerGrabX
dbcRulerGrabY0
dbcRulerGrabX
dbcRulerGrabY
dbcRulerGrabElbow

Parameters

<i>rulerPtr</i>	Address of ruler.
-----------------	-------------------

Command Order

Use this command after creating at least one ruler.

Example

```
innovus 10> dbRulerGrab $rulerPtr  
dbcRulerGrabXY
```

The grab type is XY.

dbRulerGrabPosition

```
dbRulerGrabPosition rulerPtr
```

Returns the grab position of the specified ruler. Possible values are:

dbcLocOutside
dbcLocInside
dbcLocSWCorner

dbcLocSECorner
dbcLocNWCorner
dbcLocNECorner
dbcLocWSide
dbcLocESide
dbcLocNSide
dbcLocSSide
dbcLocCenter
dbcLocNoPosition

Parameters

<i>rulerPtr</i>	Address of ruler.
-----------------	-------------------

Command Order

Use this command after creating at least one ruler.

Example

```
innovus 11> dbRulerGrabPosition $rulerPtr  
dbcLocNoPosition
```

The grab position is no position.

dbRulerLoc

```
dbRulerLoc rulerPtr
```

Returns the ruler location of the specified ruler.

Parameters

<i>rulerPtr</i>	Address of ruler.
-----------------	-------------------

Command Order

Use this command after creating at least one ruler.

Example

```
innovus 12> dbRulerLoc $rulerPtr  
300 300
```

The location is (0.3 0.3).

dbRulerLoc0

```
dbRulerLoc0 rulerPtr
```

Returns the location of the first point for the specified ruler.

Parameters

<i>rulerPtr</i>	Address of ruler.
-----------------	-------------------

Command Order

Use this command after creating at least one ruler.

Example

```
innovus 13> dbRulerLoc0 $rulerPtr  
300 300
```

The location of the first point is (0.3 0.3).

Database Commands S - Z

- [S](#)
- [T](#)
- [U](#)
- [V](#)
- [W](#)

S

dbScreenBox

`dbScreenBox screenPtr`

Returns the box for the specified screen.

Parameters

<code>screenPtr</code>	Address of density screen.
------------------------	----------------------------

Command Order

Use this command after importing the design and creating at least one density screen.

Example

```
innovus 85> set s [dbHeadSelList]  
0x96e928c  
innovus 86> set objPtr [dbSelPtr $s]  
0x91fac98
```

```
innovus 87> dbScreenBox $objPtr  
2646300 3400300 3181300 3913300
```

The lower left coordinate for the screen is at (2646.3 3400.3) and the upper right coordinate is at (3181.3 3913.3).

dbScreenCapacity

`dbScreenCapacity screenPtr`

Returns the capacity for the specified screen.

Parameters

<code>screenPtr</code>	Address of screen.
------------------------	--------------------

Command Order

Use this command after importing the design and creating at least one screen.

Example

```
innovus 88> dbScreenCapacity $objPtr  
8
```

The capacity is 8.

dbScreenName

`dbScreenName screenPtr`

Returns the name of the specified screen.

Parameters

<code>screenPtr</code>	Address of screen.
------------------------	--------------------

Command Order

Use this command after importing the design and creating at least one screen.

Example

```
innovus 89> dbScreenName $objPtr  
defScreenName
```

The name is `defScreenName`.

dbScreenNext

```
dbScreenNext screenPtr
```

Returns the address of the next density screen. The command returns `0x0` if there is not another screen.

Parameters

<code>screenPtr</code>	Address of screen.
------------------------	--------------------

Command Order

Use this command after importing the design and creating at least one screen.

Example

```
innovus 90> dbScreenNext $objPtr  
0x0
```

There is not another screen.

dbSelectInstanceByFlag

```
dbSelectInstanceByFlag flagName
```

Returns the number of the selected instances with the specified flag.

Parameters

<i>flagName</i>	Flag
-----------------	------

Command Order

Use this command after importing the design.

Example

```
innovus 91> dbSelectInstanceByFlag marked 1  
0
```

There are no instances with the specified flag.

dbSelNext

```
dbSelNext selPtr
```

Returns the address of the next selected object after the specified object.

Parameters

<i>selPtr</i>	Address of selected object.
---------------	-----------------------------

Command Order

Use this command after importing the design and selecting objects.

Example

```
innovus 91> dbSelNext $objPtr  
0x8135aac
```

The address of the next selected object is 0x8135aac.

dbSelPtr

dbSelPtr *selPtr*

Returns the address of the select pointer.

Parameters

<i>selPtr</i>	Address of selected object.
---------------	-----------------------------

Command Order

Use this command after importing the design and selecting at least one object.

Example

```
innovus 92> dbSelPtr $objPtr  
0x46000032
```

The address of the select pointer is 0x46000032.

dbSetAllNetFrequency

dbSetAllNetFrequency

Sets all of the net frequencies.

Command Order

Use this command after timing analysis.

Example

```
innovus 92>dbSetAllNetFrequency  
innovus 93>
```

dbSetAreaECOWindow

`dbSetAreaECOWindow llx lly urx ury`

Returns address of the specified ECO window area. The command returns `0x0` if there is no area ECO window.

Parameters

<code>llx</code>	lower left x coordinate.
<code>lly</code>	lower left y coordinate.
<code>urx</code>	upper right x coordinate.
<code>ury</code>	upper right y coordinate.

Command Order

Use this command after placement.

Example

```
innovus 16> dbSetAreaECOWindow 1000 1000 2000 2000  
0x0
```

There is no area ECO window.

dbSetByPassDeleteNetWireTerm

`dbSetByPassDeleteNetWireTerm 1|0`

Determines whether to bypass the deletion of wire terms.

Parameters

1	Bypasses the deletion of wire terms.
0	Does not bypass the deletion of wire terms.

Command Order

Use this command after routing.

Example

```
innovus 19>dbSetByPassDeleteNetWireTerm 1
```

```
innovus 20>
```

Bypasses the deletion of wire terms.

dbSetCellCongestionView

```
dbSetCellCongestionView cellPtr congestionViewType
```

Sets the congestion view type for the specified cell. Possible values are:

dbcTrialRouter

dbcNanoRouter

dbcWarpRouter

Parameters

<i>cellPtr</i>	Address of cell.
<i>congestionViewType</i>	Congestion view type.

Command Order

Use this command after routing.

Example

```
innovus 19> dbSetCellCongestionView [dbgTopCell] dbcTrialRoute  
innovus 20>
```

The congestion view type is set to Trial Route.

dbSetCellDoNotFlatten

```
dbSetCellDoNotFlatten cellPtr {0 | 1}
```

Determines whether to mark the specified cell to not be flattened.

Parameters

<i>cellPtr</i>	Address of cell.
0	No.
1	Yes.

Command Order

Use this command before importing the design.

Example

```
innovus 102> dbSetCellDoNotFlatten $cellPtr 0  
innovus 103>
```

The cell will not be flattened.

dbSetCellLeakagePower

```
dbSetCellLeakagePower cellPtr float
```

Sets the leakage power for the specified cell.

Parameters

<i>cellPtr</i>	Address of cell.
<i>float</i>	Floating number

Command Order

Use this command after importing the design.

Example

```
innovus 22> dbSetCellLeakagePower [dbGetCellByName AND2] 2.3
```

Sets the leakage power for the cell `AND2` to `2.3`.

dbSetCellRouteAlgUsed

```
dbSetCellRouteAlgUsed cellPtr routingAlgorithm
```

Sets the routing algorithm user for the specified cell. Possible algorithms are:

`dbcTrialRouter`
`dbcNanoRouter`
`dbcWarpRouter`

Parameters

<i>cellPtr</i>	Address of cell.
<i>routingAlgorithm</i>	Routing algorithm

Command Order

Use this command after routing.

Example

```
innovus 22> dbSetCellRouteAlgUsed [dbgTopCell] dbcTrialRouter  
innovus 23>
```

The route algorithm is set to Trial Route.

dbSetCellSpareGate

```
dbSetCellSpareGate cellName [1 | 0]
```

Sets whether the specified cell is a spare gate.

Parameters

<i>cellName</i>	Name of cell.
1	Is a spare gate.
0	Is not a spare gate.

Command Order

Use this command after importing the design.

Example

```
innovus 62>dbSetCellSpareGate YFD2 1  
innovus 63>
```

Sets the cell `YFD2` as a spare gate.

dbSetCellType

```
dbSetCellType cellPtr type
```

Sets the cell type for the specified cell. The possible cell types are

```
dbcGateCell 1
```

```
dbcBlockCell 4
dbcSuperCell 7
dbcSuperBlockCell 8
```

Parameters

<i>cellPtr</i>	Address of cell.
<i>type</i>	Type.

Command Order

Use this command after importing the design.

Example

```
innovus 23> dbSetCellType [dbgTopCell] dbcSuperCell
innovus 24>
```

The top cell is a SuperCell.

dbSetCloneListOrient

```
dbSetCloneListOrient cloneListPtr orient
```

Sets the orientation of clones for the specified list of clones.

Parameters

<i>cloneListPtr</i>	Address of clone list.
<i>orient</i>	Orientation.

Command Order

Use this command after loading a design with partitioned clones.

Example

```
innovus 32> dbForEachPtnCloneList [dbGetPtnByName sheet1] cloneListPtr {  
    dbSetCloneListOrient $cloneListPtr R180  
}
```

Sets the orientation of all the clones to R180.

dbSetConstraintType

dbSetConstraintType

```
dbSetConstraintType constPtr constTypeId
```

Sets the constraint type. Possible values are:

dbcGuide
dbcFence
dbcRegion

Parameters

<i>constPtr</i>	Address of constraint.
<i>constTypeId</i>	Constraint type.

Command Order

Use this command after importing the design.

Example

```
innovus 117> set constraintPtr [dbHInstConstraint $objPtr]  
0x96e3c78  
innovus 118> dbSetConstraintType $constraintPtr dbcFence  
innovus 119>
```

The constraint is set to fence.

dbSetCurrentNet

dbSetCurrentNet {*netName* | *netPtr* | *wirePtr* | *termPtr* | *ftermPtr*}

Sets the current net to that specified by the argument and returns the address.

Parameters

<i>netName</i>	Net name.
<i>netPtr</i>	Address of net.
<i>wirePtr</i>	Address of wire.
<i>termPtr</i>	Address of terminal.
<i>ftermPtr</i>	Address of fTerm.

Command Order

Use this command after importing the design.

Example

```
innovus 119> dbSetCurrentNet BLU  
0x54dcff8
```

The address is 0x54dcff8.

dbSetCurrentZ

dbSetCurrentZ

dbSetCurrentZ *z*

Sets the current metal layer.

Parameters

<i>z</i>	Metal layer.
----------	--------------

Command Order

Use this command after starting the Innovus software.

Example

```
innovus 24> dbsetCurrentZ 3  
3
```

The current metal layer is 3.

dbSetFPinLoc

dbSetFPinLoc

```
dbSetFPinLoc fpinPtr x y z
```

Sets the location and metal layer for the specified FPin.

Parameters

<i>fpinPtr</i>	Address of fpin
<i>x</i>	x coordinate.
<i>y</i>	y coordinate.
<i>z</i>	Metal layer.

Command Order

Use this command after importing the design.

Example

```
dbForEachFTermFPin $ftermPtr FPinPtr {  
    set fpinloc [dbSetFPinLoc $FPinPtr 1000 1000 3]  
}
```

The pin is located at (1.0 1.0) on *meta3*.

dbSetFPlanBox

dbSetFPlanBox

```
dbSetFPlanBox fplanPtr llx lly urx ury
```

Sets the specified floorplan box to the coordinates specified.

Parameters

<i>fplanPtr</i>	Address of fplan.
<i>llx</i>	Lower left x coordinate.
<i>lly</i>	Lower y coordinate.
<i>urx</i>	Upper x coordinate.
<i>ury</i>	Upper y coordinate.

Command Order

Use this command after importing the design.

Example

```
innovus 122> dbSetFPlanBox [dbHeadFPlan] 1000 1000 10000 10000  
innovus 123>
```

The lower-left coordinate of the floorplan is at (1.0 1.0) and the upper-right coordinate is at (10.0 10.0).

dbSetFPlanIoBox

dbSetFPlanIoBox

```
dbSetFPlanIoBox fplanPtr llx lly urx ury
```

Sets I/O box of the specified floorplan to the specified coordinates.

Parameters

<i>fplanPtr</i>	Address of floor plan.
<i>llx</i>	Lower x coordinate.
<i>lly</i>	Lower y coordinate.
<i>urx</i>	Upper x coordinate.
<i>ury</i>	Upper y coordinate.

Command Order

Use this command after importing the design.

Example

```
innovus 33>dbSetFPlanCoreBox [dbHeadFPlan] 2000 2000 8000 800  
innovus 34>
```

The lower-left coordinate for the I/O box is at (2.0 2.0) and the upper-right is coordinate is at (8.0 8.0).

dbSetFPlanLayerHalo

```
dbSetFPlanLayerHalo fplanPtr layerHalo
```

Sets the layer for halos for the specified floorplan.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>layerHalo</i>	Layer.

Command Order

Use this command after importing the design.

Example

```
innovus 93>dbSetFPlanLayerHalo [dbHeadFPlan] 0
```

```
innovus 94>
```

The layer for halos is 0.

dbSetFPlanRegBox

dbSetFPlanRegBox

```
dbSetFPlanRegBox fplanPtr l1x l1y urx ury
```

Sets the area that encloses the rows. The regular boundary is not necessarily the same as the core area. If you have spacing between your core and your row, then the core boundary is not the same as the regular boundary.

Parameters

<i>fplanPtr</i>	Address of floorplan.
<i>l1x</i>	Lower left x coordinate.
<i>l1y</i>	Lower left y coordinate.
<i>urx</i>	Upper right x coordinate.
<i>ury</i>	Upper right y coordinate.

Command Order

Use this command after importing the design.

Example

```
innovus 27> dbSetFPlanRegBox [dbHeadFPlan] 1000 1000 3000 3000  
innovus 28>
```

The lower-left coordinate of the regular boundary is at (1.0 1.0) and the upper-right is at (3.0 3.0).

dbSetFTermDepth

dbSetFTermDepth

```
dbSetFTermDepth ftermPtr depth
```

Sets the f-term depth for the specified f-term.

Parameters

<i>ftermPtr</i>	Address of f-term.
<i>depth</i>	Depth.

Command Order

Use this command after importing the design.

Example

```
innovus 129> set term [dbGetFTermByName BLU]  
0x7cb5f30  
innovus 130> dbSetFTermDepth $term 600  
innovus 131>
```

The f-term depth is set to 0.6.

dbSetFTermLoc

dbSetFTermLoc

`dbSetFTermLoc ftermPtr x y`

Sets the f-term location to the specified coordinate.

Parameters

<code>ftermPtr</code>	Address of f-term.
<code>x</code>	x coordinate.
<code>y</code>	y coordinate.

Command Order

Use this command after importing the design.

Example

```
innovus 129> set term [dbGetFTermByName BLU]
0x7cb5f30
innovus 130> dbSetFTermLoc $term 0 2300
innovus 131>
```

The f-term location is (0.0 2.3).

dbSetFTermPlacementStatus

`dbSetFTermPlacementStatus ftermPtr status`

Sets the placement status for the specified f-term. Possible states are:

`dbcUnplaced`

dbcPlaced

dbcFixed

dbcCovered

Parameters

<i>ftermPtr</i>	Address of f-term.
<i>status</i>	State.

Command Order

Use this command after importing the design.

Example

```
innovus 129> set term [dbGetFTermByName BLU]  
0x7cb5f30  
innovus 130> dbSetFTermPlacementStatus $term 600  
dbcUnplaced
```

The f-term status is set to unplaced.

dbSetFTermType

dbSetFTermType *ftermPtr type*

Sets the type for the specified f-term. The possible types are:

dbcNormalTerm - 0

dbcClockTerm - 1

dbcLatchQTerm - 2

dbcFFQTerm - 3

dbcDTerm - 4

dbcDQTerm - 5

dbcTriStateTerm - 6

dbcPowerTerm - 7
dbcGroundTerm - 8
dbcFeedTerm - 9
dbcRSTerm - 10
dbcAsyncCtrlTerm - 11
dbcGatedClockTerm - 12
dbcAnalogTerm - 13
dbcLatchDTerm - 14
dbcLatchDQTerm - 15

Parameters

<i>ftermPtr</i>	Address of f-term.
<i>type</i>	Type.

Command Order

Use this command after importing the design.

Example

```
innovus 129> set term [dbGetFTermByName BLU]  
0x7cb5f30  
innovus 130> dbSetFTermType $term 600  
dbcNormalTerm
```

The f-term is a normal term.

dbSetFTermWidth

dbSetFTermWidth ftermPtr depth

Sets the f-term width for the specified f-term.

Parameters

<i>ftermPtr</i>	Address of f-term.
<i>width</i>	Width.

Command Order

Use this command after importing the design.

Example

```
innovus 129> set term [dbGetFTermByName BLU]
0x7cb5f30
innovus 130> dbSetFTermWidth $term 600
innovus 131>
```

The f-term width is set to 0.6.

dbSetGeomBoxBox

dbSetGeomBoxBox

`dbSetGeomBoxBox geomBoxPtr llx lly urx ury`

Sets the geometry box to the specified coordinates.

Parameters

<i>geomBoxPtr</i>	Address of geometry box.
<i>llx</i>	Lower x coordinate.
<i>lly</i>	Lower y coordinate.
<i>urx</i>	Upper x coordinate.

ury	Upper y coordinate.
-----	---------------------

Command Order

Use this command after importing the design and creating at least one geometry box.

Example

```
innovus 145> dbSetGeomBoxBox $geomBoxPtrPtr 200 200 400 400  
innovus 146>
```

The lower-left coordinate of the geometry box is at (0.2 0.2) and the upper-right coordinate is at (0.4 0.4).

dbSetGeomLineEndPt

dbSetGeomLineEndPt

```
dbSetGeomLineEndPt geomBoxPtr x1 y1
```

Sets the specified geometry box line end point to the specified coordinates.

Parameters

geomBoxPtr	Address of geometry box.
x1	Lower x coordinate.
y1	Lower y coordinate.

Command Order

Use this command after importing the design and creating at least one geometry box.

Example

```
innovus 145> dbSetGeomLineEndPt $geomBoxPtr 400 400  
innovus 146>
```

The end point is (0.4 0.4).

dbSetGeomLineStartPt

```
dbSetGeomLineStartPt geomBoxPtr x1 y1
```

Sets the geometry box line start point to the specified coordinates.

Parameters

<i>geomBoxPtr</i>	Address of geometry box.
<i>x1</i>	Lower x coordinate.
<i>y1</i>	Lower y coordinate.

Command Order

Use this command after importing the design and creating at least one geometry box.

Example

```
innovus 145> dbSetGeomLineStartPt $geomBoxPtr 200 200
```

```
innovus 146
```

The start point is (0.2 0.2).

dbSetGeomTextHeight

```
dbSetGeomTextHeight geomTextPtr height
```

Sets the geometry box text height to the specified coordinates.

Parameters

<i>geomTextPtr</i>	Address of geometry box.
<i>height</i>	Lower x coordinate.

Command Order

Use this command after importing the design and creating at least one geometry box.

Example

```
innovus 145> dbSetGeomTextHeight $geomBoxPtr 400
innovus 146>
```

The text height is 0.4.

dbSetGeomTextLoc

dbSetGeomTextLoc *geomTextPtr* *x1* *y1*

Sets the geometry box text location to the specified coordinates.

Parameters

<i>geomTextPtr</i>	Address of geometry text.
<i>x1</i>	Lower x coordinate.
<i>y1</i>	Lower y coordinate.

Command Order

Use this command after importing the design and creating at least one geometry box.

Example

```
innovus 145> dbSetGeomTextLoc $geomBoxPtr 200 200
innovus 146>
```

The text location is (0.2 0.2).

dbSetGeomTextText

`dbSetGeomTextText geomTextPtr name`

Sets the geometry box text to the specified name.

Parameters

<code>geomTextPtr</code>	Address of geometry text.
<code>name</code>	Name.

Command Order

Use this command after importing the design and creating at least one geometry box.

Example

```
innovus 145> dbSetGeomTextText $geomTextPtr INST3
innovus 146>
```

The name is `INST3`.

dbSetGroupBox

dbSetGroupBox

`dbSetGroupBox groupPtr l1x l1y urx ury`

Sets the specified group box to the specified coordinates.

Parameters

<i>groupPtr</i>	Address of group.
<i>llx</i>	Lower x coordinate.
<i>lly</i>	Lower y coordinate.
<i>urx</i>	Upper x coordinate.
<i>ury</i>	Upper y coordinate.

Command Order

Use this command after importing the design and creating at least one group.

Example

```
innovus 145> dbSetGroupBox $groupPtr 200 200 400 400  
innovus 146>
```

The lower-left coordinate of the group box is at (0.2 0.2) and the upper-right coordinate is at (0.4 0.4).

dbSetGroupConType

dbSetGroupConType

```
dbSetGroupConType groupPtr constTypeID
```

Sets the constraint type for the specified group.

The constraint type IDs you can specify are:

- 0 – dbcGuide
- 1 – dbcRegion
- 2 – dbcFence
- 3 – dbcCluster
- 4 – dbcNoConstraint

Parameters

<i>groupPtr</i>	Address of group
<i>constTypeId</i>	Constraint type ID.

Command Order

Use this command after importing the design and creating at least one group.

Example

```
innovus 9>encounter 6> createInstGroup my_group -guide 100 100 400 400
innovus 10>addInstToInstGroup my_group SH16/I402
innovus 11>addInstToInstGroup my_group SH16/I430
innovus 12>set groupPtr [dbGetGroupByName my_group]
0x9659fd0
innovus 13>dbSetGroupConType $groupPtr 1
innovus 14>
```

The constraint is set to `dbcRegion`.

dbSetHeadDBUPerIGU

`dbSetHeadDBUPerIGU headPtr dbuPerIGU`

Sets the database units per instance grid.

Parameters

<i>headPtr</i>	Address of head.
<i>dbuPerIGU</i>	Database units per grid.

Command Order

Use this command after importing the design.

Example

```
innovus 76>dbSetHeadDBUPerIGU [dbgHead] .85
innovus 77>
```

The database units per instance grid is .85.

dbSetHeadInHierMode

```
dbSetHeadInHierMode headPtr {0 | 1}
```

Sets the hierarchical mode.

Parameters

<i>headPtr</i>	Address of head.
0 1	Mode (1 equals hierarchical).

Command Order

Use this command after importing the design.

Example

```
innovus 72>dbSetHeadInHierMode [dbgHead] 1
```

The mode is set to hierarchical.

dbSetHeadStdCellHgt

```
dbSetHeadStdCellHgt headPtr stdCellHeight
```

Sets the standard cell height.

Parameters

<i>headPtr</i>	Address of head.
<i>stdCellHeight</i>	Standard cell height.

Command Order

Use this command after importing the design.

Example

```
innovus 73>dbSetHeadStdCellHgt [dbgHead] 5600
```

The standard cell height is set to 5.6.

dbSetHInstColorId

dbSetHInstColorId hinstPtr colorId

Sets the color ID for the specified hierarchical instance (module/guide).

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
<i>colorId</i>	Color ID.

Command Order

Use this command after importing the design.

Example

```
innovus 56>dbSetHInstColorId $hinstPtr 2
```

The color ID is set to 2.

dbSetHInstDensity

dbSetHInstDensity

`dbSetHInstDensity hinstPtr density`

Sets the density for the specified hierarchical instance (module/guide).

Parameters

<code>hinstPtr</code>	Address of hierarchical instance.
<code>density</code>	Density.

Command Order

Use this command after importing the design.

Example

```
innovus 44>dbSetHInstDensity $objPtr .85
```

The density is set to 0.85.

dbSetHInstHorOrder

dbSetHInstHorOrder

`dbSetHInstHorOrder hinstPtr positiveNum`

Sets the horizontal order for the specified hierarchical instance.

Parameters

<code>hinstPtr</code>	Address of inst.
-----------------------	------------------

<i>positiveNum</i>	Order.
--------------------	--------

Command Order

Use this command after importing the design.

Example

```
innovus 135> set s [dbHeadSelList]  
0x96e928c  
innovus 136> set objPtr [dbSelPtr $s]  
0x8b96bf0  
innovus 137> dbSetHInstHorOrder $objPtr 1  
innovus 138>
```

The horizontal order is set to 1.

dbSetHInstPin

dbSetHInstPin

dbSetHInstPin hinstPtr side order pin

Sets the number of pins for the specified hierarchical instance on the specified side with the specific order.

Parameters

<i>hinstPtr</i>	Address of HInst.
<i>side</i>	Side S/N/W/E.
<i>order</i>	Order.
<i>pin</i>	Number of pins.

Command Order

Use this command after importing the design.

Example

```
innovus 151> dbSetHInstPin $objPtr S 0 5  
innovus 152>
```

The pins for the south side of the module are limited to 5 and have an order of 0.

dbSetHInstRootX

dbSetHInstRootX

```
dbSetHInstRootX {0 | 1}
```

Sets whether the hierarchical instance root is the original, or the duplicate from design importation.

Parameters

0	HInst root is the original.
1	HInst root is the duplicate.

Command Order

Use this command after importing the design.

Example

```
innovus 151> dbSetHInstRootX 0  
innovus 152>
```

dbSetHInstUserDensity

dbSetHInstUserDensity

`dbSetHInstUserDensity hinstPtr density`

Sets the specified user density for the specified hierarchical instance.

Parameters

<code><i>hinstPtr</i></code>	Address of hierarchical instance.
<code><i>density</i></code>	Density.

Command Order

Use this command after importing the design.

Example

```
innovus 152> dbSetHInstUserDensity $objPtr .85
innovus 153>
```

The density is set to 0.85.

dbSetHInstVerOrder

dbSetHInstVerOrder

`dbSetHInstVerOrder hinstPtr positiveNum`

Sets the vertical order for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
<i>positiveNum</i>	Number.

Command Order

Use this command after importing the design.

Example

```
innovus 154> set s [dbHeadSelList]  
0x96e928c  
innovus 155> set objPtr [dbSelPtr $s]  
0x8b96bf0  
innovus 156> dbSetHInstVerOrder $objPtr 2  
innovus 157>
```

The vertical order is set to 2.

dbSetInstFlag

dbSetInstFlag flagName flagExpression

Sets the instance flag with the specified expression.

Parameters

<i>flagName</i>	Flag.
<i>flag_expression</i>	

Command Order

Use this command after importing the design.

dbSetInstHorOrder

dbSetInstHorOrder

`dbSetInstHorOrder instPtr positiveNum`

Sets the horizontal order for the specified instance.

Parameters

<code>instPtr</code>	Address of instance.
<code>positiveNum</code>	Number.

Command Order

Use this command after importing the design.

Example

```
innovus 158> dbSetInstHorOrder $objPtr 2
innovus 159>
```

The horizontal order is set to 2.

dbSetInstObstruct

dbSetInstObstruct

`dbSetInstObstruct instPtr 0xObstructLayerBitMask`

Sets the metal obstruction layers for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
<i>0xObstructLayerBitMask</i>	Bit mask.

Command Order

Use this command after importing the design.

Example

```
innovus 168> dbSetInstObstruct $objPtr 0x3
innovus 169>
```

The obstruction bit mask is 0x3.

dbSetInstPlacementStatus

`dbSetInstPlacementStatus instPtr status`

Sets the placement status for the specified instance. The possible states are:

`dbcUnplaced`
`dbcPlaced`
`dbcFixed`
`dbcCovered`

Parameters

<i>instPtr</i>	Address of instance.
<i>status</i>	Status

Command Order

Use this command after importing the design.

Example

```
innovus 158> dbSetInstPlacementStatus [dbGetInstByName DTMF_INST/i_10048] dbcUnplaced
```

Sets the placement status for the instance to unplaced.

dbSetInstPriority

dbSetInstPriority

```
dbSetInstPriority instPtr {0 | 1}
```

Sets whether the specified instance has priority.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 169> dbSetInstPriority $objPtr 1  
innovus 170>
```

The instance priority is set to 1.

dbSetInstRoutingHaloBottomLayer

dbSetInstRoutingHaloBottomLayer

dbSetInstRoutingHaloBottomLayer *instPtr* *layer*

Sets the bottom layer of the routing halo for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
<i>layer</i>	Number.

Command Order

Use this command after importing the design.

Example

```
innovus 158> dbSetInstRoutingHaloBottomLayer [dbGetInstByName DTMF_INST/i_148] 3
```

The bottom layer of the routing halo for instance DTMF_INST/i_148 is set to 3.

dbSetInstRoutingHaloSideSize

dbSetInstRoutingHaloSideSize *instPtr* *int*

Sets the side size for the routing halo of the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
<i>int</i>	Number.

Command Order

Use this command after importing the design.

Example

```
dbSetInstRoutingHaloSideSize [dbGetInstByName DTMF_INST/i_148] 3
```

dbSetInstRoutingHaloTopLayer

dbSetInstRoutingHaloTopLayer

```
dbSetInstRoutingHaloTopLayer instPtr layer
```

Sets the top layer of the routing halo for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
<i>layer</i>	Number.

Command Order

Use this command after importing the design.

Example

```
dbSetInstRoutingHaloTopLayer [dbGetInstByName DTMF_INST/i_148] 5
```

The top layer of the routing halo for instance DTMF_INST/i_148 is set to 5.

dbSetInstSpareGate

dbSetInstSpareGate

```
dbSetInstSpareGate instName [0 | 1]
```

Determines whether to set the specified instance as a spare gate.

Parameters

<i>instName</i>	Instance name.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 176> dbSetInstSpareGate SH19/I0/SH7/I119 1  
innovus 177>
```

The instance SH19/I0/SH7/I119 is a spare gate.

dbSetInstVerOrder

dbSetInstVerOrder

```
dbSetInstVerOrder instPtr positiveNum
```

Sets the vertical order for the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
<i>positiveNum</i>	Number.

Command Order

Use this command after importing the design.

Example

```
innovus 177> dbSetInstVerOrder $objPtr 2  
innovus 178>
```

The vertical order is set to 2.

dbSetIoBox

```
dbSetIoBox ioPtr llx lly urx ury
```

Sets the box for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>llx</i>	Lower left x coord.
<i>lly</i>	Lower left ycoord.
<i>urx</i>	Upper right x coord.
<i>ury</i>	Upper right y coord.

Command Order

Use this command after importing the design.

Example

```
innovus 178> dbSetIoBox [dbGetIoByName IOPADS_INST/Ptdigop2] 2000 2000 3000 3000  
innovus 179>
```

dbSetIoBump

dbSetIoBump *ioPtr bump*

Sets the bump for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>bump</i>	Bump

Command Order

Use this command after importing the design.

Example

```
dbSetIoBump [dbGetIoByName IOPADS_INST/Ptdigop2] VDDPAD1
```

Sets the bump for the I/O IOPADS_INST/Ptdigop2 to VDDPAD1.

dbSetIoCell

dbSetIoCell *ioPtr cell*

Sets the cell for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
--------------	------------------

cell	Cell
------	------

Command Order

Use this command after importing the design.

Example

```
dbSetIoCell dbSetIoCell [dbGetIoByName IOPADS_INST/Ptdigop2] PDO04CD
```

Sets the cell for I/O IOPADS_INST/Ptdigop2 to PDO04CD.

dbSetIoFTerm

```
dbSetIoFTerm ioPtr fterm
```

Sets the f-term for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>fterm</i>	f-term

Command Order

Use this command after importing the design.

Example

```
dbSetIoFTerm [dbGetIoByName IOPADS_INST/Ptdigop2] [dbGetFTermByName int]
```

Sets the f-term for I/O IOPADS_INST/Ptdigop to int.

dbSetIoIndent

```
dbSetIoIndent ioPtr indent
```

Sets the indent for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>indent</i>	Indent

Command Order

Use this command after importing the design.

Example

```
dbSetIoIndent [dbGetIoByName IOPADS_INST/Ptdigop2] 3.0
```

Sets the indent for I/O IOPADS_INST/Ptdigop2 to 3.0.

dbSetIoInstance

```
dbSetIoInstance ioPtr inst
```

Sets the instance for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>inst</i>	Address of instance

Command Order

Use this command after importing the design.

Example

```
dbSetIoInstance [dbGetIoByName IOPADS_INST/Ptdigop2] [dbGetInstByName DTMF_INST/io3]
```

Sets the instance for I/O `IOPADS_INST/Ptdigop2` to `DTMF_INST/ios3`.

dbSetIoLayerId

`dbSetIoLayerId ioPtr layerId`

Sets the layer ID for the specified I/O.

Parameters

<code>ioPtr</code>	Address of cell.
<code>layerId</code>	Layer ID

Command Order

Use this command after importing the design.

Example

`dbSetIoLayerId [dbGetIoByName IOPADS_INST/Ptdigop2] 3`

Sets the layer ID for I/O `IOPADS_INST/Ptdigop` to 3.

dbSetIoLoc

`dbSetIoLoc ioPtr llx lly`

Sets the location for the specified I/O.

Parameters

<code>ioPtr</code>	Address of cell.
<code>llx</code>	Lower left x coord
<code>lly</code>	Lower left y coord

Command Order

Use this command after importing the design.

Example

```
innovus 178> dbSetIoBump [dbGetIoByName IOPADS_INST/Ptdigop2] 2000 3000
innovus 179>
```

Sets the location of I/O IOPADS_INST/Ptdigop2 to 2000 3000.

dbSetIoName

```
dbSetIoName ioPtr name
```

Sets the name for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>name</i>	Name

Command Order

Use this command after importing the design.

Example

```
innovus 178> dbSetIoName [dbGetIoByName IOPADS_INST/Ptdigop2] VDDPADA1
innovus 179>
```

Sets the name of I/O IOPADS_INST/Ptdigop2 to VDDPADA1.

dbSetIoNext

```
dbSetIoNext ioPtextr layerId
```

Sets the next address for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>next</i>	Address

Command Order

Use this command after importing the design.

Example

```
innovus 178> dbSetIoLayerId [dbGetIoByName IOPADS_INST/Ptdigop2] 0x29570012
innovus 179>
```

Sets the next address of I/O IOPADS_INST/Ptdigop2 to 0x29570012.

dbSetIoOffset

```
dbSetIoOffset ioPtr offset
```

Sets the offset for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>offset</i>	Number

Command Order

Use this command after importing the design.

Example

```
innovus 178> dbSetIoOffset [dbGetIoByName IOPADS_INST/Ptdigop2] 3
innovus 179>
```

Sets the offset I/O IOPADS_INST/Ptdigop2 to 3.

dbSetIoOrder

```
dbSetIoOrder ioPtr order
```

Sets the order for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>order</i>	Number

Command Order

Use this command after importing the design.

Example

```
innovus 178> dbSetIoOrder [dbGetIoByName IOPADS_INST/Ptdigop2] 3
innovus 179>
```

Sets the order of I/O IOPADS_INST/Ptdigop2 to 3.

dbSetIoOrient

```
dbSetIoOrient ioPtr orient
```

Sets the orientation for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>orient</i>	Orientation

Command Order

Use this command after importing the design.

Example

```
innovus 178> dbSetIoOrient [dbGetIoByName IOPADS_INST/Ptdigop2] R180
```

```
innovus 179>
```

Sets the orientation of I/O IOPADS_INST/Ptdigop2 to R180.

dbSetIoRow

```
dbSetIoRow ioPtr row
```

Sets the row of the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>row</i>	Row

Command Order

Use this command after importing the design.

Example

```
innovus 178> dbSetIoRow $ioPtr 4
```

innovus 179>

The I/O row is set to 4.

dbSetIoRowMargin

`dbSetIoRowMargin row side value`

Sets the margin of the specified I/O row.

Parameters

<i>row</i>	Address of row.
<i>side</i>	Side.
<i>value</i>	Number.

Command Order

Use this command after importing the design.

dbSetIoSide

`dbSetIoSide ioPtr side`

Sets the side for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>side</i>	Side. Possible values are: W, E, N, S.

Command Order

Use this command after importing the design.

Example

```
dbSetIoSide $ioPtr w
```

The I/O is set to the west side.

dbSetIoSpacing

```
dbSetIoSpacing ioPtr side
```

Sets the spacing for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>spacing</i>	Number

Command Order

Use this command after importing the design.

Example

```
dbSetIoSide [dbGetIoByName IOPADS_INST/Ptdigop2] 3
```

The spacing for I/O IOPADS_INST/Ptdigop2 is set to 3.

dbSetIoTdfLibName

```
dbSetIoTdfLibName ioPtr tdfLibName
```

Sets the TDF library name for the specified I/O.

Parameters

<i>ioPtr</i>	Address of cell.
<i>tdfLibName</i>	Name.

Command Order

Use this command after importing the design.

Example

```
dbSetIoTdfLibName [dbGetIoByName IOPADS_INST/Ptdigop2] umc_std18
```

The TDF library name for I/O IOPADS_INST/Ptdigop2 is set to umc_std18.

dbSetIsCellDontTouch

```
dbSetIsCellDontTouch cellPtr {0 | 1}
```

Sets whether the specified cell is `dont_touch`.

Parameters

<i>cellPtr</i>	Address of cell.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsCellDontTouch $cellPtr 1
```

The cell is set to `dont_touch`.

dbSetIsCellDontUse

dbSetIsCellDontUse

`dbSetIsCellDontUse cellPtr {0 | 1}`

Sets whether the specified cell is not to be used.

Parameters

<code>cellPtr</code>	Address of cell.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

`dbSetIsCellDontUse $cellPtr 1`

The cell is not to be used.

dbSetIsCellDummy

dbSetIsCellDummy

`dbSetIsCellDummy cellPtr {0 | 1}`

Sets whether the specified cell is a dummy cell.

Parameters

cellPtr	Address of cell.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsCellDummy $cellPtr 1
```

The cell is a dummy cell.

dbSetIsCellIo

dbSetIsCellIo

```
dbSetIsCellIo cellPtr {0 | 1}
```

Sets whether the specified cell is an I/O

Parameters

cellPtr	Address of cell.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsCellIo $cellPtr 0
```

The cell is not an I/O.

dbSetIsCellSpareGate

dbSetIsCellSpareGate

```
dbSetIsCellSpareGate cellPtr {0 | 1}
```

Sets whether the specified cell is a spare gate.

Parameters

<i>cellPtr</i>	Address of cell.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsCellSpareGate $cellPtr 0
```

The cell is not a spare gate.

dbSetIsCellITGDelayUpdated

```
dbSetIsCellITGDelayUpdated cellPtr {0 | 1}
```

Sets whether the specified cell has been timing graph delay updated.

Parameters

<i>cellPtr</i>	Address of cell.
0	No.

1	Yes.
---	------

Command Order

Use this command after timing analysis

Example

```
innovus 182> dbSetIsCellTGDelayUpdated [dbGetCellByName [dbgTopCell]] 0
innovus 183>
```

The design has not been updated by timing.

dbSetIsFTermAssigned

dbSetIsFTermAssigned

```
dbSetIsFTermAssigned ftermPtr {0 | 1}
```

Sets whether the specified f-term is assigned.

Parameters

<i>ftermPtr</i>	Address of f-term.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsFTermAssigned [dbGetFTermByName BLU] 0
```

The f-term is not assigned.

dbSetIsFTermPreassigned

dbSetIsFTermPreassigned

```
dbSetIsFTermPreassigned ftermPtr {0 | 1}
```

Sets whether the specified f-term is preassigned.

Parameters

<i>ftermPtr</i>	Address of f-term.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 189> dbSetIsFTermPreassigned [dbGetFTermByName BLU] 0
```

The term is not preassigned.

dbSetIsFTermScanClk

dbSetIsFTermScanClk

```
dbSetIsFTermScanClk ftermPtr {0 | 1}
```

Sets whether the specified f-term is a scan clock.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsFTermScanClk [dbGetFTermByName BLU] 0
```

The f-term is not a scan clock.

dbSetIsFTermScanIn

dbSetIsFTermScanIn

```
dbSetIsFTermScanIn ftermPtr {0 | 1}
```

Sets whether the specified f-term is a scan in.

Parameters

<i>ftermPtr</i>	Address of f-term.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsFTermScanIn [dbGetFTermByName BLU] 0
```

The f-term is not a scan in.

dbSetIsFTermScanInv

dbSetIsFTermScanInv

```
dbSetIsFTermScanInv ftermPtr {0 | 1}
```

Sets whether the specified f-term is an inverted scan.

Parameters

<i>ftermPtr</i>	Address of FTerm.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsFTermScanInv [dbGetFTermByName BLU] 0
```

The f-term is not an inverted scan.

dbSetIsFTermScanOut

dbSetIsFTermScanOut

```
dbSetIsFTermScanOut ftermPtr {0 | 1}
```

Sets whether the specified f-term is a scan out.

Parameters

<i>ftermPtr</i>	Address of f-term.
-----------------	--------------------

0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsFTermScanOut [dbGetFTermByName BLU] 0
```

The f-term is not a scan out.

dbSetIsGroupPhyHier

dbSetIsGroupPhyHier

```
dbSetIsGroupPhyHier groupPtr {0 | 1}
```

Sets whether the specified group has physical hierarchy.

Parameters

<i>groupPtr</i>	Address of group.
0	No.
1	Yes.

Command Order

Use this command after importing the design and creating at least one group.

Example

```
dbSetIsGroupPhyHier $groupPtr 0
```

The group does not have physical hierarchy.

dbSetIsGroupUngroup

dbSetIsGroupUngroup

`dbSetIsGroupUngroup groupPtr {0 | 1}`

Sets whether the specified group should be ungrouped.

Parameters

<i>groupPtr</i>	Address of group.
0	No.
1	Yes.

Command Order

Use this command after importing the design and creating at least one group.

Example

`dbSetIsGroupUngroup $groupPtr 0`

The group is not ungrouped.

dbSetIsHeadCustomLayerChanged

`dbSetIsHeadCustomLayerChanged headPtr {0 | 1}`

Sets whether the specified custom layer has changed.

Parameters

<i>headPtr</i>	Address of head.
0	No.

1	Yes
---	-----

Command Order

Use this command after importing the design.

Example

```
dbSetIsHeadCustomLayerChanged [dbgHead] 0
```

The custom layer has not changed.

dbSetIsHeadDesignInMemory

```
dbSetIsHeadDesignInMemory hinstPtr {0 | 1}
```

Specifies whether design is in memory.

Parameters

<i>hinstPtr</i>	Address of HInst.
0	No.
1	Yes

Command Order

Use this command after importing the design.

Example

```
dbSetIsHeadDesignInMemory $hinstPtr 0
```

The design is not in memory.

dbSetIsHeadFPlanChanged

dbSetIsHeadFPlanChanged *headPtr* {0 | 1}

Specifies whether the floorplan has changed.

Parameters

<i>headPtr</i>	Address of head.
0	No.
1	Yes

Command Order

Use this command after importing the design.

Example

```
dbSetIsHeadFPlanChanged [dbgHead] 0
```

The floorplan has not changed.

dbSetIsHeadHiliteSticky

dbSetIsHeadHiliteSticky *headPtr* {0 | 1}

Sets whether the highlighted sets should be cleared with the select set.

Parameters

<i>headPtr</i>	Address of head
0	Should not be cleared.
1	Should be cleared

Command Order

Use this command after importing the design.

Example

```
dbSetIsHeadHiliteSticky $objPtr 0
```

The highlighted sets should be cleared with the select set.

dbSetIsHInstHidden

dbSetIsHInstHidden

```
dbSetIsHInstHidden hinstPtr {0 | 1}
```

Sets whether the specified hierarchical instance is hidden.

Parameters

<i>hinstPtr</i>	Address of HInst.
0	No.
1	Yes

Command Order

Use this command after importing the design.

Example

```
dbSetIsHInstHidden $objPtr 0
```

The hierarchical instance is not hidden.

dbSetIsHInstInFPlan

dbSetIsHInstInFPlan

```
dbSetIsHInstInFPlan hinstPtr {0 | 1}
```

Sets whether the specified hierarchical instance is in the floorplan.

Parameters

<i>hinstPtr</i>	Address of HInst.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsHInstInFPlan $objPtr 1
```

The hierarchical instance is in the floorplan.

dbSetIsHInstPinInited

dbSetIsHInstPinInited

```
dbSetIsHInstPinInited hinstPtr {0 | 1}
```

Sets whether the specified hierarchical instance is pin limited.

Parameters

<i>hinstPtr</i>	Address of HInst.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 200> dbSetIsHInstPinLimited $objPtr 1  
innovus 201>
```

The hierarchical instance is pin limited.

dbSetIsHInstUngroup

dbSetIsHInstUngroup

```
dbSetIsHInstUngroup hinstPtr {0 | 1}
```

Sets whether the specified module should be ungrouped.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 201> dbSetIsHInstUngroup $objPtr 1  
innovus 202>
```

The hierarchical instance is ungrouped.

dbSetIsInstDefCovered

```
dbSetIsInstDefCovered instPtr {0 | 1}
```

Sets whether the specified instance is defined in DEF.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 206> set s [dbHeadSelList]  
0x96e928c  
innovus 207> set objPtr [dbSelPtr $s]  
0x8b96bf0  
innovus 208> dbSetIsInstDefCovered $objPtr 1
```

The instance is defined in DEF.

dbSetIsInstDontTouch

```
dbSetIsInstDontTouch instPtr {0 | 1}
```

Sets whether the specified instance is `dont_touch`.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 206> set s [dbHeadSelList]  
0x96e928c  
innovus 207> set objPtr [dbSelPtr $s]  
0x8b96bf0  
innovus 208> dbSetIsInstDontTouch $objPtr 1
```

The instance is `dont_touch`.

dbSetIsInstHilite

dbSetIsInstHilite

```
dbSetIsInstHilite instPtr {0 | 1}
```

Sets whether the specified instance is highlighted.

Parameters

<i>instPtr</i>	Address of instance.
0	Is not highlighted
1	Is highlighted.

Command Order

Use this command after importing the design.

Example

```
innovus 206> set s [dbHeadSelList]  
0x96e928c  
innovus 207> set objPtr [dbSelPtr $s]  
0x8b96bf0  
innovus 208> dbSetIsInstHilite $objPtr 1
```

The instance is highlighted.

dbSetIsInstIPOed

dbSetIsInstIPOed

```
dbSetIsInstIPOed instPtr {0 | 1}
```

Sets whether the specified instance has been optimized for timing.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design and optimizing for timing.

Example

```
innovus 209> dbSetIsInstIPOed $objPtr 1
```

innovus 210>

The instance has been optimized for timing.

dbSetIsInstMarked

dbSetIsInstMarked *instPtr* {0 | 1}

Sets whether the specified instance is marked.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

innovus 210> dbSetIsInstMarked \$objPtr 1

innovus 211>

The instance is marked.

dbSetIsInstMarked2

dbSetIsInstMarked2 *instPtr* {0 | 1}

Sets whether the specified instance is marked 2.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 210> dbSetIsInstMarked2 $objPtr 1
innovus 211>
```

The instance is marked 2.

dbSetIsInstMarked3

```
dbSetIsInstMarked3 instPtr {0 | 1}
```

Sets whether the specified instance is marked 3.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 210> dbSetIsInstMarked3 $objPtr 1
innovus 211>
```

The instance is marked 3.

dbSetIsInstMarked4

dbSetIsInstMarked4 *instPtr* {0 | 1}

Sets whether the specified instance is marked 4.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 210> dbSetIsInstMarked4 $objPtr 1
innovus 211>
```

The instance is marked 4.

dbSetIsInstMoved

dbSetIsInstMoved *instPtr* {0 | 1}

Sets whether the specified instance has been moved.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 210> dbSetIsInstMoved $objPtr 1  
innovus 211>
```

The instance has been moved.

dbSetIsInstPlaced

dbSetIsInstPlaced

```
dbSetIsInstPlaced instPtr {0 | 1}
```

Sets whether the specified instance is placed.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 210> dbSetIsInstPlaced $objPtr 1  
innovus 211>
```

The instance is placed.

dbSetIsInstPreplaced

dbSetIsInstPreplaced

```
dbSetIsInstPreplaced instPtr {0 | 1}
```

Sets whether the specified instance is preplaced.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 211> dbSetIsInstPreplaced $objPtr 1  
innovus 212>
```

The instance is preplaced.

dbSetIsInstSpareGate

dbSetIsInstSpareGate

```
dbSetIsInstSpareGate instPtr {0 | 1}
```

Sets whether the specified instance is a spare gate.

Parameters

<i>instPtr</i>	Address of instance.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 213> dbSetIsInstSpareGate $objPtr 0
innovus 214>
```

The instance is a spare gate.

dbSetIsInstTAIgnored

```
dbSetIsInstTAIgnored instPtr {0 | 1}
```

Sets whether to ignore the specified instance during timing analysis.

Parameters

<i>instPtr</i>	Address of instance.
----------------	----------------------

0	Does not ignore.
1	Ignores.

Command Order

Use this command after importing the design.

Example

```
innovus 213> dbSetIsInstTAIgnored [dbGetInstByName DTMF_INST/i_7268] 0
innovus 214>
```

The instance is not ignored during timing analysis.

dbSetIsInstUnused

```
dbSetIsInstUnused instPtr {0 | 1}
```

Sets whether the specified instance is unused.

Parameters

<i>instPtr</i>	Address of instance.
0	Is unused.
1	Is not unused.

Command Order

Use this command after importing the design.

Example

```
innovus 213> dbSetIsInstUnused [dbGetInstByName DTMF_INST/i_10048] 0
innovus 214>
```

The instance is unused.

dbSetIsIoAssigned

dbSetIsIoAssigned

```
dbSetIsIoAssigned ioPtr {0 | 1}
```

Sets whether the specified I/O is assigned.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoAssigned [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O IOPADS_INST/Ptdigop2 is not assigned.

dbSetIsIoBump

dbSetIsIoBump

```
dbSetIsIoBump ioPtr {0 | 1}
```

Sets whether the specified I/O is a bump.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoBump [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O IOPADS_INST/Ptdigop2 is not a bump.

dbSetIsIoClearance

dbSetIsIoClearance

```
dbSetIsIoClearance ioPtr {0 | 1}
```

Sets whether the specified I/O is clearance.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoClearance [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O IOPADS_INST/Ptdigop2 is not clearance.

dbSetIsIoCorner

```
dbSetIsIoCorner ioPtr {0 | 1}
```

Sets whether the specified I/O is a corner pad.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoCorner [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O IOPADS_INST/Ptdigop2 is not a corner pad.

dbSetIsIoCovered

```
dbSetIsIoCovered ioPtr {0 | 1}
```

Sets whether the specified I/O is covered.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoCovered [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O IOPADS_INST/Ptdigop2 is not covered.

dbSetIsIoDefCovered

```
dbSetIsIoDefCovered ioPtr {0 | 1}
```

Sets whether the specified I/O is DEF covered.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.

1	Yes.
---	------

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoDefCovered [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O IOPADS_INST/Ptdigop2 is not DEF covered.

dbSetIsIoDummyPad

dbSetIsIoDummyPad

```
dbSetIsIoDummyPad ioPtr {0 | 1}
```

Sets whether the specified I/O is a dummy pad.

Parameters

ioPtr	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoDummyPad [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O `IOPADS_INST/Ptdigop2` is not a dummy pad.

dbSetIsIoFixed

dbSetIsIoFixed

```
dbSetIsIoFixed ioPtr {0 | 1}
```

Sets whether the specified I/O is fixed.

Parameters

<code>ioPtr</code>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoFixed [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O `IOPADS_INST/Ptdigop2` is not fixed.

dbSetIsIoGapFixed

dbSetIsIoGapFixed

```
dbSetIsIoGapFixed ioPtr {0 | 1}
```

Sets whether the specified I/O is gap fixed.

Parameters

<i>ioPtr</i>	Address of I/O
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoGapFixed [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O IOPADS_INST/Ptdigop2 is not gap fixed.

dbSetIsIoGroundPad

dbSetIsIoGroundPad

```
dbSetIsIoGroundPad ioPtr {0 | 1}
```

Sets whether the specified I/O is a ground pad.

Parameters

<i>ioPtr</i>	Address of I/O
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoGroundPad $objPtr 0
innovus 215>
```

The I/O is not a ground pad.

dbSetIsIoHilite

dbSetIsIoHilite

```
dbSetIsIoHilite ioPtr {0 | 1}
```

Sets whether the specified I/O is highlighted.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoHilite [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O is not highlighted.

dbSetIsIoMarked

```
dbSetIsIoMarked ioPtr {0 | 1}
```

Sets whether the specified I/O is marked.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoMarked [dbGetIoByName IOPADS_INST/Ptdigop2] 0
innovus 215>
```

The I/O IOPADS_INST/Ptdigop2 is not marked.

dbSetIsIoPowerPad

dbSetIsIoPowerPad

```
dbSetIsIoPowerPad ioPtr {0 | 1}
```

Sets whether the specified I/O is a power pad.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
dbSetIsIoPowerPad $objPtr 0
```

The I/O is not a power pad.

dbSetIsIoPreplaced

dbSetIsIoPreplaced

```
dbSetIsIoPreplaced ioPtr {0 | 1}
```

Sets whether the specified I/O is preplaced.

Parameters

<i>ioPtr</i>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 214> dbSetIsIoPreplaced [dbGetIoByName IOPADS_INST/Ptdigop2] 0
```

innovus 215>

The I/O `IOPADS_INST/Ptdigop2` is not preplaced.

dbSetIsIoSel

dbSetIsIoSel

`dbSetIsIoSel ioPtr {0 | 1}`

Sets whether the specified I/O is selected.

Parameters

<code>ioPtr</code>	Address of I/O.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

innovus 214> dbSetIsIoSel [dbGetIoByName IOPADS_INST/Ptdigop2] 0

innovus 215>

The I/O `IOPADS_INST/Ptdigop2` is not selected.

dbSetIsNetAnalog

`dbSetIsNetAnalog netPtr {0 | 1}`

Sets whether the specified net is analog.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 216>dbSetIsNetAnalog [ dbGetNetByName BLU] 0  
innovus 217>
```

The net is not analog.

dbSetIsNetAvoidDetour

```
dbSetIsNetAvoidDetour netPtr {0 | 1}
```

Sets whether the specified net avoids detours.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 216>dbSetIsNetAvoidDetour [ dbGetNetByName BLU] 0  
innovus 217>
```

The net does not avoid detours.

dbSetIsNetClock

```
dbSetIsNetClock netPtr {0 | 1}
```

Sets whether the net is a clock.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Examples

```
innovus 182> dbSetIsNetClock [dbGetNetByName ibias] 0  
innovus 183>
```

The net is not a clock net

dbSetIsNetCritical

dbSetIsNetCritical

```
dbSetIsNetCritical netPtr {0 | 1}
```

Sets whether the specified net is critical.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 216> dbSetIsNetCritical [ dbGetNetByName BLU] 0
innovus 217>
```

The net is not critical.

dbSetIsNetDefInClock

```
dbSetIsNetDefInClock netPtr {0 | 1}
```

Sets whether the specified net is clock from DEF import.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 217>dbSetIsNetDefInClock [ dbGetNetByName BLU] 0
innovus 218>
```

The net is not defined in DEF.

dbSetIsNetDontTouch

```
dbSetIsNetDontTouch netPtr {0 | 1}
```

Sets whether the specified net is `dont_touch`.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 217> dbSetIsNetDontTouch [ dbGetNetByName BLU] 0
innovus 218>
```

The net is not `dont_touch`.

dbSetIsNetGnd

dbSetIsNetGnd

`dbSetIsNetGnd netPtr`

Sets the specified net as a ground net.

Parameters

<code>netPtr</code>	Address of net.
---------------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 217> dbSetIsNetGnd [ dbGetNetByName BLU]
```

Sets the net `BLU` as ground.

dbSetIsNetHilite

`dbSetIsNetHilite netPtr {0 | 1}`

Sets whether the specified net is highlighted.

Parameters

<code>netPtr</code>	Address of net.
0	No.

1	Yes.
---	------

Command Order

Use this command after importing the design.

Example

```
innovus 217> dbSetIsNetHilite [ dbGetNetByName BLU] 0
innovus 218>
```

The net is not highlighted.

dbSetIsNetIgnoreInRoute

```
dbSetIsNetIgnoreInRoute netPtr {0 | 1}
```

Sets whether the specified net should be ignored during routing.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 218> dbSetIsNetIgnoreInRoute [ dbGetNetByName BLU] 0
innovus 219>
```

The net is not ignored during routing.

dbSetIsNetIPOed

dbSetIsNetIPOed

```
dbSetIsNetIPOed netPtr {0 | 1}
```

Sets whether the specified net is optimized for timing.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 218> dbSetIsNetIPOed [ dbGetNetByName BLU] 0
innovus 219>
```

The net `BLU` is not optimized for timing.

dbSetIsNetIPOIgnored

```
dbSetIsNetIPOIgnored netPtr {0 | 1}
```

Sets whether the specified net should be ignored during optimization for timing.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 218> dbSetIsNetIPOIgnored [ dbGetNetByName BLU] 0
innovus 219>
```

The net `BLU` is not ignored during optimization for timing.

dbSetIsNetListChangedForClink

```
dbSetIsNetListChangedForClink {0 | 1}
```

Sets whether the netlist is changed for clink.

Parameters

0	No.
1	Yes.

Command Order

Use this command after importing the design.

Examples

```
innovus 182> dbSetIsNetlistCangedForClink 0
innovus 183>
```

The netlist is not changed for clink.

dbSetIsNetMarked

dbSetIsNetMarked *netPtr* {0 | 1}

Sets whether the specified net is marked.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 218>dbSetIsNetMarked [ dbGetNetByName BLU] 0  
innovus 219>
```

The net is not marked.

dbSetIsNetMarked2

dbSetIsNetMarked2 *netPtr* {0 | 1}

Sets whether the specified net is marked 2.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 218>dbSetIsNetMarked2 [ dbGetNetByName BLU] 0
innovus 219>
```

The net is not marked 2.

dbSetIsNetMarked3

```
dbSetIsNetMarked3 netPtr {0 | 1}
```

Sets whether the specified net is marked 3.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 218>dbSetIsNetMarked3 [ dbGetNetByName BLU] 0
innovus 219>
```

The net is not marked 3.

dbSetIsNetMarked4

dbSetIsNetMarked4 *netPtr* {0 | 1}

Sets whether the specified net is marked 4.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 218> dbSetIsNetMarked4 [ dbGetNetByName BLU] 0
innovus 219>
```

The net is not marked 4.

dbSetIsNetPostRouteSiFix

dbSetIsNetPostRouteSiFix

dbSetIsNetPostRouteSiFix *netPtr* {0 | 1}

Sets whether the specified net has had post-route signal integrity repair performed.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after routing the design.

Example

```
innovus 220> dbSetIsNetPostRouteSiFix [ dbGetNetByName BLU] 0
innovus 221>
```

The net `BLU` is not post-route signal integrity repaired.

dbSetIsNetPwr

dbSetIsNetPwr

```
dbSetIsNetPwr netPtr
```

Sets the specified net as a power net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 221> dbSetIsNetPwr [ dbGetNetByName BLU]  
innovus 222>
```

Sets the net `BLU` as a power net.

dbSetIsNetScanNet

dbSetIsNetScanNet

```
dbSetIsNetScanNet netPtr {0 | 1}
```

Sets whether the specified net is a scan net.

Parameters

<code>netPtr</code>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 222> dbSetIsNetScanNet [ dbGetNetByName BLU] 0
```

The net is not a scan net.

dbSetIsNetSkipRouting

```
dbSetIsNetSkipRouting netPtr {0 | 1}
```

Sets whether the specified net should be skipped during routing.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 222> dbSetIsNetSkipRouting [ dbGetNetByName BLU] 0
innovus 223>
```

The net `BLU` is not skipped during routing.

dbSetIsNetTrialRouted

```
dbSetIsNetTrialRouted netPtr {0 | 1}
```

Sets whether the specified net has been routed with Trial Route.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after routing.

Example

```
innovus 222> dbSetIsNetTrialRouted [ dbGetNetByName BLU] 0
innovus 223>
```

The net `BLU` has not been routed by Trial Route.

dbSetIsRouteGuideOptimizePin

```
dbSetIsRouteGuideOptimizePin routeGuidePtr {0 | 1}
```

Sets whether the specified route guide is pin optimized.

Parameters

<code>routeGuidePtr</code>	Address of route guide.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Example

```
innovus 21> dbSetIsRouteGuideOptimizePin $routeGuidePtr
```

```
1
```

The route guide is pin optimized.

dbSetIsRulerHilite

dbSetIsRulerHilite

```
dbSetIsRulerHilite rulerPtr {0 | 1}
```

Sets whether the specified ruler is highlighted.

Parameters

<i>rulerPtr</i>	Address of ruler.
0	Is not highlighted
1	Is highlighted.

Command Order

Use this command after invoking encounter and creating at least one ruler.

Example

```
innovus 29> set rulerPtr [dbCreateRuler 300 300]  
0x6979ce0  
innovus 31> dbSetIsRulerHilite $rulerPtr 1  
innovus 32>
```

The ruler is highlighted.

dbSetIsRulerSelHTick

dbSetIsRulerSelHTick

```
dbSetIsRulerSelHTick rulerPtr {0 | 1}
```

Sets whether the specified ruler's horizontal ticks are selectable.

Parameters

<i>rulerPtr</i>	Address of ruler.
0	No.
1	Yes.

Command Order

Use this command after invoking encounter and creating at least one ruler.

Example

```
innovus 33> dbSetIsRulerSelHTick $rulerPtr 1  
innovus 34>
```

The ticks are selectable.

dbSetIsRulerSelVTick

dbSetIsRulerSelVTick

```
dbSetIsRulerSelVTick rulerPtr {0 | 1}
```

Sets whether the specified ruler's vertical ticks are selectable.

Parameters

<i>rulerPtr</i>	Address of ruler.
0	No.
1	Yes.

Command Order

Use this command after invoking encounter and creating at least one ruler.

Example

```
innovus 34> dbSetIsRulerSelVTick $rulerPtr 1  
innovus 35>
```

The ticks are selectable.

dbSetIsStripBoxRail

dbSetIsStripBoxRail

```
dbSetIsStripBoxRail stripbox {0 | 1}
```

Sets whether the specified strip box is a rail.

Parameters

<i>stripbox</i>	Address of strip box.
0	No.
1	Yes.

Command Order

Use this command after design import and creation of at least one power strap.

Example

```
innovus 226> dbSetIsStripBoxRail $objPtr 0  
innovus 227>
```

The strip box is not a rail.

dbSetIsTermIgnored

```
dbSetIsTermIgnored termPtr {0 | 1}
```

Sets whether the specified term is ignored.

Parameters

<i>termPtr</i>	Address of term.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Examples

```
innovus 226> dbSetIsTermIgnored $objPtr 1  
innovus 227>
```

The term is ignored.

dbSetIsTermIPOIgnored

```
dbSetIsTermIPOIgnored termPtr {0 | 1}
```

Sets whether the term is ignored during timing optimization.

Parameters

<i>termPtr</i>	Address of term.
0	No.

1	Yes.
---	------

Command Order

Use this command after importing the design.

Examples

```
innovus 226> dbSetIsTermIPOIgnored $objPtr 0
innovus 227>
```

The term is not ignored during timing optimization.

dbSetIsTermMarked

```
dbSetIsTermMarked termPtr {0 | 1}
```

Sets whether the term is marked.

Parameters

<i>termPtr</i>	Address of term.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Examples

```
innovus 226> dbSetIsTermMarked $objPtr 1
```

The term is marked.

dbSetIsTermMarked2

dbSetIsTermMarked2 *termPtr* {0 | 1}

Sets whether the term is marked 2.

Parameters

<i>termPtr</i>	Address of term.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Examples

```
innovus 226> dbSetIsTermMarked2 $objPtr 0
```

The term is not marked 2.

dbSetIsTermMarked3

dbSetIsTermMarked3 *termPtr* {0 | 1}

Sets whether the term is marked 3.

Parameters

<i>termPtr</i>	Address of term.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Examples

```
innovus 226> dbSetIsTermMarked3 $objPtr 0
```

The term is not marked 3.

dbSetIsTermPlaceIgnored

```
dbSetIsTermPlaceIgnored termPtr {0 | 1}
```

Sets whether the term is ignored during placement.

Parameters

<i>termPtr</i>	Address of term.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Examples

```
innovus 226> dbSetIsTermPlaceIgnored $objPtr 1
```

```
innovus 227>
```

The term is ignored during placement.

dbSetIsTermTAIgnored

```
dbSetIsTermTAIgnored termPtr {0 | 1}
```

Sets whether the term is ignored during timing analysis.

Parameters

<i>termPtr</i>	Address of term.
0	No.
1	Yes.

Command Order

Use this command after importing the design.

Examples

```
innovus 226> dbSetIsTermTAIgnored $objPtr 0
innovus 227>
```

The term is not ignored during timing analysis.

dbSetIsVNetGnd

dbSetIsVNetGnd

```
dbSetIsVNetGnd vnetPtr
```

Sets whether the specified Verilog net is ground.

Parameters

<i>vnetPtr</i>	Address of Verilog net.
----------------	-------------------------

Command Order

Use this command after design import.

Example

```
dbSetIsVNetGnd $objPtr
```

Returns:

1

The Verilog net is a ground net.

dbSetIsVNetPwr

dbSetIsVNetPwr

```
dbSetIsVNetPwr vnetPtr
```

Sets whether the specified Verilog net is power.

Parameters

<i>vnetPtr</i>	Address of Verilog net.
----------------	-------------------------

Command Order

Use this command after design import.

Example

```
innovus 23> dbSetIsVNetPwr$objPtr
```

0

The Verilog net is not a power net.

dbSetIsXNetGnd

dbSetIsXNetGnd

dbSetIsXNetGnd *Ptr*

Sets the duplicate net to ground.

Parameters

<i>Ptr</i>	Address of net.
------------	-----------------

Command Order

Use this command after design import.

Example

```
innovus 40> dbSetIsXNetGnd [dbGetNetByName BLU]  
innovus 41>
```

The net is set to ground.

dbSetIsXNetPwr

dbSetIsXNetPwr

dbSetIsXNetPwr *Ptr*

Sets the duplicate net to power.

Parameters

<i>Ptr</i>	Address of net.
------------	-----------------

Command Order

Use this command after design import.

Example

```
innovus 41> dbSetIsXNetPwr [dbGetNetByName BLU]  
innovus 42>
```

The net is not power.

dbSetLayerBlkCutLayer

dbSetLayerBlkCutLayer

`dbSetLayerBlkCutLayer layerBlkPtr int`

Sets the specified cut layer blockages.

Parameters

<code>layerBlkPtr</code>	Address of routing blockage.
<code>int</code>	Metal layer.

Command Order

Use this command after design import and creation of at least one routing blockage.

Example

```
innovus 230> dbSetLayerBlkCutLayer $objPtr 3  
innovus 231>
```

The cut layer blockage is on metal 3.

dbSetLayerBlkLayer

dbSetLayerBlkLayer

`dbSetLayerBlkLayer layerBlkPtr int`

Sets the metal layer blockages for the specified layer blockage.

Parameters

<code>layerBlkPtr</code>	Address of routing blockage.
<code>int</code>	Metal layer.

Command Order

Use this command after design import and creation of at least one routing blockage.

Example

```
innovus 230> dbSetLayerBlkLayer $objPtr 3
innovus 231>
```

The layer blockage is blocked on metal 3.

dbSetLayerBlkName

dbSetLayerBlkName

`dbSetLayerBlkName layerBlkPtr name`

Sets the name for the specified blockage.

Parameters

<i>layerBlkPtr</i>	Address of routing blockage.
<i>name</i>	Name.

Command Order

Use this command after design import and creation of at least one routing blockage.

Example

```
innovus 231> dbSetLayerBlkName $objPtr my_blockage
innovus 232>
```

The name is set to `my_blockage`.

dbSetLayerCapPerSQ

dbSetLayerCapPerSQ

```
dbSetLayerCapPerSQ layerPtr cap
```

Sets the capacitance per square micron for the specified layer.

Parameters

<i>layerPtr</i>	Address of layer.
<i>cap</i>	Number.

Command Order

Use this command after design import.

Example

```
innovus 234> dbSetLayerCapPerSQ [dbGetLayerByName M2] 0.002
innovus 235>
```

The metal 2 capacitance per square micron is 0.002.

dbSetLayerDrawName

dbSetLayerDrawName

`dbSetLayerDrawName layerPtr drawingName`

Sets the layer drawn name for the specified layer.

Parameters

<code>layerPtr</code>	Address of layer.
<code>drawingName</code>	User name.

Command Order

Use this command after design import.

Example

```
innovus 242> dbSetLayerDrawName [dbGetLayerByName M2] metal2
innovus 243>
```

The drawn name is metal2.

dbSetLayerMaxCap

dbSetLayerMaxCap

`dbSetLayerMaxCap layerPtr maxCap`

Specifies the maximum capacitance for the layer.

Parameters

<i>layerPtr</i>	Address of layer.
<i>maxCap</i>	Number.

Command Order

Use this command after design import.

Example

```
innovus 247> dbSetLayerMaxCap [dbGetLayerByName M2] 0.5
```

```
innovus 248>
```

The maximum capacitance for metal 2 is 0.5.

dbSetLayerMinCap

dbSetLayerMinCap

```
dbSetLayerMinCap layerPtr minCap
```

Sets the minimum capacitance for the layer.

Parameters

<i>layerPtr</i>	Address of layer.
<i>minCap</i>	Number.

Command Order

Use this command after design import.

Example

```
innovus 249> dbSetLayerMinCap [dbGetLayerByName M2] 0.05
innovus 250>
```

The minimum capacitance for metal 2 is 0.05.

dbSetLayerName

dbSetLayerName

```
dbSetLayerName layerPtr name
```

Sets the layer name for the specified layer.

Parameters

<i>layerPtr</i>	Address of layer.
<i>name</i>	Name.

Command Order

Use this command after design import.

Example

```
innovus 250> dbSetLayerName [dbGetLayerByName M2] M2
innovus 251>
```

The layer name is set to M2.

dbSetLayerPrefDir

dbSetLayerPrefDir

dbSetLayerPrefDir *layerPtr* H | V

Sets the preferred direction for the specified layer.

Parameters

<i>layerPtr</i>	Address of layer.
H V	Specifies direction.

Command Order

Use this command after design import.

Example

```
innovus 251> dbSetLayerPrefDir [dbGetLayerByName M2] V  
innovus 252>
```

The preferred direction for metal2 is vertical.

dbSetLayerShapeType

dbSetLayerShapeType

dbSetLayerShapeType *layerShapePtr* *type*

Sets the type for the specified layer.

Parameters

<i>layerShapePtr</i>	Address of layer shape.
<i>type</i>	Type.

Command Order

Use this command after creating at least one shape.

dbSetLayerSpace

dbSetLayerSpace

`dbSetLayerSpace layerPtr minSpacing`

Sets the minimum spacing for the layer specified.

Parameters

<i>layerPtr</i>	Address of layer.
<i>minSpacing</i>	Number.

Command Order

Use this command after design import.

Example

```
innovus 253> dbSetLayerSpace [dbGetLayerByName M2] 0.5
innovus 254>
```

The minimum spacing for metal 2 is 0.5.

dbSetLayerThickness

dbSetLayerThickness

`dbSetLayerThickness layerPtr wireThick`

Sets the thickness for the specified layer.

Parameters

<code>layerPtr</code>	Address of layer.
<code>wireThick</code>	Number.

Command Order

Use this command after design import.

Example

```
innovus 254> dbSetLayerThickness [dbGetLayerByName M2] 0.5
innovus 255>
```

The layer thickness for metal 2 is 0.5.

dbSetLayerWirePitch

dbSetLayerWirePitch

`dbSetLayerWirePitch layerPtr wirePitch`

Sets the pitch for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
<i>wirePitch</i>	Number.

Command Order

Use this command after design import.

Example

```
innovus 255> dbSetLayerWirePitch [dbGetLayerByName M2] 1.0
innovus 256>
```

The metal 2 pitch is set to 1.0.

dbSetLayerWireWidth

dbSetLayerWireWidth

`dbSetLayerWireWidth layerPtr wireWidth`

Sets the wire width for the specified layer.

Parameters

<i>layerPtr</i>	Address of the layer.
<i>WireWidth</i>	Number.

Command Order

Use this command after design import.

Example

```
innovus 256> dbSetLayerWireWidth [dbGetLayerByName M2] 0.5
innovus 257>
```

The wire width for metal 2 is 0.5.

dbSetMacroDensity

```
dbSetMacroDensity
-cell cellname
-layer layerList
[-area "x1 y1 x2 y2"]
-value density
```

Sets the metal density for the specified macro.

Parameters

-cell <i>cellName</i>	Specifies the name of the macro.
-layer <i>layerList</i>	Specifies the layers on which to create the rectangle.
-area " <i>x1 y1 x2 y2</i> "	
	Specifies the coordinates of the rectangle.
-value <i>density</i>	Specifies the density of the rectangle, as a percentage. For example, a value of 50.0 indicates that the rectangle has a density of 50 percent on the layers. <i>Type:</i> Float <i>Value:</i> 0-100

Command Order

Use this command after importing the design.

dbSetNetDblWidth

```
dbSetNetDblWidth netPtr {0 | 1}
```

Sets the specified net to be double width or not.

Parameters

<i>netPtr</i>	Address of net.
0	No.
1	Yes.

Command Order

Use this command after design import.

Example

```
innovus 259> dbSetNetDblWidth [dbGetNetByName BLU] 0
innovus 260>
```

The net BLU is not double width.

dbSetNetExtRule

dbSetNetExtRule

```
dbSetNetExtRule netPtr extRulePtr
```

Sets the specified net to use an external rule.

Parameters

<i>netPtr</i>	Address of net.
<i>extRulePtr</i>	Address of external rule.

Command Order

Use this command after design import.

Example

```
innovus 14> dbSetNetExtRule [dbGetNetByName BLU] [dbGetExtRuleByName RULE1]  
innovus 15>
```

The net BLU will apply the external rule RULE1.

dbSetNetFlag

dbSetNetFlag flagName op flag1...

Used to do flag Boolean manipulation.

Parameters

<i>flagName</i>	FlagName: Critical, IPOed, PostRouteSiFix, TrialRouted, RouteDirty, Marked, Marked2, Marked3, Marked4, Hilite
<i>op</i>	Number 0-9???
<i>flag1</i>	Number 0-9???

Command Order

Use this command after design import.

Example

```
innovus 262>dbSetNetFlag Critical 0  
innovus 263>dbSetNetFlag Critical 1  
innovus 264>dbSetNetFlag Marked2 "! Critical"  
innovus 265>dbSetNetFlag Marked3 "Clock | External| Critical"
```

Various net flags are set.

dbSetNetFrequency

`dbSetNetFrequency netPtr frequency`

Used to set the frequency of a net.

Parameters

<i>netPtr</i>	Address of net.
<i>frequency</i>	Number.
<i>flag1</i>	Number 0-9???

Command Order

Use this command after design import.

dbSetNetHorWt

dbSetNetHorWt

`dbSetNetHorWt netPtr horizontalWt`

Sets the horizontal weight for the specified net.

Parameters

<i>netPtr</i>	Address of net.
<i>horizontalWt</i>	Number 0-9???

Command Order

Use this command after design import.

Example

```
innovus 262> dbSetNetHorWt [dbGetNetByName BLU] 3
innovus 263>
```

The net BLU has a horizontal net weight of 3.

dbSetNetPCell

dbSetNetPCell

`dbSetNetPCell netPtr cellPtr`

Sets the specified net in the specified physical cell.

Parameters

<code>netPtr</code>	Address of net.
<code>cellPtr</code>	Address of cell.

Command Order

Use this command after design import.

Example

```
innovus 45> dbSetNetPCell [dbGetNetByName BLU] [dbgTopCell]
innovus 46>
```

The net BLU is set.

dbSetNetPrefExtraSpace

`dbSetNetPrefExtraSpace netPtr 0 | 1 | 2 | 3`

Sets extra space for the specified net.

Parameters

<i>netPtr</i>	Address of net.
0 1 2 3	

Command Order

Use this command after design import.

Example

```
innovus 262>dbSetNetPrefExtraSpace [dbGetNetByName BLU] 2
```

```
innovus 263>
```

dbSetNetUserWt

dbSetNetUserWt

```
dbSetNetUserWt netPtr wt
```

Sets net weight for the specified net.

Parameters

<i>netPtr</i>	Address of net.
<i>wt</i>	Number 0-9???

Command Order

Use this command after design import.

Example

```
innovus 263> dbSetNetUserWt [dbGetNetByName BLU] 3
innovus 264>
```

The net weight for the net BLU is set to 3.

dbSetNetVerWt

dbSetNetVerWt

dbSetNetVerWt netPtr verticalWt

Sets net vertical weight for the specified net.

Parameters

<i>netPtr</i>	Address of net.
<i>verticalWt</i>	Number 0-9???s

Command Order

Use this command after design import.

Example

```
innovus 264> dbSetNetVerWt [dbGetNetByName BLU] 3
innovus 265>
```

The vertical net weight for BLU is set to 3.

dbSetNetWidthX

dbSetNetWidthX

```
dbSetNetWidthX netPtr {1 | 2 | 3 | 4 | 5 | 6 | 7}
```

Sets the width in the X direction for the specified net.

Parameters

<i>netPtr</i>	Address of net.
{1 2 3 4 5 6 7}	
	Width (or multiplier).

Command Order

Use this command after design import.

Example

```
innovus 46> dbSetNetWidthX [dbGetNetByName BLU] 3
innovus 47>
```

The width in the x dimension for the net BLU is set to 3.

dbSetNetWireStatus

```
dbSetNetWireStatus netPtr status
```

Sets the status for the specified net. Possible values are:

dbcNoWire

dbcRoutedWire

dbcFixedWire

dbcCoverWire

dbcShieldWire
dbcNoShieldWire
dbcGlobalWire

Parameters

<i>netPtr</i>	Address of net.
<i>status</i>	Status.

Command Order

Use this command after importing the design.

Example

```
innovus 46>dbSetNetWireStatus [dbGetNetByName BLU] dbcGlobalWire  
innovus 47>
```

The net `BLU` is a global wire.

dbSetNumOfIoRow

`dbSetNumOfIoRow ioRowPtr num`

Sets the number for the specified I/O row.

Parameters

<i>ioRowPtr</i>	Address of I/O row
<i>num</i>	Number.

Command Order

Use this command after importing the design.

dbSetObjFPlanBox

dbSetObjFPlanBox

`dbSetObjFPlanBox objPtr x y x2 y2`

Sets the object's box to the specified value.

Parameters

<code>objPtr</code>	Address of object.
<code>x</code>	Lower left x coordinate.
<code>y</code>	Lower left y coordinate.
<code>x2</code>	Upper right x coordinate.
<code>y2</code>	Upper right y coordinate.

Command Order

Use this command after design import.

Example

```
innovus 267> set s [dbHeadSelList]  
0x96e928c  
innovus 268> set objPtr [dbSelPtr $s]  
0x7d85a30  
innovus 269> dbSetObjFPlanBox $objPtr 200 200 800 800  
innovus 270>
```

The box's lower left coordinate is at (0.2 0.2) while the upper right coordinate is set to (0.8 0.8).

dbSetObstructName

dbSetObstructName

```
dbSetObstructName obstructPtr name
```

Sets the name for the specified obstruction.

Parameters

<i>obstructPtr</i>	Address of obstruction.
<i>name</i>	User name.

Command Order

Use this command after design import and creation of at least one placement obstruction.

Example

```
innovus 271> set s [dbHeadSelList]  
0x96e928c  
innovus 272> set objPtr [dbSelPtr $s]  
0x96d3598  
innovus 273> dbSetObstructName $objPtr my_obstruct  
innovus 274>
```

The obstruction is named `my_obstruct`.

dbSetPinGuideAsFeedthrough

dbSetPinGuideAsFeedthrough

dbSetPinGuideAsFeedthrough *routeBoxPtr Int*

Sets the pin guide as a feedthrough for the specified route box.

Parameters

<i>routeBoxPtr</i>	Address of pin guide.
<i>int</i>	Metal layer.

Command Order

Use this command after design import, partitioning and creation of at least one pin guide.

Example

```
innovus 275> set s [dbHeadSelList]  
0x96e928c  
innovus 276> set objPtr [dbSelPtr $s]  
0x96d1130  
innovus 277> dbSetPinGuideAsFeedthrough $objPtr 2  
innovus 278>
```

The pin guide is set to *metal2* pins for feedthroughs.

dbSetPrerouteAsObs

dbSetPrerouteAsObs

dbSetPrerouteAsObs *0xLayerRouteBlackageArray*

Specifies the metal layers to be seen as an obstruction.

Parameters

<i>0xLayerRouteBlackageArray</i>	
	Bit code for metal layers.

Command Order

Use this command after design import.

Example

```
innovus 279> dbSetPrerouteAsObs 0x3
innovus 280>
```

The obstruction for preroutes is set to metal layers 1-3.

dbSetPropTypeUsage

```
dbSetPropTypeUsage propTypePtr usage
```

Sets the property usage for the specified property type.

Parameters

<i>propTypePtr</i>	Address of property type.
<i>usage</i>	Usage

dbSetPropValue

dbSetPropValue

dbSetPropValue *propPtr value*

Sets the property value for the specified property.

Parameters

<i>propPtr</i>	Address of property.
<i>value</i>	Value associated with the property.

Command Order

Use this command after registering a property (dbRegisterPropType).

Example

```
set wcPtr [ dbGetOrCreatePropByName $route WIRE_COUNT ]  
dbSetPropValue $wcPtr $routeCount
```

The value of the property `WIRE_COUNT` is set to the contents of the variable `routeCount`.

dbSetPtnCellRailWidth

dbSetPtnCellRailWidth

dbSetPtnCellRailWidth *ptnPtr cellRailWidth*

Sets the partition standard cell rail width for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>cellRailWidth</i>	Number.

Command Order

Use this command after design import and partitioning.

Example

```
innovus 285> dbSetPtnCellRailWidth [dbGetPtnByName sheet7] 1.0
innovus 286>
```

The partition, sheet7, has a rail width of 1.0.

dbSetPtnFeedLayer

dbSetPtnFeedLayer

dbSetPtnFeedLayer ptnFeedPtr Int

Sets the metal layers allowed to feed through the specified partition.

Parameters

<i>ptnFeedPtr</i>	Address of partition.
<i>Int</i>	Metal layer.

Command Order

Use this command after design import.

Example

```
innovus 289> set s [dbHeadSelList]
0x96e9298
```

```
innovus 290> set objPtr [dbSelPtr $s]  
0x9888420  
innovus 291> dbSetPtnFeedLayer $objPtr 2  
innovus 292>
```

Metal 2 is allowed to feed through.

dbSetPtnFeedName

dbSetPtnFeedName

`dbSetPtnFeedName ptnFeedPtr name`

Sets the partition feed name.

Parameters

<code><i>ptnFeedPtr</i></code>	Address of partition feed through
<code><i>name</i></code>	User defined name.

Command Order

Use this command after partitioning and creation of at least one partition feed through.

Example

```
innovus 293> dbSetPtnFeedName $objPtr my_feed  
innovus 294>
```

The partition's feedthrough name is `my_feed`.

dbSetPtnFreePinLayerOnSide

dbSetPtnFreePinLayerOnSide

`dbSetPtnFreePinLayerOnSide ptnPtr side Int`

Sets the layers for free pins for the specified side for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>side</i>	Side: 0=North, 1=West, 2=South, 3=East
<i>Int</i>	Layer Map (binary).

Command Order

Use this command after partitioning.

Example

```
innovus 295> dbSetPtnFreePinLayerOnSide [dbGetPtnByName sheet7] 2 1010
innovus 296>
```

The layers for free pins on the south side of partition `sheet7` are *M2* and *M4* (1010).

dbSetPtnLayerBlocked

dbSetPtnLayerBlocked

`dbSetPtnLayerBlocked ptnPtr 0xLayerBlockageArray`

Sets the layers blocked within the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>0xLayerBlockageArray</i>	
	Metal layers blocked using 0x<bits>

Command Order

Use this command after partitioning.

Example

```
innovus 296> dbSetPtnLayerBlocked [dbGetPtnByName sheet7] 0x2
innovus 297>
```

The layers blocked are metal layers 1 and 2.

dbSetPtnLayerBlockedOnLayer

dbSetPtnLayerBlockedOnLayer

dbSetPtnLayerBlockedOnLayer *ptnPtr layerId*

Sets metal layer blocked within the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>layerId</i>	Metal layer.

Command Order

Use this command after partitioning.

Example

```
innovus 298> dbSetPtnLayerBlockedOnLayer [dbGetPtnByName sheet7] 2  
innovus 299>
```

The partition sheet7 is blocked on metal layer 2.

dbSetPtnLayerHalo

dbSetPtnLayerHalo ptnPtr layerHalo

Sets the halo layer for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>layerHalo</i>	Number.

Command Order

Use this command after partitioning.

Example

```
innovus 299>dbSetPtnLayerHalo [dbGetPtnByName sheet7] 0  
innovus 300>
```

The halo layer is 0 for the partition sheet7.

dbSetPtnLayerHaloOnLayer

dbSetPtnLayerHaloOnLayer ptnPtr layerId layerHalo

Sets the halo layer for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>layerId</i>	Layer ID
<i>layerHalo</i>	Layer

Command Order

Use this command after partitioning.

Example

```
innovus 299>dbSetPtnLayerHaloOnLayer [dbGetPtnByName sheet7] 0 0
innovus 300>
```

The halo layer is 0 for sheet7.

dbSetPtnMinPinSpace

dbSetPtnMinPinSpace

dbSetPtnMinPinSpace ptnPtr minPinSpace

Sets the minimum space between pins for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>minPinSpace</i>	Number.

Command Order

Use this command after partitioning.

Example

```
innovus 299> dbSetPtnMinPinSpace [dbGetPtnByName sheet7] 2.0
innovus 300>
```

The minimum space between pins for partition sheet7 is 2.0.

dbSetPtnMinPinSpaceOnSide

dbSetPtnMinPinSpaceOnSide ptnPtr side minPinSpace

Sets the minimum space between pins for the specified partition on a particular side.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>side</i>	Side.
<i>minPinSpace</i>	Number.

Command Order

Use this command after partitioning.

Example

```
innovus 299>dbSetPtnMinPinSpaceOnSide [dbGetPtnByName sheet7] w 2.0
innovus 300>
```

The minimum space between pins on the west side of partition sheet7 is 2.0.

dbSetPtnPinBlkLayer

dbSetPtnPinBlkLayer

`dbSetPtnPinBlkLayer ptnPinBlkPtr Int`

Sets the partition pin blockage layer.

Parameters

<code><i>ptnPinBlkPtr</i></code>	Partition pin blockage address.
<code><i>Int</i></code>	Metal layer.

Command Order

Use this command after partitioning and creation of at least one partition pin blockage.

Example

```
innovus 301> set s [dbHeadSelList]  
0x96e9298  
innovus 302> set objPtr [dbSelPtr $s]  
0x887e2e0  
innovus 303> dbSetPtnPinBlkLayer $objPtr 2  
innovus 304>
```

The partition pins are blocked on metal layer 2.

dbSetPtnPinBlkName

dbSetPtnPinBlkName

`dbSetPtnPinBlkName ptnPinBlkPtr name`

Sets the partition pin blockage name.

Parameters

<i>ptnPinBlkPtr</i>	Address of partition pin blockage.
<i>name</i>	User defined name.

Command Order

Use this command after partitioning and creation of at least one partition pin blockage.

Example

```
innovus 304> dbSetPtnPinBlkName $objPtr my_blockage
innovus 305>
```

The partition pin blockage name is set to my_blockage.

dbSetPtnPinLayerFreeOnSideOnLayer

dbSetPtnPinLayerFreeOnSideOnLayer

dbSetPtnPinLayerFreeOnSideOnLayer ptnPtr side layerId

Sets the layer free for pin assignment on the specified layer and specified side.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>side</i>	Side (N/S/W/E).
<i>layerId</i>	Layer.

Command Order

Use this command after partitioning.

Example

```
innovus 306> dbSetPtnPinLayerFreeOnSideOnLayer [dbGetPtnByName sheet7] S 2
innovus 307>
```

The partition `sheet7` has metal layer 2 free on the south side.

dbSetPtnPlacementHaloBottomSideSize

dbSetPtnPlacementHaloBottomSideSize

```
dbSetPtnPlacementHaloBottomSideSize ptnPtr Int
```

Sets the bottom side size of the placement halo for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>Int</i>	Number.

Command Order

Use this command after importing the design and creating partitions.

Example

```
dbSetPtnPlacementHaloBottomSideSize [dbGetPtnByName tdsp_core] 3
```

The bottom side size of the placement halo for the partition `tdsp_core` is 3.

dbSetPtnPlacementHaloLeftSideSize

dbSetPtnPlacementHaloLeftSideSize

`dbSetPtnPlacementHaloLeftSideSize ptnPtr Int`

Sets the left side size of the placement halo for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>Int</i>	Number.

Command Order

Use this command after importing the design and creating partitions.

Example

```
dbSetPtnPlacementHaloLeftSideSize [dbGetPtnByName tdsp_core] 3
```

The left side size of the placement halo for the partition `tdsp_core` is 3.

dbSetPtnPlacementHaloRightSideSize

dbSetPtnPlacementHaloRightSideSize

`dbSetPtnPlacementHaloRightSideSize ptnPtr Int`

Sets the right side size of the placement halo for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>Int</i>	Number.

Command Order

Use this command after importing the design and creating partitions.

Example

```
dbSetPtnPlacementHaloRightSideSize [dbGetPtnByName tdsp_core] 3
```

The left side size of the placement halo for the partition `tdsp_core` is 3.

dbSetPtnPlacementHaloTopSideSize

dbSetPtnPlacementHaloTopSideSize

```
dbSetPtnPlacementHaloTopSideSize ptnPtr Int
```

Sets the top side size of the placement halo for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>Int</i>	Number.

Command Order

Use this command after importing the design and creating partitions.

Example

```
dbSetPtnPlacementHaloTopSideSize [dbGetPtnByName tdsp_core] 3
```

The left side size of the placement halo for the partition `tdsp_core` is 3.

dbSetPtnRoutingHaloBottomLayer

```
dbSetPtnRoutingHaloBottomLayer ptnPtr layer
```

Sets the bottom layer of the routing halo for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>layer</i>	Number.

Command Order

Use this command after importing the design and creating partitions.

Example

```
dbSetPtnRoutingHaloBottomLayer [dbGetPtnByName tdsp_core] 3
```

The bottom layer of the routing halo for the partition `tdsp_core` is 3.

dbSetPtnRoutingHaloSideSize

```
dbSetPtnRoutingHaloSideSize ptnPtr Int
```

Sets the routing halo side size for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>Int</i>	Number.

Command Order

Use this command after importing the design and creating partitions.

Example

```
dbSetPtnRoutingHaloSideSize [dbGetPtnByName tdsp_core] 3
```

The routing halo side size is 3 for the partition `tdsp_core`.

dbSetPtnRoutingHaloTopLayer

`dbSetPtnRoutingHaloTopLayer ptnPtr layer`

Sets the top layer of the routing halo for the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>layer</i>	Number.

Command Order

Use this command after importing the design and creating partitions.

Example

`dbSetPtnRoutingHaloTopLayer [dbGetPtnByName tdsp_core] 3`

The top layer of the routing halo for the partition *tdsp_core* is 3.

dbSetPtnStdCellHgt

dbSetPtnStdCellHgt

`dbSetPtnStdCellHgt ptnPtr stdCellHeight`

Sets the row height inside the specified partition.

Parameters

<i>ptnPtr</i>	Address of partition.
<i>stdCellHeight</i>	Number.

Command Order

Use this command after partitioning.

Example

```
dbSetPtnStdCellHgt [dbGetPtnByName sheet7] 27.0
```

The standard cell height for partition sheet7 is set to 27.0 microns.

dbSetRouteBlkLayer

dbSetRouteBlkLayer

```
dbSetRouteBlkLayer routeBlkPtr int
```

Sets the metal route blockage for the specified layer blockage.

Parameters

routeBlkPtr	Address of routing blockage.
int	Metal layer.

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
dbSetRouteBlkLayer $objPtr 3
```

The route blockage is blocked on metal 3.

dbSetRouteBlkName

dbSetRouteBlkName

```
dbSetRouteBlkName routeBlkPtr name
```

Sets the name for the specified blockage.

Parameters

<i>routeBlkPtr</i>	Address of routing blockage.
<i>name</i>	Name.

Command Order

Use this command after importing the design and creating at least one routing blockage.

Example

```
innovus 231> dbSetRouteBlkName $objPtr my_blockage
innovus 232>
```

The name is set to `my_blockage`.

dbSetRulerGrab

dbSetRulerGrab

```
dbSetRulerGrab rulerPtr ruler_grab_type
```

Sets the ruler grab types for the specified ruler. For possible values, see `dbRulerGrab`.

Parameters

<i>rulerPtr</i>	Address of ruler.
<i>ruler_grab_type</i>	Grab type.

dbSetScreenCapacity

dbSetScreenCapacity

`dbSetScreenCapacity screenPtr capacity`

Sets the screen capacity for the specified screen.

Parameters

<code>screenPtr</code>	Address of screen.
<code>capacity</code>	Number.

Command Order

Use this command after design import and creating at least one density screen.

Example

```
innovus 309> set s [dbHeadSelList]  
0x96e9298  
innovus 310> set objPtr [dbSelPtr $s]  
0x91fac98  
innovus 311> dbSetScreenCapacity $objPtr .80  
innovus 312>
```

The screen capacity is set to 0.80.

dbSetScreenName

dbSetScreenName

`dbSetScreenName screenPtr name`

Sets the name of the screen for the specified screen.

Parameters

<i>screenPtr</i>	Address of screen.
<i>name</i>	User defined.

Command Order

Use this command after design import and creation of at least one screen.

Example

```
innovus 312> dbSetScreenName $objPtr my_screen  
innovus 313>
```

The screen's name is set to `my_screen`.

dbSetStdPlaceHInstRoot

dbSetStdPlaceHInstRoot

```
dbSetStdPlaceHInstRoot hinstPtr
```

Sets the standard cell placement for the specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 25> dbSetStdPlaceHInstRoot [dbGetHInstByName SH17]  
innovus 26>
```

The standard cell placement is set for module SH17.

dbSetStdRowOrient

dbSetStdRowOrient rowPtr orient

Sets the orientation for the specified standard row.

Parameters

<i>rowPtr</i>	Address of row.
<i>orient</i>	Orientation

Command Order

Use this command after importing the design.

Example

```
encounter 25> dbSetStdRowOrient $rowPtr R0  
encounter 26>
```

The row orientation is set to R0.

dbSetStripBoxShape

dbSetStripBoxShape stripBoxPtr shape

Sets the shape for the specified strip box.

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
<i>shape</i>	Shape: RING, STRIPE,FOLLOWPIN,IOWIRE,COREWIRE,BLOCKWIRE,FILLWIRE,PADRING,BLOCKRING

Command Order

Use this command after design import and creation of at least one strip box.

Example

```
innovus 4>dbSetStripBoxShape $objPtr STRIPE
```

The strip box is set to STRIPE.

dbSetStripBoxShieldNet

```
dbSetStripBoxShieldNet stripBoxPtr netPtr
```

Sets the specified strip box to be shielded by the specified net.

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
<i>netPtr</i>	Address of net

Command Order

Use this command after design import and creation of at least one strip box.

Example

```
innovus 4>dbSetStripBoxShieldNet $stripBoxPtr $netPtr
```

The strip box defined by the variable `stripBoxPtr` is shielded by the net specified by the variable `netPtr`.

dbSetStripBoxState

`dbSetStripBoxState stripBoxPtr z`

Sets the specified strip box state. Possible states are:

ROUTED

FIXED

COVER

SHIELD

Parameters

<code>stripBoxPtr</code>	Address of stripbox.
<code>z</code>	State.

Command Order

Use this command after importing the design and creating at least one power or ground strap.

Example

```
innovus 4>dbSetStripBoxState $objPtr FIXED  
innovus 5>
```

The strip box state is set to `FIXED`.

dbSetStripBoxZ

dbSetStripBoxZ

dbSetStripBoxZ *stripBoxPtr z*

Sets the metal layer for the specified strip box.

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
<i>z</i>	Metal layer.

Command Order

Use this command after design import and creation of at least one power/ground strap.

Example

```
innovus 4> dbSetStripBoxZ $objPtr 3
innovus 5>
```

The strip box layer is set to metal 3.

dbSetTechSite

dbSetTechSite

dbSetTechSite *site_list*

Sets the tech site.

Parameters

site_list	Name
-----------	------

Command Order

Use this command after design import.

Example

```
innovus 54> dbSetTechSite core  
innovus 55>
```

The tech site is set to core.

dbSetUserURrid

dbSetUserURrid X_Grid Y_Grid X_Offset Y_Offset

Sets the user grid.

Parameters

X_Grid	Number
Y_Grid	Number
X_Offset	Number
Y_Offset	Number

Command Order

Use this command after importing the design.

Examples

```
innovus 54> dbSetUserURGrid 1.0 2.0 0.1 0.1  
innovus 55>
```

dbSetViaCellContactId

dbSetViaCellContactId

dbSetViaCellContactId *viaCellPtr id*

Sets the contact ID for the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
<i>id</i>	Number.

Command Order

Use this command after design import.

Example

```
dbSetViaCellContactId [dbGetViaCellByName VIA3] 3
```

The contact ID is set to 3.

dbSetWireTerm

dbSetWireTerm *wlterm*

Sets the wire term for the specified wire.

Parameters

<i>wl</i>	Address of wire.
<i>term</i>	Address of term.

Command Order

Use this command after routing or creating at least one wire.

Example

```
innovus 61> dbSetWireViaDnId $wl $term  
innovus 62>
```

dbSetWireViaDnId

dbSetWireViaDnId

```
dbSetWireViaDnId netPtr {0 | 1 | 2 | 3}
```

Sets the wire via down ID for the specified net.

Parameters

<i>netPtr</i>	Address of wire.
{0 1 2 3}	Specifies via down ID.

Command Order

Use this command after design import.

Example

```
innovus 61> dbSetWireViaDnId $objPtr 2  
innovus 62>
```

The wire via down ID is set to 2.

dbSetWireViaDnStatus

dbSetWireViaDnStatus *netPtr* *status*

Sets the wire via down status for the specified net.

Parameters

<i>netPtr</i>	Address of wire.
<i>status</i>	

Command Order

Use this command after performing detail routing.

dbSetWireWidthXId

dbSetWireWidthXId

dbSetWireWidthXId *netPtr* {0 | 1 | 2 | 3}

Sets the wire width x dimension ID for the specified net.

Parameters

<i>netPtr</i>	Address of wire.
{0 1 2 3}	Width (or multiplier).

Command Order

Use this command after design import.

Example

```
innovus 62> dbSetWireWidthXId $objPtr 2
innovus 63>
```

The wire width's multiplier is 2.

dbSetWireWireStatus

```
dbSetWireWireStatus wirePtr status
```

Sets wire status for the specified wire. Possible values for status are:

```
dbcNoWire
dbcRoutedWire
dbcFixedWire
dbcCoverWire
dbcShieldWire
dbcNoShieldWire
dbcGlobalWire
```

Parameters

<i>wirePtr</i>	Address of wire.
<i>status</i>	Status.

Command Order

Use this command after design import.

Example

```
innovus 75>dbSetWireWireStatus $wirePtr dbcGlobalWire
```

The wire status is set to global wire.

dbSetWireXXMode

dbSetWireXXMode

Sets the wire mode to handle diagonals.

Command Order

Use this command after design import.

Example

```
innovus 54> dbSetWireXXMode
Wire mode switched to handle diagonal
innovus 55>
```

dbShapePoly

dbShapePoly *shapePtr*

Returns the address of the polygon for the specified shape.

Parameters

<i>shapePtr</i>	Addresses of shape.
-----------------	---------------------

Command Order

Use this command after design import.

Example

```
proc printOneShape {s} {
```

```
if {[dbIsShapeTypeRect $s]} {  
    puts "shape type is rect [dbShapeRect $s]"  
} elseif {[dbIsShapeTypePath $s]} {  
    puts "shape type is path"  
} elseif {[dbIsShapeTypePoly $s]} {  
    set poly [dbShapePoly $s]  
    puts "shape type is polygon ([llength $poly] vertices)"  
    foreach v $poly {  
        puts " $v"  
    }  
} elseif {[dbIsShapeTypeUndefined $s]} {  
    puts "shape type is undefined"  
} else {  
    puts "*** error *** shape type"  
}  
}
```

dbShapeRect

dbShapeRect *shapePtr*

Returns the address of the rectangle for the specified shape.

Parameters

<i>shapePtr</i>	Addresses of shape.
-----------------	---------------------

Command Order

Use this command after design import.

Example

```
proc printOneShape {s} {
    if {[dbIsShapeTypeRect $s]} {
        puts "shape type is rect [dbShapeRect $s]"
    } elseif {[dbIsShapeTypePath $s]} {
        puts "shape type is path"
    } elseif {[dbIsShapeTypePoly $s]} {
        set poly [dbShapePoly $s]
        puts "shape type is polygon ([llength $poly] vertices)"
        foreach v $poly {
            puts "$v"
        }
    } elseif {[dbIsShapeTypeUndefined $s]} {
        puts "shape type is undefined"
    } else {
        puts "*** error *** shape type"
    }
}
```

dbSiteName

dbSiteName *sitePtr*

Returns the name of the specified site.

Parameters

<i>sitePtr</i>	Addresses of site.
----------------	--------------------

Command Order

Use this command after design import.

Example

```
dbSiteName $sitePtr
```

Returns:

```
defSiteName
```

The name of the site is `defSiteName`.

dbSiteSizeX

```
dbSiteSizeX sitePtr
```

Returns the x dimension of the specified site.

Parameters

<code>sitePtr</code>	Addresses of site.
----------------------	--------------------

Command Order

Use this command after design import.

Example

```
dbSiteSizeX $sitePtr
```

Returns:

```
240
```

The x dimension is 0.24.

dbSiteSizeY

`dbSiteSizeY sitePtr`

Returns the y dimension of the specified site.

Parameters

<code>sitePtr</code>	Addresses of site.
----------------------	--------------------

Command Order

Use this command after design import.

Example

`dbSiteSizeY $sitePtr`

Returns:

560

The y dimension is 0.56.

dbSnapCoordToTrack

`dbSnapCoordToTrack coord layerPtn {0|1}`

Snaps the coordinate to track.

Parameters

<code>coord</code>	Coordinate
<code>layerPtn</code>	Partition Layer
{0 1}	

Command Order

Use this command after partitioning.

dbSNetBoxList

dbSNetBoxList *snetPtr*

Returns address of box list for special nets, 0x0 if there is none.

Parameters

<i>snetPtr</i>	Address of special nets.
----------------	--------------------------

Command Order

Use this command after design import.

Example

```
innovus 74> set s [dbGetSNetByName vdd]
```

```
0x9671c80
```

```
innovus 75> dbSNetBoxList $s
```

```
0x0
```

There is not a box list for `vdd`.

dbSNetCell

dbSNetCell *snetPtr*

Returns address of cell specified by the special nets, 0x0 if there is none.

Parameters

<i>snetPtr</i>	Address of special nets.
----------------	--------------------------

Command Order

Use this command after design import.

Example

```
innovus 74> set s [dbGetSNetByName vdd]  
0x9671c80  
innovus 76> dbSNetCell $s  
0x0
```

There is not a cell associated with vdd.

dbSNetName

`dbSNetName snetPtr`

Returns the name of the special net.

Parameters

<code>snetPtr</code>	Address of special net.
----------------------	-------------------------

Command Order

Use this command after design import.

Example

```
innovus 77> dbSNetName $s  
vdd
```

The name of the special net is vdd.

dbSNetNext

dbSNetNext *snetPtr*

Returns address of the next special net, 0x0 if there is none.

Parameters

<i>snetPtr</i>	Address of special net.
----------------	-------------------------

Command Order

Use this command after design import.

Example

```
innovus 79> dbSNetNext $s
```

```
0x0
```

There is not another special net.

dbSNetNrTerm

dbSNetNrTerm *snetPtr*

Returns the number of terminals for the specified special net.

Parameters

<i>snetPtr</i>	Address of special net.
----------------	-------------------------

Command Order

Use this command after design import.

Example

```
innovus 80> dbSNetNrTerm $s  
0
```

There are 0 terminals.

dbSNetTermList

```
dbSNetTermList snetPtr
```

Returns address of list of terminals for the specified special net, 0x0 if there is none.

Parameters

<i>snetPtr</i>	Address of special net.
----------------	-------------------------

Command Order

Use this command after design import.

Example

```
innovus 81> dbSNetTermList $s  
0x0
```

There is not a list of terminals.

dbSNetViaList

```
dbSNetViaList snetPtr
```

Returns address of list of vias for the specified special net.

Parameters

<i>snetPtr</i>	Address of special net.
----------------	-------------------------

Command Order

Use this command after design import.

Example

```
innovus 82> dbSNetViaList $s  
0x0
```

There is not a list of vias for the special net.

dbSortHInstByHorOrder

dbSortHInstByHorOrder

```
dbSortHInstByHorOrder hinstPtr  
Sorts the specified hierarchical instances in horizontal order.
```

Parameters

<i>hinstPtr</i>	Address of HInst.
-----------------	-------------------

Command Order

Use this command after importing the design.

Example

```
dbSortHInstByHorOrder $objPtr  
The modules are now sorted by horizontal order.
```

dbSortHInstByVerOrder

dbSortHInstByVerOrder

`dbSortHInstByVerOrder hinstPtr`

Sorts the specified hierarchical instances in vertical order.

Parameters

<code>hinstPtr</code>	Address of HInst.
-----------------------	-------------------

Command Order

Use this command after importing the design.

Example

```
innovus 9> dbSortHInstByVerOrder $objPtr  
innovus 10>
```

The hierarchical instances specified are now sorted in vertical order.

dbSpacingLayer1

dbSpacingLayer1

`dbSpacingLayer1 spacingPtr`

Returns the address of spacing layer 1 for the specified spacing.

Parameters

<code>spacingPtr</code>	Address of spacing layer.
-------------------------	---------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 9> dbSpacingLayer1 [dbExtRuleSpacingList [dbGetExtRuleByName RULE1]]  
0x4985d90
```

The address is 0x4985d90 for spacing layer 1.

dbSpacingLayer2

dbSpacingLayer2

dbSpacingLayer2 spacingPtr

Returns the address of spacing layer 2 for the specified spacing.

Parameters

<i>spacingPtr</i>	Address of spacing layer.
-------------------	---------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 10> dbSpacingLayer2 [dbExtRuleSpacingList [dbGetExtRuleByName RULE1]]  
0x48c1358
```

The address is 0x48c1358 for layer 2.

dbSpacingMinSpacing

dbSpacingMinSpacing

dbSpacingMinSpacing *spacingPtr*

Returns the minimum spacing for the specified spacing.

Parameters

<i>spacingPtr</i>	Address of spacing layer.
-------------------	---------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 11> dbSpacingMinSpacing [dbExtRuleSpacingList [dbGetExtRuleByName RULE1]]  
100
```

The minimum spacing is 0.1.

dbSpacingNext

dbSpacingNext

dbSpacingNext *spacingPtr*

Returns the address of the next spacing rule. The command returns 0x0 if there is none.

Parameters

<i>spacingPtr</i>	Address of spacing layer.
-------------------	---------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 12> dbSpacingNext [dbExtRuleSpacingList [dbGetExtRuleByName RULE1]]  
0x0
```

There is no other spacing rule.

dbStdRowBox

dbStdRowBox

```
dbStdRowBox stdRowPtr
```

Returns the box for the specified standard row.

Parameters

<code>stdRowPtr</code>	Address of standard row.
------------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set box [dbStdRowBox $stdRowPtr]  
    puts "box = $box"  
}
```

```
box = 430300 6721300 6802300 6748300  
box = 430300 6748300 6802300 6775300
```

```
box = 430300 6775300 6802300 6802300
```

There are three boxes for three standard rows.

dbStdRowId

```
dbStdRowId stdRowPtr
```

Returns the standard row ID for the specified standard row.

Parameters

<i>stdRowPtr</i>	Address of standard row.
------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set id [dbStdRowId $stdRowPtr]  
    puts "id = $id"  
}
```

```
id = 231  
id = 232  
id = 233
```

There are three IDs returned for three standard rows.

dbStdRowNext

dbStdRowNext

```
dbStdRowNext stdRowPtr
```

Returns the address of the next row. The command returns 0x0 if there is none.

Parameters

<i>stdRowPtr</i>	Address of standard row.
------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set next [dbStdRowNext $stdRowPtr]  
    puts "next = $next"  
}  
  
next = 0x9659710  
next = 0x9659750  
next = 0x0
```

The next row address is returned until the software reaches the last row (0x0).

dbStdRowNextInReg

dbStdRowNextInReg

```
dbStdRowNextInReg stdRowPtr
```

Returns the next standard regular row address. The command returns 0x0 if there is none.

Parameters

<i>stdRowPtr</i>	Address of standard row.
------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set inreg [dbStdRowNextInReg $stdRowPtr]  
    puts "inreg = $inreg"  
}
```

```
inreg = 0x9659710  
inreg = 0x9659750  
inreg = 0x0
```

The next regular row addresses are returned.

dbStdRowOrient

dbStdRowOrient

```
dbStdRowOrient stdRowPtr
```

Returns the orientation of the specified row.

Parameters

<i>stdRowPtr</i>	Address of standard row.
------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set orient [dbStdRowOrient $stdRowPtr]  
    puts "orient = $orient"  
}
```

```
orient = R0  
orient = MX
```

The orientation for the first row is `R0`, and the orientation for the second row is `MX`.

dbStdRowSite

dbStdRowSite

```
dbStdRowSite stdRowPtr
```

Returns the address of the site for the specified row. The command returns `0x0` if there is none.

Parameters

<i>stdRowPtr</i>	Address of standard row.
------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set site [dbStdRowSite $stdRowPtr]  
    puts "site = $site"  
}
```

Returns the address of the site for every row in the design.

dbStdRowTechSite

dbStdRowTechSite

```
dbStdRowTechSite stdRowPtr
```

Returns the address of the tech site for the specified row. The command returns `0x0` if there is none.

Parameters

<code>stdRowPtr</code>	Address of standard row.
------------------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set site [dbStdRowTechSite $stdRowPtr]  
    puts "site = $site"  
}
```

The script above returns the address of the tech site for every row in the design.

dbStripBoxBox

dbStripBoxBox

`dbStripBoxBox stripBoxPtr`

Returns the box for the specified strip box.

Parameters

<code>stripBoxPtr</code>	Address of stripbox.
--------------------------	----------------------

Command Order

Use this command after importing the design and creating at least one power/ground strap.

Example

```
innovus 23> set s [dbHeadSelList]  
0x96dc618  
innovus 24> set objPtr [dbSelPtr $s]  
0x9653100  
innovus 25> dbStripBoxBox $objPtr  
2380000 330351 2390000 6902329
```

The lower left coordinate for the strip box is (2380.0 330.351), and the upper right coordinate is (2390.0 6902.329).

dbStripBoxList

dbStripBoxList

```
dbStripBoxList stripPtr
```

Returns the address of the list of strip boxes.

Parameters

<i>stripPtr</i>	Address of special net.
-----------------	-------------------------

Command Order

Use this command after importing the design and creating at least one power/ground strap.

Example

```
dbStripBoxList [dbGetSNetByName vdd]
```

The addresses of the list of strip boxes for `vdd` is returned.

dbStripBoxNet

dbStripBoxNet

```
dbStripBoxNet stripBoxPtr
```

Returns the address of the net associated with the specified strip box.

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
--------------------	----------------------

Command Order

Use this command after importing the design and creating at least one power/ground strap.

Example

```
innovus 31> dbStripBoxNet $objPtr  
0x4e10b88
```

The address of the associated strip box net is 0x4e10b88.

dbStripBoxNext

dbStripBoxNext

```
dbStripBoxNext stripBoxPtr
```

Returns the address of the next strip box.

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
--------------------	----------------------

Command Order

Use this command after importing the design and creating at least one power/ground strap.

Example

```
innovus 32> dbStripBoxNext $objPtr  
0x9652e9c
```

The next strip box address is 0x9652e9c.

dbStripBoxShape

dbStripBoxShape

dbStripBoxShape *stripBoxPtr shape*

Returns the shape of the specified strip box. Possible values are:

- 0 (dbcSprNotype)
- 1 (dbcSprRing)
- 2 (dbcSprStripe)
- 3 (dbcSprFollowpin)
- 4 (dbcSprIowire)
- 5 (dbcSprCorewire)
- 6 (dbcSprBlockwire)
- 7 (dbcSprFillwire)
- 8 (dbcSprBlockagewire)
- 9 (dbcSprPadRing)
- 10 (dbcSprBlockRing)
- 11 (dbcSprDrcFill)

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
<i>shape</i>	Shape

Command Order

Use this command after importing the design and creating at least one power/ground strap.

Example

```
innovus 33> dbStripBoxShape $objPtr
```

```
2
```

The strip box is a stripe.

dbStripBoxState

dbStripBoxState

```
dbStripBoxState stripBoxPtr state
```

Returns the state of the specified strip box. Possible values are:

- 0 (dbcSprRouted)
- 1 (dbcSprFixed)
- 2 (dbcSprCover)
- 3 (dbcSprShield)

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
<i>state</i>	State.

Command Order

Use this command after importing the design and creating at least one power/ground strap.

Example

```
innovus 34> dbStripBoxState $stripBoxPtr
0
```

The strip box is routed.

dbStripBoxStrip

dbStripBoxStrip

dbStripBoxStrip *stripBoxPtr*

Returns the address of the specified strip box master.

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
--------------------	----------------------

Command Order

Use this command after importing the design and creating at least one power/ground strap.

Example

dbStripBoxStrip \$objPtr

Returns:

0x4e10b88

The strip box master's address is 0x4e10b88.

dbStripBoxZ

dbStripBoxZ

dbStripBoxZ *stripBoxPtr*

Returns the metal layer for the specified strip box.

Parameters

<i>stripBoxPtr</i>	Address of stripbox.
--------------------	----------------------

Command Order

Use this command after importing the design and creating at least one power/ground strap.

Example

```
innovus 37> dbStripBoxZ $objPtr  
2
```

The metal layer is 2.

dbStripCell

dbStripCell

```
dbStripCell stripPtr
```

Returns the address of cell specified by the strip.

Parameters

<i>stripPtr</i>	Address of special net.
-----------------	-------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 41> dbStripCell [dbGetSNetByName vdd]
```

The address of the cell for `vdd` is returned.

dbStripName

```
dbStripName stripPtr
```

Returns the name of the specified strip. The command returns 0x0 if there is no strip.

Parameters

<i>stripPtr</i>	Address of special net.
-----------------	-------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 42>dbStripName [dbGetSNetByName vdd]  
VDD
```

The name is VDD.

dbStripNext

dbStripNext

```
dbStripNext stripPtr
```

Returns the address of the next strip. The command returns 0x0 if there is none.

Parameters

<i>stripPtr</i>	Address of special net.
-----------------	-------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 42> dbStripNext [dbGetSNetByName vdd]  
0x0
```

There is no next strip.

dbStripSNet

dbStripSNet

`dbStripSNet stripPtr`

Returns the address of the special net associated with the specified strip.

Parameters

<code><i>stripPtr</i></code>	Address of special net.
------------------------------	-------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 43> dbStripSNet [dbGetSNetByName vdd]  
0x4e10b88
```

The address of the special net is `0x4e10b88`.

dbSwitchTermNet

`dbSwitchTermNet termPtr termPtr`

Switches the first term to the second term.

Parameters

<i>termPtr</i>	Address of term.
----------------	------------------

Command Order

Use this command after importing the design.

dbSymArrHInst

dbSymArrHInst

`dbSymArrHInst symArrPtr prefixId`

Returns the address of the symbol array of hierarchical instances.

Parameters

<i>symArrPtr</i>	Address of symbol array.
<i>prefixId</i>	Specifies

Command Order

Use this command after importing the design.

Example

```
innovus 47> set cellPtr [dbgTopCell]
0x72b1a28
innovus 48> dbCellSymArr $cellPtr
0x7d2e8b0
innovus 49> dbSymArrHInst [dbCellSymArr $cellPtr]
```

The address of the symbol array is returned.

dbSymArrNextPrefixIdInHier

dbSymArrNextPrefixIdInHier

`dbSymArrNextPrefixIdInHier symArrPtr prefixId`

Returns the next prefix ID.

Parameters

<code><i>symArrPtr</i></code>	Address of symbol array.
<code><i>prefixId</i></code>	Prefix ID

dbSymArrPrefix

dbSymArrPrefix

`dbSymArrPrefix symArrPtr prefixId`

Returns the prefix for the specified symbol array.

Parameters

<code><i>symArrPtr</i></code>	Address of symbol array.
<code><i>prefixId</i></code>	Prefix ID.

Command Order

Use this command after importing the design.

dbSymHInst

dbSymHInst

dbSymHInst *symPtr*

Returns the address of the hierarchical instance.

Parameters

<i>symPtr</i>	Address of symbol.
---------------	--------------------

Command Order

Use this command after importing the design.

Example

```
innovus 4> set cellPtr [dbgTopCell]  
0x72b0bd0  
innovus 5> dbCellSymArr $cellPtr  
0x7d2d988  
innovus 6> dbSymHInst [dbCellSymArr $cellPtr]  
0x5d36d48
```

The address of the associated module is 0x5d36d48.

dbSymId

dbSymId

dbSymId *symPtr*

Returns the ID of the specified symbol.

Parameters

<i>symPtr</i>	Address of symbol.
---------------	--------------------

Command Order

Use this command after importing the design.

Example

```
innovus 4> set cellPtr [dbgTopCell]  
0x72b0bd0  
innovus 5> dbCellSymArr $cellPtr  
0x7d2d988  
innovus 7> dbSymId [dbCellSymArr $cellPtr]  
0
```

The ID of the symbol is 0.

dbSymNextInArr

dbSymNextInArr

dbSymNextInArr *symPtr*

Returns the address of the next symbol.

Parameters

<i>symPtr</i>	Address of symbol.
---------------	--------------------

Command Order

Use this command after importing the design.

Example

```
innovus 8> dbSymNextInArr [dbCellSymArr $cellPtr]  
0x7d2d998
```

The address of the next symbol is 0x7d2d998.

dbSymNextPrefixIdInHier

dbSymNextPrefixIdInHier

```
dbSymNextPrefixIdInHier symPtr
```

Returns the next prefix ID for the specified symbol.

Parameters

<i>symPtr</i>	Address of symbol.
---------------	--------------------

Command Order

Use this command after importing the design.

Example

```
innovus 9> dbSymNextPrefixIdInHier [dbCellSymArr $cellPtr]  
4460
```

The prefix ID is 4460.

dbSymPrefix

```
dbSymPrefix
```

```
dbSymPrefix symPtr
```

Returns the prefix for the specified symbol.

Parameters

<i>symPtr</i>	Address of symbol.
---------------	--------------------

Command Order

Use this command after importing the design.

T

dbTechSiteName

dbTechSiteName

`dbTechSiteName techSitePtr`

Returns the name of the specified tech site.

Parameters

<i>techSitePtr</i>	Address of tech site.
--------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set site [dbStdRowTechSite $stdRowPtr]  
    set name [dbTechSiteName $site]  
    puts "name = $name"  
}
```

Reports the name of the tech site for every row in the design.

dbTechSiteNext

dbTechSiteNext

`dbTechSiteNext techSitePtr`

Returns the address of the next tech site. The command returns `0x0` if there is none.

Parameters

<code>techSitePtr</code>	Address of tech site.
--------------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 95> dbTechSiteNext [dbGetTechSiteByName tsm3site]
```

```
0x0
```

There is no other tech site.

dbTechSiteSizeX

dbTechSiteSizeX

`dbTechSiteSizeX techSitePtr`

Sets the x dimension of the specified site.

Parameters

<i>techSitePtr</i>	Address of tech site.
--------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set techSitePtr [dbStdRowTechSite $stdRowPtr]  
    set sitex [dbTechSiteSizeX $techSitePtr]  
    puts "sitex = $sitex"  
}
```

Prints the x dimension for every tech site in the design.

dbTechSiteSizeY

dbTechSiteSizeY

```
dbTechSiteSizeY techSitePtr
```

Sets the y dimension of the specified site.

Parameters

<i>techSitePtr</i>	Address of tech site.
--------------------	-----------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachFPlanStdRow [dbHeadFPlan] stdRowPtr {  
    set techSitePtr [dbStdRowTechSite $stdRowPtr]  
    set sitey [dbTechSiteSizeY $techSitePtr]  
    puts "sitey = $sitey"  
}
```

Prints every y dimension for every tech site in the design.

dbTermFTerm

dbTermFTerm

`dbTermFTerm termPtr`

Returns the address of the f-term for the specified term.

Parameters

<code><i>termPtr</i></code>	Address of terminal.
-----------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 26> dbTermFTerm $term  
0x4aa8a00
```

The address is `0x4aa8a00` of the term.

dbTermFTermWPG

dbTermFTermWPG

dbTermFTermWPG *termPtr*

Returns the address of the f-term for the specified power/ground term.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 27> dbTermFTermWPG $term
0x4aa8a00
```

The address of the power/ground term is 0x4aa8a00.

dbTermIdx

dbTermIdx

dbTermIdx *termPtr*

Returns the index for the specified term.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 28> dbTermIdx $term  
1
```

The index is 1.

dbTermIdxWPG

dbTermIdxWPG

`dbTermIdxWPG termPtr`

Returns the term index for the specified term with power/ground

Parameters

<code><i>termPtr</i></code>	Address of terminal.
-----------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 29> dbTermIdxWPG $term  
1
```

The index is 1.

dbTermInst

dbTermInst

dbTermInst *termPtr*

Returns the address of the instance specified by the terminal.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 30> dbTermInst $term
0x7c31c18
```

The address of the instance is 0x7c31c18.

dbTermInstName

dbTermInstName

dbTermInstName *termPtr*

Returns the instance name of the instance attached to the specified term.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 31> dbTermInstName $term  
SH28/I12/I36/CAPCSC/add_339/U28
```

The instance name is SH28/I12/CAPCSC/add_339/U28.

dbTermLayer

dbTermLayer

```
dbTermLayer termPtr
```

Returns the address of the layer of the specified terminal. The command returns `0x0` if there is none.

Parameters

<code>termPtr</code>	Address of terminal.
----------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 32> dbTermLayer $term  
0x0
```

There is no address.

dbTermLibraryContext

dbTermLibraryContext *term*

Returns the library context for the specified terminal.

Parameters

<i>term</i>	Address of terminal.
-------------	----------------------

Command Order

Use this command after importing the design.

dbTermLoc

dbTermLoc

dbTermLoc *termPtr*

Returns the location for the specified term.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 33> dbTermLoc $term
```

2242250 3982600

The location is (2242.25 3982.6).

dbTermLocWithZ

dbTermLocWithZ

dbTermLocWithZ *termPtr*

Returns the location and the metal layer for the specified term.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 33>dbTermLocWithZ $termPtr  
2242250 3982600 3
```

The location is (2242.250 3982.6) and the term is on metal 3.

dbTermLogicValue

dbTermLogicValue *termPtr*

Returns the logic value for the specified term.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 33>dbTermLogicValue [dbGetTermByName [dbGetInstByName DTMF/i_10048] A]  
0
```

The logic value of the instance `DTMF/i_10048` term `A` is `0`.

dbTermMaxCap

```
dbTermMaxCap termPtr
```

Returns the maximum capacitance for the specified term. The command returns `-1` if there is no maximum capacitance specified for the term.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 33>dbTermMaxCap [dbGetTermByName [dbGetInstByName DTMF/i_10048] A]  
0
```

There is no maximum capacitance specified for term `A` of instance `DTMF/i_10048`.

dbTermMaxTran

dbTermMaxTran *termPtr*

Returns the maximum transition for the specified term (in DB units). The command returns `-1` if there is no maximum transition specified for the term.

Parameters

<i>termPtr</i>	Address of the terminal.
----------------	--------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 33>dbTermMaxTran [dbGetTermByName [dbGetInstByName DTMF/i_10048] A]  
-1
```

There is no maximum transition specified for term `A` of instance `DTMF/i_10048`.

dbTermMaxTranSlack

dbTermMaxTranSlack *termPtr*

Returns the max transition of the specified term. The command returns `2147483647` if there is no maximum transition set for the term.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 34> dbTermMaxTranSlack [dbGetTermByName [dbGetInstByName SH17/I340] Q]  
2147483647
```

There is no maximum transition for the `Q` term of instance `SH17/I340`.

dbTermName

dbTermName

`dbTermName termPtr`

Returns the name of the specified term.

Parameters

<code>termPtr</code>	Address of terminal.
----------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 34> dbTermName $term  
A
```

The term name is `A`.

dbTermNet

dbTermNet

dbTermNet *termPtr*

Returns the address of the net of the specified terminal.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 35> dbTermNet $term
0x72d9cc0
```

The address of the net is 0x72d9cc0.

dbTermNextInInst

dbTermNextInInst

dbTermNextInInst *termPtr*

Returns the address of the next terminal. The command returns 0x0 if there is none.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 36> dbTermNextInInst $term  
0x0
```

There is no next terminal.

dbTermNextInInstArr

dbTermNextInInstArr

```
dbTermNextInInstArr termPtr
```

Returns the address of the next terminal in the array for the specified term. The command returns 0x0 if there is none.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 37> dbTermNextInInstArr $term  
0x0
```

There is no next term.

dbTermNextInNet

dbTermNextInNet

dbTermNextInNet *termPtr*

Returns the address of the next terminal on the net for the specified term.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 38> dbTermNextInNet $term
0x79a0300
```

The address of the next term is 0x79a0300.

dbTermTranTime

dbTermTranTime

dbTermTranTime *termPtr*

Returns the transition time for the specified terminal.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after performing timing analysis.

Example

```
innovus 39>dbForEachInstTerm $objPtr termPtr{  
    set trans [dbTermTranTime $termPtr]  
    puts "trans = $trans"
```

```
}
```

```
trans = 2962 2737  
trans = 889 752  
trans = 3378 3000  
trans = 1200 1200
```

The transition of all the terms for a specified instance are returned.

dbTermZ

dbTermZ

```
dbTermZ termPtr
```

Returns the metal layer for the specified term.

Parameters

<i>termPtr</i>	Address of terminal.
----------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 39> dbTermZ $term  
0
```

The term does not currently have a metal layer.

dbTimeArcRelFTerm

dbTimeArcRelFTerm *timeArcPtr*

Returns the address of the from pin for the specified timing arc.

Parameters

<i>timeArcPtr</i>	Address of timing library.
-------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 45>dbForEachCellTimeArc $cellPtr timeArcPtr {  
    set relPtr [dbTimeArcRelFTerm $timeArcPtr]  
    puts "Address = $relPtr"
```

Address = 0x7c2d2e0

The address of the from pin is 0x7c2d2e0.

dbTimeArcTgtFTerm

dbTimeArcTgtFTerm *timeArcPtr*

Returns the address of the to pin for the specified timing arc.

Parameters

<i>timeArcPtr</i>	Address of timing library.
-------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 45>dbForEachCellTimeArc $cellPtr timeArcPtr {  
    set relPtr [dbTimeArcTgtFTerm $timeArcPtr]  
    puts "Address = $relPtr"
```

Address = 0x7c2d2e0

The address of the to pin is 0x7c2d2e0.

dbTimeArcType

```
dbTimeArcType timeArcPtr
```

Returns the type of the specified timing arc.

Possible values are:

0 Combinational	26 nochange_high_low
1 rising_edge	27 nochange_low_high
2 falling_edge	28 nochange_low_low
3 preset	29 nochange_setup_rising
4 clear	30 nochange_setup_falling

5 setup_rising	31 nochange_hold_rising
6 setup_falling	32 nochange_hold_falling
7 hold_rising	33 min_pulse_width_rising
8 hold_falling	34 min_pulse_width_falling
9 three_state_disable	35 min_period_rising
10 recovery_rising	36 min_period_falling
11 recovery_falling	37 combinational_rise
12 removal_rising	38 combinational_fall
13 removal_falling	39 three_state_enable, rising edge
14 skew_rising	40 three_state_enable, falling edge
15 skew_falling	41 three_state_disable, rising edge
20 three_state_enable	42 three_state_disable, falling edge
21 non_seq_setup_rising	47 min_pulse_width
22 non_seq_setup_falling	48 minimum_period
23 non_seq_hold_rising	49 min_clock_tree_path
24 non_seq_hold_falling	50 max_clock_tree_path
25 nochange_high_high	

Parameters

<i>timeArcPtr</i>	Address of timing library.
-------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 45>dbForEachCellTimeArc $cellPtr timeArcPtr {
    set type [dbTimeArcType $timeArcPtr]
```

```
puts "Type = $type"
```

```
Type = 4
Type = 4
Type = 6
Type = 8
Type = 24
```

dbTimeLibDefOpCond

dbTimeLibDefOpCond

```
dbTimeLibDefOpCond timeLibPtr
```

Returns the address of the operating conditions defined by DEF.

Parameters

<i>timeLibPtr</i>	Address of timing library.
-------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 45> dbTimeLibDefOpCond [dbGetTimeLibByName t25lib_tt]
0x7c2d2e0
```

The address is 0x7c2d2e0 for the DEF operating conditions.

dbTimeLibInName

```
dbTimeLibInName timeLibPtr
```

Returns the timing library name set inside the specified timing library.

Parameters

<i>timeLibPtr</i>	Address of timing library.
-------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 46> dbTimeLibInName [dbGetTimeLibByName t25lib_tt]  
t25lib_tt
```

The timing library name is `t25lib_tt`.

dbTimeLibLibraryContext

```
dbTimeLibLibraryContext timelib
```

Returns the timing library context for the specified timing library.

Parameters

<i>timeLib</i>	Address of timing library.
----------------	----------------------------

Command Order

Use this command after importing the design.

dbTimeLibName

dbTimeLibName

dbTimeLibName *timeLibPtr*

Returns the timing library name for the specified timing library.

Parameters

<i>timeLibPtr</i>	Address of timing library.
-------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 46> dbTimeLibName [dbGetTimeLibByName t25lib_tt]  
t25lib_tt
```

The timing libraries name is t25lib_tt.

dbTimeLibNext

dbTimeLibNext

dbTimeLibNext *timeLibPtr*

Returns the address of the next timing library. The command returns 0x0 if there is none.

Parameters

<i>timeLibPtr</i>	Address of timing library.
-------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 47> dbTimeLibNext [dbGetTimeLibByName t25lib_tt]  
0x0
```

There is no other timing library.

dbTimeLibNomOpCond

`dbTimeLibNomOpCond timeLibPtr`

Returns the address of the nominal operating condition for the specified library. The command returns 0x0 if there is none.

Parameters

<code>timeLibPtr</code>	Address of timing library.
-------------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 47>dbForEachHeadTimeLib [dbgHead] p {  
    dbForEachTimeLibOpCond $p opcondPtr{  
        puts "address = \$opcondPtr  
    } }  
  
0xe3c0cc4  
0xe3c0c98
```

There are two nominal operating conditions in the design and their addresses are returned.

dbTimeLibTimeLibGrp

dbTimeLibTimeLibGrp *timeLibPtr*

Returns the address of the library group for the specified timing library. The command returns 0x0 if there is none.

Parameters

<i>timeLibPtr</i>	Address of timing library.
-------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 47>dbForEachHeadTimeLib [dbgHead] p {  
dbTimeLibTimeLibGrp $p  
}
```

0xe3c0ba4

The address for the timing group for the design is 0xe3c0ba4.

dbTimeLibTimeLibGrpName

dbTimeLibTimeLibGrpName *timeLibPtr*

Returns the name of the library group for the specified timing library. The command returns 0x0 if there is none.

Parameters

<i>timeLibPtr</i>	Address of timing library.
-------------------	----------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 47>dbForEachHeadTimeLib [dbgHead] p {  
    set name [dbTimeLibTimeLibGrpName $p]  
    puts "name = $name"  
}  
  
name = common
```

The name of the timing group is `common`.

dbTlsCellCell

```
dbTlsCellCell tlcell
```

Returns the master cell for the specified timing library cell. All non-timing library `dbCell*` commands can be run on this master cell.

Parameters

<i>tlcell</i>	Address of timing library cell.
---------------	---------------------------------

Command Order

Use this command after importing the design.

dbTIsCellLibraryContext

`dbTIsCellLibraryContext tlsCell`

Returns the library context for the specified timing library cell.

Parameters

<code><i>tlsCell</i></code>	Address of timing library cell.
-----------------------------	---------------------------------

Command Order

Use this command after importing the design.

dbTIsCellTimeLib

`dbTIsCellTimeLib t1Cell`

Returns the timing library pointer for the specified timing library cell.

Parameters

<code><i>t1Cell</i></code>	Address of timing library cell.
----------------------------	---------------------------------

Command Order

Use this command after importing the design.

dbTrimCellUnplacedYetConnectedInsts

dbTrimCellUnplacedYetConnectedInsts

```
dbTrimCellUnplacedYetConnectedInsts cellPtr {-update | -check}
```

Trims all terms of unplaced instances. This is used by the router to remove unplaced cells that have routable connections. Also returns the number of unplaced and "non-trivially connected" instances.

Parameters

<i>cellPtr</i>	Address of cell.
-update	Allows the netlist to be updated.
-check	Performs check.

Command Order

Use this command after performing placement.

Example

```
innovus 83> dbTrimCellUnplacedYetConnectedInsts -check
```

U

dbUnassignBump

```
dbUnassignBump bumpPtr
```

Unassigns the specified bump.

Parameters

<i>bumpPtr</i>	Address of bump.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
innovus 51>dbUnassignBump $objPtr  
innovus 52>
```

The bump is now unassigned.

dbUnassignTilePin

`dbUnassignTilePin instPtr termName`

Unassigns the terminal name from the specified instance.

Parameters

<i>instPtr</i>	Address of instance.
<i>termName</i>	Name of terminal.

Command Order

Use this command after importing the design.

Example

```
innovus 72>dbUnassignTilePin $instPtr IN
```

The term `IN` is now unassigned.

dbUnflattenHInst

dbUnflattenHInst *hinstPtr*

Unflattens specified hierarchical instance.

Parameters

<i>hinstPtr</i>	Address of hierarchical instance.
-----------------	-----------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 83>dbUnflattenHInst $instPtr
```

```
innovus 84>
```

The module is now unflattened.

dbUnfreezeAllInst

dbUnfreezeAllInst *vinstPtr*

Used by signal integrity and timing optimization operations to unfreeze all instances.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellVInst $cellPtr vinstPtr {  
    dbUnfreezeAllInst $vinstPtr  
  
}
```

Unfreezes all instances for the specified cell.

dbUnfreezeWindowArea

```
dbUnfreezeWindowArea vinstPtr
```

Used by signal integrity and timing optimization operations to unfreeze instances of the window area.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellVInst $cellPtr vinstPtr {  
    dbUnfreezeWindowArea $vinstPtr  
}
```

Unfreezes instances of the window area for the specified cell.

dbUpdateAllWireTerm

```
dbUpdateAllWireTerm topCellPtr
```

Updates all wire terms for the specified top cell.

Parameters

<i>topCellptr</i>	Address of top cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 83>dbUpdateAllWireTerm [dbgHeadTopCell]  
innovus 84>
```

The wire terms are updated.

dbUpdateIoSidesFromLocs

dbUpdateIoSidesFromLocs

Updates all I/Os on all sides.

Command Order

Use this command after performing placement.

Example

```
innovus 83>dbUpdateIoSidesFromLocs  
innovus 84>
```

dbUpdateNetTiming

dbUpdateNetTiming *netPtr*

Updates the timing for the specified net.

Parameters

<i>netPtr</i>	Address of net.
---------------	-----------------

Command Order

Use this command after timing analysis.

Example

```
innovus 83>dbUpdateNetTiming [dbGetNetByName BLU]
```

```
innovus 84>
```

Updates the timing for the net called BLU.

dbUpdatePinGuideRouteBox

dbUpdatePinGuideRouteBox

dbUpdatePinGuideRouteBox *rboxPtr rgName pinGroupName layerZ mSP*

Updates the specified route guide with new data. This command is automatically called by the GUI when you edit a pin guide box (route box).

Parameters

<i>rboxPtr</i>	Address of route box.
<i>rgName</i>	Route guide name.

<i>pinGroupCellName</i>	Pin group name.
<i>layerZ</i>	Metal layer.
<i>mSP</i>	Routing blockages for special routes.

Command Order

Use this command after importing the design and creating at least one pin guide.

dbUpdateTermZFromFTerm

dbUpdateTermZFromFTerm

Updates term metal layers based on f-terms

Command Order

Use this command after importing the design.

Example

dbUpdateTermZFromFTerm

The term metal layers are updated.

V

dbVerifyAllFTermLeaf

dbVerifyAllFTermLeaf *headPtr*

Returns 1 if all leaf cells have a pin and a pin capacitance specified and 0 if they do not.

Parameters

<i>headPtr</i>	Address of head
----------------	-----------------

Command Order

Use this command after importing the design.

Example

```
innovus 83>dbVerifyAllFTermLeaf [dbgHead]  
1  
innovus 84>
```

dbViaCellContactId

dbViaCellContactId viaCellPtr

Returns ID of the specified via cell contact.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

dbViaCellCutBox

dbViaCellCutBox

dbViaCellCutBox viaCellPtr

Returns the cut box of the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 9> dbViaCellCutBox [dbGetViaCellByName VIA3]  
-160 -160 160 160
```

The lower left coordinate of the cut for VIA3 is at (-0.16 -0.16), and the upper right is at (0.16 0.16).

dbViaCellCutLayer

dbViaCellCutLayer

```
dbViaCellCutLayer viaCellPtr
```

Returns the address of the cut layer for the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 10> dbViaCellCutLayer [dbGetViaCellByName VIA3]  
0x4a799a0
```

The address of the cut layer for *VIA3* is 0x4a799a0.

dbViaCellDx

dbViaCellDx

dbViaCellDx *viaCellPtr*

Returns the x dimension for the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 11> dbViaCellDx [dbGetViaCellByName VIA3]  
400
```

The x dimension for *VIA3* is 0.4.

dbViaCellDy

dbViaCellDy

dbViaCellDy *viaCellPtr*

Returns the y dimension for the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 12> dbViaCellDy [dbGetViaCellByName VIA3]  
400
```

The y dimension for *VIA3* is 0.4.

dbViaCellHiBox

dbViaCellHiBox

```
dbViaCellHiBox viaCellPtr
```

Returns the box of the metal layer above the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 13> dbViaCellHiBox [dbGetViaCellByName VIA3]  
-200 -200 200 200
```

The metal layer above *VIA3* has a lower left coordinate of (-0.2 -0.2), and the upper right coordinate is at (0.2 0.2).

dbViaCellHiLayer

dbViaCellHiLayer

`dbViaCellHiLayer viaCellPtr`

Returns the address of the metal layer above the specified via cell.

Parameters

<code>viaCellPtr</code>	Address of via cell.
-------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 14> dbViaCellHiLayer [dbGetViaCellByName VIA3]  
0x48c1338
```

The address of the metal layer above *VIA3* is 0x48c1338.

dbViaCellListViaCell

`dbViaCellListViaCell viaCellPtr`

Returns an address of the list of via cells for the specified via cell.

Parameters

<code>viaCellPtr</code>	Address of via cell.
-------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 15> dbViaCellListViaCell [dbgTopCell]  
0xe0006002
```

The list's address is `0xe0006002`.

dbViaCellLoBox

dbViaCellLoBox

```
dbViaCellLoBox viaCellPtr
```

Returns the box of the metal layer below the specified via cell.

Parameters

<code>viaCellPtr</code>	Address of via cell.
-------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 15> dbViaCellLoBox [dbGetViaCellByName VIA3]  
-200 -200 200 200
```

The lower left coordinate of the metal layer below `VIA3` is $(-0.2 \ -0.2)$, and the upper right coordinate is at $(0.2$

0.2).

dbViaCellLoLayer

dbViaCellLoLayer

`dbViaCellLoLayer viaCellPtr`

Returns the address of the metal layer below the specified via cell.

Parameters

<code>viaCellPtr</code>	Address of via cell.
-------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 16> dbViaCellLoLayer [dbGetViaCellByName VIA3]  
0x48c0f18
```

The address of the metal layer below `VIA3` is `0x48c0f18`.

dbViaCellLx

dbViaCellLx

`dbViaCellLx viaCellPtr`

Returns the lower left x coordinate of the via for the specified via cell.

Parameters

<code>viaCellPtr</code>	Address of via cell.
-------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 17> dbViaCellLx [dbGetViaCellByName VIA3]  
-200
```

The lower left x coordinate is at `-0.2`.

dbViaCellLy

dbViaCellLy

```
dbViaCellLy viaCellPtr
```

Returns the lower left y coordinate of the via for the specified via.

Parameters

<code>viaCellPtr</code>	Address of via cell.
-------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 18> dbViaCellLy [dbGetViaCellByName VIA3]  
-200
```

The lower left y coordinate for `VIA3` is `-0.2`.

dbViaCellName

dbViaCellName

dbViaCellName *viaCellPtr*

Returns the name of the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 20> dbViaCellName $viaCellPtr
VIA3
```

The via's name is VIA3.

dbViaCellNext

dbViaCellNext

dbViaCellNext *viaCellPtr*

Returns the address of the next via cell. The command returns 0x0 if there is none.

Parameters

<code>viaCellPtr</code>	Address of via cell.
-------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 21> dbViaCellNext $viaCellPtr  
0x0
```

There is no next via cell.

dbViaCellPatternName

dbViaCellPatternName

```
dbViaCellPatternName viaCellPtr
```

Returns the pattern of the specified via cell. The command returns nothing if there is no pattern.

Parameters

<code>viaCellPtr</code>	Address of via cell.
-------------------------	----------------------

Command Order

Use this command after importing the design.

dbViaCellRectanglesNext

```
dbViaCellRectanglesNext viaCellPtr
```

Returns the next rectangle for the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
set iter [dbIterViaCellRectangles LAYER $vc]
while {[set rect [dbViaCellRectangleNext $iter]] != ""} {
    # $rect is list of four coordinates: lx ly hx hy
    ...
    if (want to break out of the iterator loop)
        dbEndIterViaCellRectangles $iter
    ...
}
```

The code above cycles through all of the rectangles for a specified via cell.

dbViaCellRes

dbViaCellRes

dbViaCellRes *viaCellPtr*

Returns the resistance for the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 42> dbViaCellRes [dbGetViaCellByName VIA3]  
3
```

dbViaCellRuleId

dbViaCellRuleId

`dbViaCellRuleId viaCellPtr`

Returns the rule ID for the specified via cell.

Parameters

<code>viaCellPtr</code>	Address of via cell.
-------------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 42> dbViaCellRuleId [dbGetViaCellByName VIA3]  
0
```

The rule ID is 0 for VIA3.

dbViaCellViaCode

dbViaCellViaCode

dbViaCellViaCode *viaCellPtr*

Returns the via code for the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 23> dbViaCellViaCode [dbGetViaCellByName VIA3]  
5
```

The via code is 5 for VIA3.

dbViaCellZ

dbViaCellZ

dbViaCellZ *viaCellPtr*

Returns the upper metal layer for the specified via cell.

Parameters

<i>viaCellPtr</i>	Address of via cell.
-------------------	----------------------

Command Order

Use this command after importing the design.

Example

```
innovus 24> dbViaCellZ [dbGetViaCellByName VIA3]  
3
```

The upper metal layer for *VIA3* is 3.

dbViaLoc

dbViaLoc

```
dbViaLoc viaPtr
```

Returns the location of the specified via.

Parameters

<i>viaPtr</i>	Address of via.
---------------	-----------------

Command Order

Use this command after routing the design.

Example

```
innovus 45> dbViaLoc $viaPtr  
34500 47500
```

The via is located at (34.5 47.5).

dbViaViaCell

dbViaViaCell

dbViaViaCell *viaPtr*

Returns the address of the master for the specified via.

Parameters

<i>viaPtr</i>	Address of via.
---------------	-----------------

Command Order

Use this command after importing the design.

dbVInstCell

dbVInstCell *vinstPtr*

Returns the address of the cell for the specified Verilog instance.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after importing the design.

dbVInstName

dbVInstName

```
dbVInstName vinstPtr
```

Returns the name of the Verilog instance for the specified Verilog instance.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellVInst $cellPtr vinstPtr {  
    set name [dbVInstName $vinstPtr]  
    puts "name = $name"  
}
```

The script above prints all of the Verilog instance names for the specified cell.

dbVInstNrTerm

dbVInstNrTerm

```
dbVInstNrTerm vinstPtr
```

Returns the number of terminals for the specified Verilog instance.

Parameters

<i>vinstPtr</i>	Address of Verilog instance.
-----------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellVInst $cellPtr vinstPtr {  
    set nr [dbVInstNrTerm $vinstPtr]  
    puts "nr = $nr"  
}
```

The script above prints out the number of terminals for each Verilog instance for the specified cell.

dbVNetBus

```
dbVNetBus vnetPtr
```

Returns the address of the bus for the specified Verilog net.

Parameters

<i>vnetPtr</i>	Address of Verilog net.
----------------	-------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellVNet $cellPtr vnetPtr {  
    set name [dbVNetBus $vnetPtr]  
    puts "bus name = $name"  
}
```

The above script reports the names of all of the buses for the specified cell.

dbVNetName

dbVNetName

```
dbVNetName vnetPtr
```

Returns the name of the specified Verilog net.

Parameters

<i>vnetPtr</i>	Address of Verilog net.
----------------	-------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellVNet $cellPtr vnetPtr {  
    set name [dbVNetName $vnetPtr]  
    puts "name = $name"  
}
```

The script above returns the name of all the Verilog nets for the specified cell.

dbVTermFTerm

dbVTermFTerm

```
dbVTermFTerm vtermPtr
```

Returns the f-term address of the specified Verilog terminal.

Parameters

<i>vtermPtr</i>	Address of Verilog terminal.
-----------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellVNet $cellPtr vnetPtr {  
    dbForEachVNetVTerm $vnetPtr vtermPtr {  
        set term [dbVTermFTerm $vtermPtr]  
    }  
}
```

The script above sets the variable term to the address of a term associated with the Verilog terminal for each Verilog net of the specified cell.

dbVTermInst

dbVTermInst

```
dbVTermInst vtermPtr
```

Returns the address of the instance specified by the Verilog terminal.

Parameters

<i>vtermPtr</i>	Address of Verilog terminal.
-----------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
dbForEachCellVNet $cellPtr vnetPtr {  
    dbForEachVNetVTerm $vnetPtr vtermPtr {  
        set inst [dbVTermInst $vtermPtr]  
    }  
}
```

The script above set the variable instance to the address of the instance associated with the Verilog terminal for every Verilog net in the specified cell.

dbVTermNet

dbVTermNet

```
dbVTermNet vtermPtr
```

Returns the address of the net for the specified Verilog terminal.

Parameters

<i>vtermPtr</i>	Address of Verilog terminal.
-----------------	------------------------------

Command Order

Use this command after importing the design.

Example

```
innovus 26> dbVTermNet $objPtr  
0x805bbb4
```

The address of the net is 0x805bbb4.

W

dbWasHeadDesignChanged

dbWasHeadDesignChanged

dbWasHeadDesignChanged *headPtr*

Returns 1 if the design was changed and 0 if it was not.

Parameters

<i>headPtr</i>	Address of Head.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
innovus 52> dbWasHeadDesignChanged [dbgHead]  
0
```

The design was not changed.

dbWasSelSetChanged

dbWasSelSetChanged

dbWasSelSetChanged

Returns 1 if the selection set was changed and 0 if it was not.

Command Order

Use this command after importing the design.

Example

```
innovus 53> dbWasSelSetChanged  
0
```

The set was not changed.

dbWireAdjD

dbWireAdjD

```
dbWireAdjD wirePtr
```

Returns the address of the adjacent wire that is below the specified wire. The command returns `0x0` if there is none.

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 31> dbWireAdjD $objPtr  
0xd8cc104
```

The adjacent wire's address is `0xd8cc104`.

dbWireAdjDir

dbWireAdjDir

`dbWireAdjDir wirePtr`

Returns the direction of the adjacent wire. Possible values are: N, W, S, E, U, D.

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 32> dbWireAdjDir $objPtr
```

```
N
```

The direction is north.

dbWireAdjE

dbWireAdjE

`dbWireAdjE wirePtr`

Returns the address of any adjacent wire to the east of the specified wire. The command returns `0x0` if there is none.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 33> dbWireAdjE $objPtr  
0x0
```

There is no adjacent wire to the east of the specified wire.

dbWireAdjN

dbWireAdjN

```
dbWireAdjN wirePtr
```

Returns the address of any adjacent wire to the north of the specified wire. The command returns `0x0` if there is none.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 34> dbWireAdjN $objPtr  
0xd8cc14c
```

The address of the adjacent wire to the north is 0xd8cc14c.

dbWireAdjS

dbWireAdjS

dbWireAdjS *wirePtr*

Returns the address of any adjacent wire to the south of the specified wire. The command returns 0x0 if there is none.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 35> dbWireAdjS $objPtr
0x0
```

There is no adjacent wire to the south of the wire.

dbWireAdjU

dbWireAdjU

dbWireAdjU *wirePtr*

Returns the address of any adjacent wire above the specified wire. The command returns 0x0 if there is none.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 30> dbWireAdjU $objPtr  
0x0
```

There is no adjacent wire above the wire.

dbWireAdjW

dbWireAdjW

```
dbWireAdjW wirePtr
```

Returns the address of any adjacent wire to the west of the specified wire. The command returns `0x0` if there is none.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 29> dbWireAdjW $objPtr
```

0x0

There is no adjacent wire to the west.

dbWireBox

dbWireBox

dbWireBox *wirePtr*

Returns the box of the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 27> dbWireBox $objPtr
150137 94250 150337 267000
```

The lower left coordinate of the box of the specified wire is (150.137 94.250), and the upper right coordinate is at (150.337 267.000).

dbWireCleanup

dbWireCleanup

dbWireCleanup

Deletes all signal wires in a design.

Command Order

Use this command after routing the design.

dbWireConnect

dbWireConnect

```
dbWireConnect w1 w2
```

Connects the wire specified by *w1* to the wire specified by *w2*.

Parameters

<i>w1</i>	Address of first wire.
<i>w2</i>	Address of second wire.

Command Order

Use this command after routing the design.

dbWireCreate

```
dbWireCreate net x y z
```

Creates a wire on the net located at *x* *y* on the layer specified by *z*.

Parameters

<i>net</i>	Address of net.
<i>x</i>	x coordinate for net.
<i>y</i>	y coordinate for net.

<i>z</i>	Layer
----------	-------

Command Order

Use this command after performing placement.

dbWireDegree

dbWireDegree

`dbWireDegree wirePtr`

Returns the degree of the specified wire. Possible values are 1, 2, 3, 4, 5, and 6.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 37> dbWireDegree $objPtr
```

```
2
```

The degree is 2.

dbWireDir

dbWireDir

`dbWireDir wirePtr`

Returns the direction of the specified wire. Possible values are:

`dbcWireN`

`dbcWireS`

`dbcWireE`

`dbcWireW`

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 38> dbWireDir $objPtr  
dbcWireN
```

The wire direction is north.

dbWireDnViaCell

dbWireDnViaCell

`dbWireDnViaCell wirePtr`

Returns address of the via cell to be used to connect to the metal layer below the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 39> dbWireDnViaCell $objPtr  
0xac2b3c0
```

The address of the via is 0xac2b3c0.

dbWireDnViaRuleId

dbWireDnViaRuleId

```
dbWireDnViaRuleId wirePtr
```

Returns the rule ID for the via to be used to connect to the metal layer below the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
dbWireDnViaRuleId $objPtr
```

Returns:

0

The rule ID is 0.

dbWireDup

dbWireDup

`dbWireDup wirePtr`

Returns the address of the duplicate wire for the specified wire. The command returns `0x0` if there is no duplicate wire.

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

Command Order

Use this command after routing the design.

Example

`dbWireDup $objPtr`

Returns:

`0x0`

There is no duplicate wire.

dbWireDupSafe

dbWireDupSafe

`dbWireDupSafe wirePtr`

Returns the address of the duplicate wire for the specified wire and checks it. The command returns `0x0` if there is no duplicate wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

dbWireExt1

dbWireExt1

dbWireExt1 *wirePtr*

Returns the wire extension for the specified wire. Possible values are:

- 0 (dbcWireExtHalf)
- 1 (dbcWireExtZero)
- 2 (dbcWireExtRule)

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

dbWireExt1 \$objPtr

Returns:

0

The wire extension is half.

dbWireExt2

dbWireExt2

`dbWireExt2 wirePtr`

Returns the extension of the specified wire. Possible values are:

- 0 (dbcWireExtHalf)
- 1 (dbcWireExtZero)
- 2 (dbcWireExtRule)

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

Command Order

Use this command after routing the design.

dbWireGetOrCreateInWireDir

dbWireGetOrCreateInWireDir

`dbWireGetOrCreateInWireDir wirePtr x y dir`

Returns or creates a wire in the specified direction.

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

x	X coordinate.
y	Y coordinate.
dir	

Command Order

Use this command after importing the design.

Example

```
dbWireGetOrCreateInWireDir $objPtr 500 500 N
```

Returns:

```
0x9ef2bf0
```

The address of the wire at (0.5 0.5) with a direction of north is 0x9ef2bf0.

dbWireHead

dbWireHead

```
dbWireHead wirePtr
```

Returns the wire head address for the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

dbWireHeadWidth

dbWireHeadWidth

dbWireHeadWidth *wirePtr*

Returns the wire head width for the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

dbWireHeadWidth \$objPtr

Returns:

200

The wire width is 0.2.

dbWireLen

dbWireLen

dbWireLen *wirePtr*

Returns the wire length for the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
dbWireLen $objPtr
```

Returns:

109438

The wire length is 109.438.

dbWireLoc

dbWireLoc

```
dbWireLoc wirePtr
```

Returns the location of the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
dbWireLoc $objPtr
```

Returns:

81387 142800

The location of the wire is at (81.387 142.8).

dbWireLocWithZ

dbWireLocWithZ

```
dbWireLocWithZ wirePtr
```

Returns the location of the specified wire and the metal layer for the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after importing the design.

Example

```
dbWireLocWithZ $objPtr
```

Returns:

```
81387 142800 3
```

The location of the wire is (81.387 142.8) and is on metal 3.

dbWireNet

dbWireNet

```
dbWireNet wirePtr
```

Returns the address of the net associated with the wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
dbWireNet $objPtr
```

Returns:

```
0xd5d81d4
```

The net address is 0xd5d81d4.

dbWireNextInNet

dbWireNextInNet

```
dbWireNextInNet wirePtr
```

Returns the address of the next net.

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 8> dbWireNextInNet $objPtr
0xd86b1e8
```

The next net address is 0xd86b1e8.

dbWireRawTerm

dbWireRawTerm

dbWireRawTerm *wirePtr*

Returns the term reference address of the wire on which there maybe a term. The command returns 0x0 if there is no term.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 9> dbWireRawTerm $objPtr
0x0
```

There is no term.

dbWireRuleId

dbWireRuleId

dbWireRuleId *wirePtr*

Returns rule ID for the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 10> dbWireRuleId $objPtr  
0
```

The rule ID is 0.

dbWireRuleIdx

dbWireRuleIdx

```
dbWireRuleIdx wirePtr
```

Returns rule index for the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

dbWireTerm

dbWireTerm

```
dbWireTerm wirePtr
```

Returns the address of the term for the associated specified wire. The command returns `0x0` if there is no term.

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 11> dbWireTerm $objPtr  
0x0
```

There is no term.

dbWireViaDnId

dbWireViaDnId

```
dbWireViaDnId wirePtr
```

Returns the ID of the via to be used to connect to the metal layer below the specified wire.

Parameters

<code>wirePtr</code>	SAddress of route.
----------------------	--------------------

Command Order

Use this command after routing the design.

Example

```
innovus 12> dbWireViaDnId $objPtr  
9
```

The ID is 9.

dbWireViaDnRuleIdx

dbWireViaDnRuleIdx

`dbWireViaDnRuleIdx wirePtr`

Returns the index for the via to be used to connect to the metal layer below the specified wire.

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 12>dbWireViaDnRuleIdx $objPtr  
9
```

The index is 9.

dbWireViaDnStatus

dbWireViaDnStatus

dbWireViaDnStatus *wirePtr*

Returns the status of the specified wire. Possible values are:

dbcNoWire
dbcRoutedWire
dbcFixedWire
dbcCoverWire
dbcShieldWire
dbcNoShieldWire
dbcGlobalWire

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 13> dbWireViaDnStatus $objPtr  
dbcNoWire
```

There is no wire.

dbWireWidthXId

dbWireWidthXId

dbWireWidthXId *wirePtr*

Returns the ID of the x dimension wire width for the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
innovus 14>dbWireWidthXId $wirePtr  
4
```

The ID is 4.

dbWireWidthXIdx

dbWireWidthXIdx *wirePtr*

Returns the index of the x dimension wire width for the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

```
dbWireWidthXIdx $wirePtr
```

Returns:

0

The index is 0.

dbWireWidthYIdx

```
dbWireWidthYIdx wirePtr
```

Returns the index of the y dimension wire width for the specified wire.

Parameters

<code>wirePtr</code>	Address of wire.
----------------------	------------------

Command Order

Use this command after routing the design.

Example

```
dbWireWidthYIdx $wirePtr
```

Returns:

6

The index is 6.

dbWireWireStatus

dbWireWireStatus

```
dbWireWireStatus wirePtr
```

Returns the status of the specified wire. Possible values are:

dbcNoWire
dbcRoutedWire
dbcFixedWire
dbcCoverWire
dbcShieldWire
dbcNoShieldWire
dbcGlobalWire

Parameters

<i>wirePtr</i>	Address of route.
----------------	-------------------

Command Order

Use this command after routing the design.

Example

```
dbWireWireStatus $objPtr
```

Returns:

dbcRoutedWir

The wire is routed.

dbWireZ

dbWireZ

dbWireZ *wirePtr*

Returns the metal layer for the specified wire.

Parameters

<i>wirePtr</i>	Address of wire.
----------------	------------------

Command Order

Use this command after routing the design.

Example

dbWireZ \$objPtr

Returns:

5

The wire is on metal layer 5.

dbg Commands

- [dbgCellList](#)
- [dbgConvertSetupHoldOnAsyncToRecRem](#)
- [dbgCurRuler](#)
- [dbgDBUPerIGU](#)
- [dbgDBUPerloGU](#)
- [dbgDBUPerMGrid](#)
- [dbgDesignName](#)
- [dbgHead](#)
- [dbgHierChar](#)
- [dbgIsHInstRootX](#)
- [dbgLayerArr](#)
- [dbgMicronPerDBU](#)
- [dbgNrLayer](#)
- [dbgPicoFPerDBU](#)
- [dbgPicoSecPerDBU](#)
- [dbgTopCell](#)

dbgCellList

`dbgCellList`

Returns the address of the cell list.

Command Order

You can use this command after importing the design.

Example

```
innovus 20> dbgCellList  
0xd5c6100
```

dbgConvertSetupHoldOnAsyncToRecRem

dbgConvertSetupHoldOnAsyncToRecRem

Controls whether single-edged setup and hold checks on asynchronous pins in the Liberty library are inferred as recovery and removal checks.

Command Order

You can use this command before the design is restored.

Example

```
innovus 32> dbgConvertSetupHoldOnAsyncToRecRem  
0
```

dbgCurRuler

dbgCurRuler

Returns the address of the current active ruler.

Command Order

You can use this command after importing the design.

dbgDBUPerIGU

dbgDBUPerIGU

Returns the number of database units for instance grid.

Command Order

You can use this command after importing the design.

Example

```
innovus 21> dbgDBUPerIGU
```

```
425
```

dbgDBUPerIoGU

dbgDBUPerIoGU

Returns the database units for the I/O grid.

Command Order

You can use this command after importing the design.

Example

```
innovus 22> dbgDBUPerIoGU
```

```
0
```

dbgDBUPerMGrid

dbgDBUPerMGrid

Returns the database units for the manufacturing grid.

Command Order

You can use this command after importing the design.

Example

```
innovus 23> dbgDBUPerMGrid  
1
```

dbgDesignName

dbgDesignName

Returns the design name.

Command Order

You can use this command after importing the design.

Example

```
innovus 24> dbgDesignName  
mif_top
```

dbgHead

dbgHead

Returns the address of the head.

Command Order

You can use this command after importing the design.

Example

```
innovus 26> dbgHead  
0xbfa1e20
```

dbgHierChar

dbgHierChar

Returns the hierarchical character.

Command Order

You can use this command after importing the design.

Example

```
innovus 27> dbgHierChar  
/
```

dbgIsHInstRootX

dbgIsHInstRootX

Reports whether the X tree has been grouped. The command returns 1 if the X tree has been grouped, and 0 if it has not.

Command Order

You can use this command after importing the design.

Example

```
innovus 28> dbgIsHInstRootX  
0
```

dbgLayerArr

dbgLayerArr

Returns address of the layer array.

Command Order

You can use this command after importing the design.

Example

```
innovus 28> dbgLayerArr  
0xdc22980
```

dbgMicronPerDBU

dbgMicronPerDBU

Returns the microns per database unit.

Command Order

You can use this command after importing the design.

Example

```
innovus 30> dbgMicronPerDBU  
0.001
```

dbgNrLayer

dbgNrLayer

Returns the number of layers, in bit format.

Command Order

You can use this command after importing the design.

Example

```
innovus 31> dbgNrLayer  
20
```

dbgPicoFPerDBU

dbgPicoFPerDBU

Returns the number of pico farads per database unit.

Command Order

You can use this command after importing the design.

Example

```
innovus 32> dbgPicoFPerDBU
```

0.001

dbgPicoSecPerDBU

dbgPicoSecPerDBU

Returns the number of pico seconds per database unit.

Command Order

You can use this command after importing the design.

Example

```
innovus 32> dbgPicoSecPerDBU
```

0.1

dbgTopCell

dbgTopCell

Returns the address of the top cell.

Command Order

You can use this command after importing the design.

Example

```
innovus 33> dbgTopCell
```

0xd5c6100

