

ISA Dialog Manager

ATTRIBUTREFERENZ

A.06.03.b

In diesem Handbuch sind alle vordefinierten Attribute der ISA Dialog Manager Objekte beschrieben. Es enthält die Definitionen und Datentypen der Attribute für die Regelsprache und die Programmiersprachen C und COBOL.



ISA Informationssysteme GmbH

Meisenweg 33

70771 Leinfelden-Echterdingen

Deutschland

Microsoft, Windows, Windows 2000 bzw. NT, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10 und Windows 11 sind eingetragene Warenzeichen von Microsoft Corporation.

UNIX, X Window System, OSF/Motif und Motif sind eingetragene Warenzeichen von The Open Group.

HP-UX ist ein eingetragenes Warenzeichen von Hewlett-Packard Development Company, L.P.

Micro Focus, Net Express, Server Express und Visual COBOL sind Warenzeichen oder eingetragene Warenzeichen von Micro Focus International plc und/oder ihrer Tochterunternehmen in den USA, Großbritannien und anderen Ländern.

Qt ist ein eingetragenes Warenzeichen von The Qt Company Ltd. und/oder ihrer Tochterunternehmen.

Eclipse ist ein eingetragenes Warenzeichen von Eclipse Foundation, Inc.

TextPad ist ein eingetragenes Warenzeichen von Helios Software Solutions.

Alle genannten und ggf. durch Dritte geschützten Marken- und Warenzeichen unterliegen uneingeschränkt den Bestimmungen des jeweils gültigen Kennzeichenrechts und den Besitzrechten der jeweiligen eingetragenen Eigentümer. Allein aufgrund der bloßen Nennung ist nicht der Schluss zu ziehen, dass Markenzeichen nicht durch Rechte Dritter geschützt sind.

© 1987 – 2024; ISA Informationssysteme GmbH, Leinfelden-Echterdingen, Deutschland

Darstellungskonventionen

DM wird in diesem Handbuch synonym zu "Dialog Manager" verwendet.

Die Bezeichnung UNIX schließt generell alle unterstützten UNIX-Derivate ein - außer in den explizit angegebenen Fällen.

Dort wo für geläufige englische Fachbegriffe keine gängigen deutschen Übersetzungen existieren, wird zur Vermeidung von Unklarheiten der englische Begriff verwendet.

< > muss durch einen entsprechenden Wert ersetzt werden

color Schlüsselwort ("keyword")

.bgc Attribut

{ } optional (0 oder einmal)

[] optional (0 oder n-mal)

<A> | entweder <A> oder

Beschreibungsmodus

Alle Schlüsselwörter sind fett und unterstrichen, z.B.

variable **integer** **function**

Indizierung von Attributen

Syntax für indizierte Attribute:

[I]

[I,J] bzw. [row,column]

Identifikatoren

Identifikatoren müssen mit einem Großbuchstaben oder einem "Unterstrich" ('_') beginnen. Die weiteren Zeichen können Groß-, Kleinbuchstaben, Zahlen oder Unterstriche sein.

Der Bindestrich ('-') ist für die Benennung von Identifikatoren als Zeichen **nicht** zugelassen!

Die maximale Länge eines Identifikators beträgt 31 Zeichen.

Beschreibung der zugelassenen Identifikatoren in Backus-Naur-Form

<Identifikator> ::= <erstes Zeichen>{<Zeichen>}

<erstes Zeichen>	::=	_ <Großbuchstabe>
<Zeichen>	::=	_ <Kleinbuchstabe> <Großbuchstabe> <Ziffer>
<Ziffer>	::=	1 2 3 ... 9 0
<Kleinbuchstabe>	::=	a b c ... x y z
<Großbuchstabe>	::=	A B C ... X Y Z

Inhalt

Darstellungskonventionen	3
Inhalt	5
1 Zusammenhänge zwischen Attributen	19
1.1 Standardattribute	19
1.2 Hierarchieattribute	19
1.3 Geometrieattribute	20
1.4 Scrollbarattribute	21
1.5 Layoutattribute	23
1.6 Textattribute	24
2 Attribute in alphabetischer Reihenfolge	25
2.1 .acc_label	25
2.2 .acc_text	27
2.3 .accelerator	29
2.4 .action	30
2.5 .active	31
2.6 .active[integer]	33
2.7 .active[index]	34
2.8 .activeitem	35
2.9 .activeobject	37
2.10 .alignment	38
2.11 .allowundefined	40
2.12 .application	41
2.13 .arrows	42
2.14 .attribute	43
2.15 .attribute[integer]	44
2.16 .autoalign	46
2.17 .autosize	47
2.18 .backpage	48
2.19 .barwidth	49
2.20 .bgc	50

2.21 .bgc[index]	51
2.22 .binding	52
2.23 .bordercolor	53
2.24 .borderraster	54
2.25 .borderstyle	59
2.26 .borderwidth	62
2.27 .button[integer]	63
2.28 .bw	64
2.29 .calendaralignment	66
2.30 .canvasfunc	67
2.31 .certificatefile	68
2.32 .changedir	69
2.33 .child[integer]	70
2.34 .childcount	71
2.35 .class	72
2.36 .closeable	73
2.37 .codepage	74
2.38 .colalignment[integer]	77
2.39 .colcount	78
2.40 .colfirst	79
2.41 .colheader	80
2.42 .colheadfgc	81
2.43 .colheadfont	82
2.44 .colheadshadow	83
2.45 .colheadvisible	84
2.46 .collinewidth[integer]	85
2.47 .color	86
2.48 .color_type	87
2.49 .color_type[integer]	88
2.50 .colorcount	89
2.51 .colorcount[integer]	90
2.52 .colorname[integer]	91
2.53 .colsizeable[integer]	92
2.54 .coltitle[integer]	93
2.55 .colvisible[integer]	94
2.56 .colwidth[integer]	95
2.57 .configurable	97

2.58 .connect	98
2.59 .constant	101
2.60 .content	102
2.61 .content[integer]	104
2.62 .content[index]	105
2.63 .contentfunc	107
2.64 .control	108
2.65 .count	109
2.66 .cursor	110
2.67 .cursorname[integer]	111
2.68 .curvalue	112
2.69 .cut_pending	113
2.70 .cut_pending_changed	114
2.71 .data	115
2.72 .dataget[attribute]	116
2.73 .dataindex[attribute]	117
2.74 .datamap[attribute]	118
2.75 .datamodel[attribute]	119
2.76 .dataoptions[enum]	120
2.77 .dataselect[attribute]	122
2.78 .dataselectattr[attribute]	124
2.79 .dataselectcount[attribute]	125
2.80 .dataselecttype[attribute]	126
2.81 .dataset[attribute]	128
2.82 .defbutton	129
2.83 .deltavalue	130
2.84 .depth	131
2.85 .dialog	132
2.86 .dialogbox	133
2.87 .direction	135
2.88 .directory	139
2.89 .display	140
2.90 .doccursor[integer]	141
2.91 .dock_line	142
2.92 .dock_offset	143
2.93 .dockable[enum]	144
2.94 .docking	146

2.95 .document[integer]	147
2.96 .editable	148
2.97 .editable[index]	150
2.98 .editpos	151
2.99 .edittext	152
2.100 .edittype	153
2.101 .endsel	154
2.102 .env[string]	155
2.103 .envvar[string]	156
2.104 .errfile	157
2.105 .errorcode	158
2.106 .event[integer]	159
2.107 .event_code	160
2.108 .eventcount	161
2.109 .exec	162
2.110 .export	164
2.111 .extension	165
2.112 .external	166
2.113 .external[integer]	167
2.114 .extevent	168
2.115 .face	169
2.116 .fgc	170
2.117 .fgc[index]	171
2.118 .field[index]	172
2.119 .fieldactive[index]	173
2.120 .fieldfocus	174
2.121 .fieldfocus[index]	175
2.122 .fieldfocusable	176
2.123 .fieldshadow	177
2.124 .filled	178
2.125 .firstchar	179
2.126 .firstchild	180
2.127 .firstmenu	181
2.128 .firstrecord	182
2.129 .firstsubcontrol	183
2.130 .firsttoolbar	184
2.131 .focus	185

2.132 .focus_on_click	188
2.133 .focusable	189
2.134 .focusitem	190
2.135 .font	191
2.136 .font[index]	192
2.137 .fontname[integer]	193
2.138 .format	194
2.139 .format[index]	197
2.140 .formatfunc	198
2.141 .function	199
2.142 .gradient	199
2.143 .grey	201
2.144 .groupbox	202
2.145 .height	203
2.146 .height[class]	204
2.147 .height[enum]	204
2.148 .help	206
2.149 .helpmenu	207
2.150 .helppos	208
2.151 .hinttext	209
2.152 .hls[enum]	209
2.153 .hsb_arrows	211
2.154 .hsb_linemotion	212
2.155 .hsb_optional	213
2.156 .hsb_pagemotion	214
2.157 .hsb_visible	215
2.158 .icon	216
2.159 .iconic	218
2.160 .iconifyable	219
2.161 .idispach	220
2.162 .ignorecursor	221
2.163 .imagebgc	222
2.164 .imagefgc	223
2.165 .incrtime	224
2.166 .index	225
2.167 .indexscope[attribute]	227
2.168 .input[integer]	228

2.169 .instance[integer]	229
2.170 .interfaceid	230
2.171 .is_applet	231
2.172 .itemcount	232
2.173 .itemorder	233
2.174 .ixmlomdocument2	234
2.175 .ixmlomnode	235
2.176 .ixmlomodelist	236
2.177 .keyboard	237
2.178 .label	238
2.179 .label[anyvalue]	239
2.180 .language	242
2.181 .language[integer]	243
2.182 .lastchild	244
2.183 .lastmenu	245
2.184 .lastrecord	246
2.185 .lastsubcontrol	247
2.186 .lasttoolbar	248
2.187 .layoutbox	249
2.188 .level[integer]	250
2.189 .license_key	252
2.190 .linemotion	253
2.191 .load	254
2.192 .local	255
2.193 .logfile	256
2.194 .majortabheight	257
2.195 .majortabwidth	258
2.196 .mapped	259
2.197 .mapping[integer]	260
2.198 .masterapplication[enum]	261
2.199 .maxchars	265
2.200 .maxchars[index]	266
2.201 .maxheight	267
2.202 .maximized	268
2.203 .maxsize[integer]	269
2.204 .maxvalue	270
2.205 .maxwidth	272

2.206 .member[integer]	273
2.207 .membercount	274
2.208 .menu	275
2.209 .menu[integer]	276
2.210 .menubgc	277
2.211 .menucount	278
2.212 .menufgc	279
2.213 .message[integer]	280
2.214 .mincolwidth	281
2.215 .mincolwidth[integer]	282
2.216 .minheight	283
2.217 .minortabheight	284
2.218 .minortabwidth	285
2.219 .minrowheight	286
2.220 .minsize[integer]	287
2.221 .minvalue	288
2.222 .minwidth	290
2.223 .mode	291
2.224 .model	292
2.225 .module	293
2.226 .mouse_buttons	294
2.227 .mouseover	295
2.228 .moveable	296
2.229 .msgboxtext[enum]	297
2.230 .multiline	299
2.231 .multisel	300
2.232 .mustexist	303
2.233 .name	304
2.234 .navigable	306
2.235 .navigation	308
2.236 .nextactive[integer]	309
2.237 .nextactive[index]	310
2.238 .nodetype	311
2.239 .notepage	312
2.240 .open[integer]	313
2.241 .opsys_string	314
2.242 .opsys_type	315

2.243 .options[enum]	316
2.243.1 Motif-Option .options[opt_scroll_on_focus]	337
2.243.2 Optionen für Gruppierungsobjekte unter Windows	338
2.243.3 Optionen für die Rasterberechnung unter Windows	339
2.244 .order	340
2.245 .output[integer]	342
2.246 .overridecursor	343
2.247 .pagemotion	345
2.248 .parent	346
2.249 .password	347
2.250 .path	348
2.251 .pattern	349
2.252 .picheight	350
2.253 .picture	352
2.254 .picture[enum]	353
2.255 .picture[integer]	357
2.256 .picture_hilite[integer]	359
2.257 .picwidth	360
2.258 .pointer_height	362
2.259 .pointer_height[integer]	363
2.260 .pointer_width	364
2.261 .pointer_width[integer]	365
2.262 .posraster	366
2.263 .preedit	367
2.264 .preeditssel	368
2.265 .privatekeyfile	369
2.266 .propscale	370
2.267 .publicid	371
2.268 .publickeyfile	372
2.269 .real_height	373
2.270 .real_modified	374
2.271 .real_screen	375
2.272 .real_sensitive	376
2.273 .real_shadowobject	377
2.274 .real_size[integer]	378
2.275 .real_version[enum]	379
2.276 .real_visible	380
2.277 .real_width	381

2.278 .real_x	382
2.279 .real_xraster	383
2.280 .real_y	384
2.281 .real_yraster	385
2.282 .record[integer]	386
2.283 .recordcount	387
2.284 reexport	388
2.285 .reffont	391
2.286 .refstring	393
2.287 .rgb[enum]	393
2.288 .root	395
2.289 .rowalignment[integer]	396
2.290 .rowcount	397
2.291 .rowfirst	398
2.292 .rowheader	399
2.293 .rowheadfgc	400
2.294 .rowheadfont	401
2.295 .rowheadshadow	402
2.296 .rowheadvisible	403
2.297 .rowheight[integer]	404
2.298 .rowlinewidth[integer]	405
2.299 .rowsizeable[integer]	406
2.300 .rowvisible[integer]	407
2.301 .scale	408
2.302 .scalestyle	410
2.303 .scope	412
2.304 .scope[attribute]	413
2.305 .screen	414
2.306 .screen[integer]	415
2.307 .screen_height	416
2.308 .screen_height[integer]	417
2.309 .screen_width	418
2.310 .screen_width[integer]	419
2.311 .screen_x	420
2.312 .screen_x[integer]	421
2.313 .screen_y	422
2.314 .screen_y[integer]	423

2.315 .screencount	424
2.316 .searchpath	425
2.317 .selected[integer]	426
2.318 .selection[enum]	427
2.319 .self	428
2.320 .selstyle	429
2.321 .sensitive	431
2.322 .sensitive[integer]	433
2.323 .sensitive[index]	434
2.324 .shadowattr	435
2.325 .shadowindex	436
2.326 .shadowinstance	437
2.327 .shadowobject	439
2.328 .shortdaynames	440
2.329 .showitem	441
2.330 .size	441
2.331 .size[integer]	442
2.332 .sizeable	443
2.333 .sizeable[class]	444
2.334 .sizeraster	445
2.335 .smallpicheight	446
2.336 .smallpicture[integer]	447
2.337 .smallpicwidth	448
2.338 .source	449
2.339 .spacing	450
2.340 .specified	451
2.341 .startsel	452
2.341.1 edittext, poptext	452
2.341.2 filereq	453
2.342 .starttime	454
2.343 .state	455
2.344 .static	456
2.345 .statusbar	457
2.346 .statushelp	458
2.347 .style	459
2.347.1 checkbox, image, listview, menuitem und menusep	459
2.347.2 datetime, poptext und toolbar	462
2.347.3 filereq (Filerequestor)	464

2.347.4 font	465
2.347.5 spinbox	465
2.348 .style[enum]	467
2.349 .subcontrol[integer]	469
2.350 .subcontrolcount	470
2.351 .sysmodal	471
2.352 .systemerror	472
2.353 .systemid	473
2.354 .tabalignment	474
2.355 .tabshape	475
2.356 .tabtype	476
2.357 .target	477
2.358 .terminal	478
2.359 .terminaltype	479
2.360 .text	480
2.361 .text[enum]	481
2.362 .text[integer]	483
2.363 .textbgc	484
2.364 .textfgc	485
2.365 .textwidth	486
2.366 .tile	487
2.367 .tiledpi	488
2.368 .tilestyle	489
2.369 .timeout	491
2.370 .title	492
2.371 .titlebar	493
2.372 .titlebgc	494
2.373 .titlefgc	495
2.374 .today	496
2.375 .todaymarker	497
2.376 .toolbar	498
2.377 .toolbar[integer]	499
2.378 .toolbarcount	500
2.379 .toolhelp	501
2.380 .toolkit	503
2.381 .toolkit_string	504
2.382 .toolkit_version	505
2.383 .top_most	506

2.384 .topitem	507
2.385 .tracefile	508
2.386 .tracetime	509
2.387 .tracing	511
2.388 .trailingdates	512
2.389 .transformer[integer]	513
2.390 .transport	514
2.391 .type	515
2.392 .typescope	516
2.393 .typescope[integer]	517
2.394 .userdata	518
2.395 .userdata[integer]	519
2.396 .userdata[index]	520
2.397 .username	521
2.398 .userplaced	522
2.399 .uuid	523
2.400 .value	524
2.400.1 datetime	524
2.400.2 doccursor (XML-Cursor)	526
2.400.3 filereq (Datei- und Verzeichnisdialoge)	526
2.400.4 Globale Variablen	526
2.401 .value[integer]	527
2.402 .version[enum]	528
2.403 .version_string	530
2.404 .vheight	531
2.405 .visible	532
2.406 .vsb_arrows	533
2.407 .vsb_linemotion	534
2.408 .vsb_optional	535
2.409 .vsb_pagemotion	536
2.410 .vsb_visible	537
2.411 .vscreen_height	538
2.412 .vscreen_width	539
2.413 .vscreen_x	540
2.414 .vscreen_y	541
2.415 .vwidth	542
2.416 .weeknumbers	543
2.417 .weight	544

2.418 .width	545
2.419 .width[class]	546
2.420 .width[enum]	546
2.421 .window	548
2.422 .winsys	549
2.423 .winsys_string	550
2.424 .winsys_version	551
2.425 .wrap	552
2.426 .x	553
2.427 .xalignment[index]	554
2.428 .xauto	555
2.429 .xdpi	556
2.430 .xdpi[integer]	557
2.431 .xleft	558
2.432 .xmargin	559
2.433 .xml	561
2.434 .xorigin	562
2.435 .xraster	563
2.436 .xright	565
2.437 .xspacing	566
2.438 .y	567
2.439 .yalignment[index]	568
2.440 .yauto	569
2.441 .ybottom	570
2.442 .ydpi	571
2.443 .ydpi[integer]	572
2.444 .ymargin	573
2.445 .yorigin	575
2.446 .yraster	576
2.447 .yspacing	578
2.448 .ytop	579
Index	581

1 Zusammenhänge zwischen Attributen

1.1 Standardattribute

Die Attribute *.visible* und *.sensitive* geben die Sichtbarkeit bzw. die Erreichbarkeit eines Objektes aus Benutzersicht an. Da die tatsächliche Sichtbarkeit/Erreichbarkeit eines Objekts aber auch vom jeweiligen Vaterobjekt abhängig ist kann über die Attribute *.real_visible* und *.real_sensitive* die momentane, tatsächliche Sichtbarkeit/Erreichbarkeit eines Objekts erfragt werden. Dies kann insbesondere bei Setzen des Fokus über *.focus* sinnvoll sein, da die Setzung des Fokus auf unsichtbare bzw. nicht erreichbare Objekte zu Fehlermeldungen führt.

Als Besonderheit ist noch das Attribut *.mapped* verfügbar. Es bestimmt, ob ein Objekt schon sichtbar gezeichnet wird oder nicht, wenn *.visible* dieses Objekts bereits auf *true* gesetzt wird. Ist *.mapped false* und *.visible true*, sind bei diesem Objekt auch im unsichtbaren Zustand schon Operationen zulässig, die ansonsten nur an einem sichtbaren Objekt ausgeführt werden könnten.

Um in einer Statusbar eines Fensters beim Verweilen mit dem Mauscursor auf einem Objekt einen Hilfetext anzuzeigen wird dieser im Attribut *.statushelp* am jeweiligen Objekt angegeben. Selbiges gilt für die aufpoppenden Hilfstexte (Tooltip) welche im Attribut *.toolhelp* angegeben werden. Zusätzlich können im Attribut *.help* noch beliebige Hilfetexte angegeben werden, wobei hier der Programmierer selbst für die Anzeige verantwortlich ist.

Zur Verwendung von Tastaturkürzeln, die beispielsweise an einem Objekt ein *select*-Ereignis auslösen sollen, wird die jeweilige Tastenkombination zuerst als Ressource definiert und danach dem Objekt über das Attribut *.accelerator* zugewiesen.

1.2 Hierarchieattribute

Hierarchieattribute dienen zum indirekten Zugriff auf Objekte innerhalb der Vater-Kind-Hierarchie.

Während die Attribute *.groupbox*, *.notepage*, *.control*, *.window*, *.toolbar* und *.dialog* das jeweils nächste Objekt der entsprechenden Klasse aufwärts vom aktuellen Objekt betrachtet referenzieren, bezeichnet *.parent* das direkte Vaterobjekt.

Auch kann innerhalb der Kindobjekte mit *.child[integer]* ein direktes Kind eines Objekts referenziert werden, wobei *.childcount* die Gesamtzahl der direkten Kinder angibt.

Eine Sonderstellung nehmen Menüs, Records und Subcontrols ein die, entsprechend zu den direkten Kindern, mit den Attributen *.menucount* und *.menu[integer]*, *.recordcount* und *.record[integer]*, *.sub-*

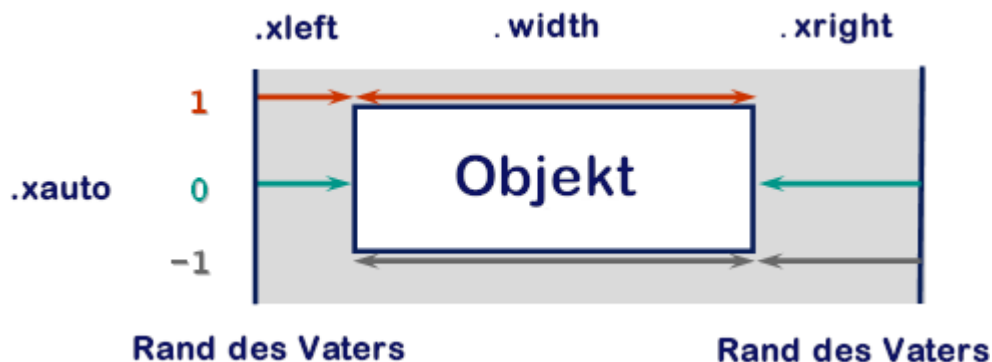
`controlcount` und `.subcontrol[integer]` referenziert werden können. Diese Objektklassen werden über `.child[integer]` nicht referenziert!

1.3 Geometrieattribute

Berechnung der Position eines Objekts

Zur Berechnung der Position eines Objekts sind die Attribute `.xauto`, `.xleft`, `.width` und `.xright` bzw. `.yauto`, `.ytop`, `.height` und `.ybottom` notwendig. Die Ausrichtung wird dabei immer relativ zum Vaterobjekt definiert. Es werden jeweils zwei Positionswerte angegeben und der dritte berechnet. Dabei steuern die Attribute `.xauto` bzw. `.yauto` welche Werte angegeben werden und welcher Wert berechnet wird.

Die Grafik illustriert dies beispielhaft für die X-Richtung, für die Y-Richtung gilt Entsprechendes.



Die Attribute `.posraster` und `.sizeraster` geben hierbei an, ob die gesetzten Werte Raster- oder Pixelwerten entsprechen. Sind eins oder beide Attribute auf `true` gesetzt, so wird das zugrundeliegende Raster über die Attribute `.xraster`, `.yraster` und `.reffont` bestimmt.

Zusätzliche Geometrieattribute

Fenster und Toolbars haben die zusätzlichen Geometrieattribute `.minheight`, `.minwidth`, `.maxheight` und `.maxwidth`. Mit ihnen können die Grenzen definiert werden, in denen die Größe des Objekts interaktiv (durch Ziehen mit der Maus) geändert werden kann.

Objekte, deren Inhalt größer sein kann, als die zur Verfügung stehende Anzeigefläche, haben zusätzliche Geometrieattribute mit denen ihre virtuelle Größe und der sichtbare Ausschnitt des Inhalts definiert werden. Die Attribute `.vheight` und `.vwidth` legen fest, welche virtuelle Höhe und Breite ein Objekt haben soll. Um die nicht sichtbaren Teile anzuzeigen, muss ein Benutzer in dem Objekt scrollen. Dabei wird der sichtbare Ausschnitt über die Attribute `.xorigin` und `.yorigin` gesteuert. Sie definieren die relative Verschiebung des Objektnullpunkts, das heißt die Lage der sichtbaren Fläche auf der größeren virtuellen Fläche des Objekts. Normalerweise liegt der Objektnullpunkt in der linken oberen Ecke des Objekts.

Über die Attribute *.xraster* und *.yraster* werden die horizontalen und vertikalen Einheiten definiert, in denen die Position und Dimension der Kinder angegeben werden. Die Koordinaten der Kinder in Pixeln erhält man, indem man die angegebenen Koordinaten (in Rastereinheiten) mit den Rasterfaktoren *.xraster* und *.yraster* multipliziert.

Mit dem Attribut *.reffont* kann die Angabe der Rastereinheiten an den verwendeten Zeichensatz geknüpft werden. Der DM berechnet dann die Werte *.xraster* und *.yraster* selbst.

Randlose Geometrie

Um das Geometrieverhalten von Objekten mit Rahmen konfigurierbar zu machen, gibt es das Attribut *.borderraster*. Mit ihm kann festgelegt werden, dass die Geometrie des Objekts so berechnet wird, als ob es keinen Rahmen hätte.

Geometrieattribute für hochauflösende Bildschirme

Pixelbasierte Werte von Geometrieattributen werden vom IDM passend zum Skalierungsfaktor des Systems transformiert um eine zu den Systemeinstellungen konsistente Darstellung zu gewährleisten. Die Geometrieeinheiten werden dabei wie bisher in **IDM-Pixelwerten** oder Raster gesetzt.

Die **IDM-Pixelwerte** werden intern entsprechend dem Skalierungsfaktor in **reale Pixelwerte** hochgerechnet und umgesetzt. Diese Konvertierung geschieht in beide Richtungen.

Rasterwerte sind entweder an den verwendeten (HighDPI-fähigen) Zeichensatz geknüpft oder werden ebenfalls in **IDM-Pixelwerten** gesetzt und dann skaliert. Die Rastereinheiten können basierend auf der zugrundeliegenden Schriftart ebenfalls zu **IDM-Pixelwerten** ermittelt werden.

Das Attribut *.propscale* steuert, ob das horizontale und vertikale Raster proportional auf den maximalen Wert gesetzt werden soll, welcher durch die Werteberechnung von *xraster* und *yraster* des Font-Rasters ermittelt wird. Die genaue Berechnung des Raster ist im Kapitel „Berechnung der Rastergröße aus einem Referenzfont“ *Calculating the Grid Size from a Reference Font* der font-Resource aufgeführt.

Siehe auch Kapitel „HighDPI Unterstützung“ im Handbuch „Programmiertechniken“

1.4 Scrollbarattribute

Einige Objekte, z.B. Fenster, Groupbox und Notepage, können Scrollbars (Schieberegler, Bildlaufleisten) besitzen, mit denen die angezeigten Inhalte des Objekts horizontal und vertikal verschoben werden können. Mit den Scrollbarattributen werden die Anzeige und das Verhalten der Scrollbars definiert.

Es gibt folgende Scrollbarattribute:

- » *.hsb_linemotion*
- » *.hsb_optional*
- » *.hsb_pagemotion*
- » *.hsb_visible*
- » *.vsb_linemotion*

- » *.vsb_optional*
- » *.vsb_pagemotion*
- » *.vsb_visible*

Die mit *.hsb_* beginnenden Attribute beziehen sich auf die horizontale Scrollbar und die mit *.vsb_* beginnenden Attribute auf die vertikale Scrollbar.

Die Sichtbarkeit von Scrollbars unterliegt dem Einfluss folgender Attribute:

- » *.hsb/vsb_optional*
- » *.hsb/vsb_visible*

und

- » *.width/height*
- » *.vwidth/vheight*

Dabei gelten folgende Regeln:

- » Ein Objekt kann nur dann eine horizontale Scrollbar haben, wenn seine virtuelle Breite *.vwidth* gesetzt und >0 ist.
- » Ein Objekt kann nur dann eine vertikale Scrollbar haben, wenn seine virtuelle Höhe *.vheight* gesetzt und >0 ist.

Wenn keine Scrollbars gesetzt sind, wird die virtuelle Größe ignoriert.

Die Attribute *.hsb/vsb_visible* und *.hsb/vsb_optional* kontrollieren, ob die Scrollbars ständig sichtbar sein sollen oder nur, wenn sie auch benötigt werden.

Wie oben erwähnt, erfordern Scrollbars das Setzen der virtuellen Größe, d.h. wenn keine virtuelle Größe gesetzt ist, sind Scrollbars nicht verfügbar.

Wenn die Scrollbars sichtbar sind, und

- » *.hsb/vsb_optional = true*
=> wird die Scrollbar nur sichtbar, wenn sie wirklich benötigt wird.
- » *.hsb/vsb_optional = false*
=> ist die Scrollbar immer sichtbar.

Anmerkung

In der Motif-Version wird *.hsb/vsb_optional* ignoriert, d.h. intern wird *.hsb/vsb_optional* implizit immer auf true gesetzt.

Um die Stellung des Scrollbarsliders zu erfragen bzw. zu ändern, müssen die Attribute *.xorigin* und *.yorigin* verwendet werden. Diese beiden Attribute geben die Verschiebung des eigentlichen Objektnullpunktes an.

Die Attribute *.hsb/vsb_linemotion* und *.hsb/vsb_pagemotion* definieren, um wie viele Einheiten *.xorigin* bzw. *.yorigin* – und damit der Objekthalt – bei der jeweiligen Scrollaktion verschoben werden

soll. Einheiten sind entweder Rastereinheiten oder Pixel, je nachdem ob beim Objekt `.sizeraster` gesetzt ist oder nicht. Der Wert `0` bedeutet, dass der Standardwert des Systems genommen wird. Bei `.hsb/vsb_pagemotion` wird dann seitenweise gescrollt.

Es gibt Objekte, z.B. die Listbox, die zwar Scrollbars anzeigen, wenn ihr Inhalt nicht in die vorhandene Anzeigefläche passt, die aber nicht über Scrollbarattribute verfügen. Bei diesen Objekten werden die Scrollbars vom Fenstersystem gesteuert und können vom ISA Dialog Manager nicht beeinflusst werden.

1.5 Layoutattribute

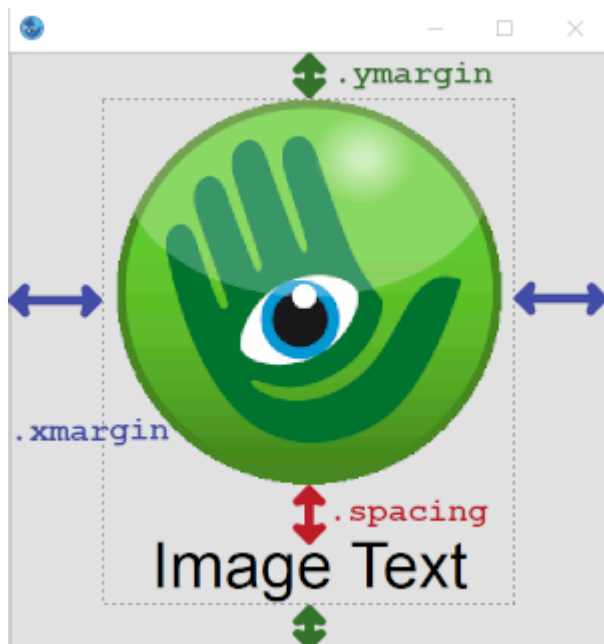
Die Hintergrundfarbe eines Objekts wird über die Zuweisung einer Farbressource am Attribut `.bgc` bestimmt. Für die Vordergrundfarbe, d.h. meist die Farbe des Textes innerhalb eines Objekts, erfolgt die Angabe der Farbressource im Attribut `.fgc`.

Die Art der Schriftdarstellung erfolgt durch die Zuweisung einer Fontressource am `.font` Attribut des jeweiligen Objekts.

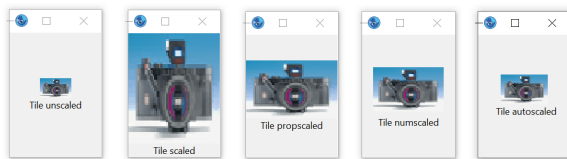
Objekte, die Bilder darstellen können, bekommen diese im `.picture` Attribut als Bildressource zugewiesen. Dabei ist dieses Attribut mit Integerwerten indiziert, wenn es sich um ein Listenobjekt (z.B. Treeview) handelt.

Layoutattribute am Image Objekt

Beim Image-Objekt bestimmen die Attribute `.xmargin` und `.ymargin` die Abstände zwischen Image-Rand und Darstellungsbereich. Das Attribut `.spacing` dagegen definiert den Abstand zwischen Tile und Text innerhalb des Darstellungsbereichs des Image. Die folgende Grafik zeigt das Zusammenspiel dieser Attribute:



Über das Attribut `.scalestyle` kann darüber hinaus die Art der Skalierung des Tiles innerhalb des Darstellungsbereichs festgelegt werden. Die folgende Grafik zeigt die verfügbaren Skalierungsstile:



Layoutattribute für hochauflösende Bildschirme

Über das Attribut `.tiledpi` am Setup-Objekt kann angegeben werden für welche DPI-Auflösung die Tiles der Anwendung entworfen wurden. Die Größe eines Bildes oder Musters wird dann ausgehend von diesem Wert auf den aktuell geltenden DPI-Wert umgerechnet.

Siehe auch Kapitel „HighDPI UnterstützungSupport“ im Handbuch „Programmiertechniken“

1.6 Textattribute

Die Titelzeile eines Fensters und der Titel einer Menübox werden im Attribut `.title` festgelegt. Für alle anderen darzustellenden Texte werden die Attribute `.text` und `.content` benutzt, wobei ein Objekt normalerweise nur eines dieser Attribute benutzt. So gilt das Attribut `.content` nur für den Edittext und die Listenobjekte, für letztere indiziert.

Eine **Sonderstellung** nimmt das Objekt **Poptext** / Combobox ein. Hier wird auf den Listeninhalt über `.text[integer]` zugegriffen; gleichzeitig wird auch der lesende Zugriff auf den momentan angezeigten Text mit `.content` unterstützt. Da der schreibende Zugriff auf `.content` vom Wert im Attribut `.style` (poptext, edittext, listbox) am Poptext abhängt, ist hier die Attribut- und Objektbeschreibung zu beachten.

2 Attribute in alphabetischer Reihenfolge

2.1 .acc_label

Mit diesem Attribut kann der Automation-Identifikator überschrieben werden, der für ein Oberflächenobjekt über die MICROSOFT UIA-Schnittstelle vom IDM erfragt wird. Bei einem leeren String wird der Automation-Identifikator normalerweise vom Windows Control bzw. durch die UIA-Unterstützung im IDM sinnvoll vorgegeben.

Definition

<i>Datentyp</i> string, object [text]	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_acc_label Datentyp: DT_string, DT_text		<i>COBOL</i> Bezeichner: AT-acc-label Datentyp: DT-string, DT-text
<i>Vererbung</i> ja	<i>Standardwert</i> ""	
<i>Klassifizierung</i> Standardattribut		

Unterstützung des Attributs durch Objekte

Objekt	Unterstützung des Attributs
<i>filereq, messagebox</i>	Attribut hat keine Auswirkung
<i>menubox, menuitem, menusep</i>	

Objekt	Unterstützung des Attributs
<i>canvas, spinbox, statusbar, tablefield</i>	Attribut wird unterstützt
<i>groupbox, notebook, notepage, splitbox</i>	
<i>image, layoutbox, window</i>	
<i>rectangle, scrollbar</i>	
<i>checkbox, pushbutton, radiobutton</i>	
<i>edittext, poptext, statictext</i>	
<i>control, listbox, treeview</i>	
andere Objektklassen	Attribut wird nicht unterstützt

Anmerkung

Dieses Attribut ist nur für die automatisierte Fremdsteuerung mit aktiver MICROSOFT UIA-Unterstützung relevant. Unter QT und MOTIF hat dieses Attribut keine Funktion.

Bei einer Überschreibung sollten die Regeln beachtet werden, welche in der MICROSOFT UI Automation Dokumentation für die AutomationId vorgegeben sind.

Verfügbarkeit

Ab IDM-Version A.06.01.e

Siehe auch

Kapitel „UIA Objektidentifikation“ im Handbuch „Automatisches Testen und Barrierefreiheit“

2.2 .acc_text

Mit diesem Attribut kann der Automation-Name überschrieben werden, der für ein Oberflächenobjekt über die MICROSOFT UIA-Schnittstelle vom IDM erfragt wird. Bei einem Wert von *null* wird der Name normalerweise vom Windows Control bzw. durch die UIA-Unterstützung im IDM sinnvoll vorgegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [text], string	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_acc_text	Bezeichner: AT-acc-text
Datentyp: DT_text, DT_string	Datentyp: DT-text, DT-string

<i>Vererbung</i>	<i>Standardwert</i>
ja	null

Klassifizierung
Standardattribut

Unterstützung des Attributs durch Objekte

Objekt	Unterstützung des Attributs
<i>filereq, messagebox</i>	Attribut hat keine Auswirkung
<i>menubox, menuitem, menusep</i>	
<i>canvas, spinbox, statusbar, tablefield</i>	Attribut wird unterstützt
<i>groupbox, notebook, notepage, splitbox</i>	
<i>image, layoutbox, window</i>	
<i>rectangle, scrollbar</i>	
<i>checkbox, pushbutton, radiobutton</i>	
<i>edittext, poptext, statictext</i>	
<i>control, listbox, treeview</i>	
andere Objektklassen	Attribut wird nicht unterstützt

Anmerkung

Dieses Attribut ist nur für die automatisierte Fremdsteuerung mit aktiver MICROSOFT UIA-Unterstützung relevant. Unter QT und MOTIF hat dieses Attribut keine Funktion.

Verfügbarkeit

Ab IDM-Version A.06.01.e

Siehe auch

Kapitel „UIA Objektidentifikation“ im Handbuch „Automatisches Testen und Barrierefreiheit“

2.3 .accelerator

Mit Hilfe dieses Attributs wird auf der Tastatur eine äquivalente Taste definiert, über die man dann ein entsprechendes Objekt selektieren kann. Der Accelerator muss dabei zuerst als Ressource definiert werden.

Definition

<i>Datentyp</i> object [accelerator]	<i>Zugriff</i> get, set get (thisevent)	<i>changed-Ereignis</i> ja
--	--	-------------------------------

C
Bezeichner: AT_accelerator
Datentyp: DT_accel

COBOL
Bezeichner: AT-accelerator
Datentyp: DT-accel

Klassifizierung
Standardattribut

Beim Objekt **Notepage** kann das Attribut dazu verwendet werden, um eine bestimmte Notepage zu aktivieren, nach oben zu bringen. Hierbei wird ein select-activate-Ereignis ausgelöst.

Bei **thisevent** kann mit diesem Attribut der Accelerator-Identifikator erfragt werden (gültig beim *key*-Ereignis).

2.4 .action

Mit diesem Attribut werden die möglichen Drag&Drop-Aktionen der Ressourcen **source** und **target** bestimmt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_action	Bezeichner: AT-action
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	source, target

Wertebereich

action_cut

„Ausschneiden“, löscht das Element an seinem aktuellen Platz.

action_copy

„Kopieren“, behält das Element an seinem aktuellen Platz.

action_paste

„Einfügen“ des ausgeschnittenen oder kopierten Elements (nur **target**).

2.5 .active

Dieses Attribut legt fest, ob das Objekt aktiv, d.h. für die Eingabe bereit ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_active	Bezeichner: AT-active
Datentyp: DT_boolean	Datentyp: DT-boolean

Klassifizierung
Standardattribut

Application

Mit Hilfe dieses Attributes kann beim **Application**-Objekt definiert und abgefragt werden, ob die Anwendung gerade läuft. Eine Änderung dieses Attributes von *false* auf *true* bewirkt, dass die Anwendung gestartet wird. Wird dieses Attribut von *true* auf *false* geändert, so wird die Anwendung wieder beendet.

Edittext

Das Attribut .active definiert, ob der editierbare Text einen aktuellen Fokus besitzt.

Notepage

Bei einer **Notepage** ist .active = true für die Seite, die aktuell die oberste Notepage ist, für alle anderen Notepages ist .active = false. D.h. dieses Attribut kann immer nur für eine einzige Notepage den Wert true besitzen.

Der Notepage-Label dieser Notepage ist im Attribut .activeobject des Notebooks gespeichert.

Timer

Mit Hilfe dieses Attributes kann definiert und abgefragt werden, ob der **Timer** gerade läuft. Eine Änderung dieses Attributes von false auf true bewirkt, dass der Timer gestartet wird. Wird dieses Attribut von true auf false geändert, wird der Timer wieder zurückgesetzt und ist nicht weiter aktiv.

Image

Bei einem image wird hier bei [.style](#) = checkbox bestimmt, ob das Objekt aktiv („gedrückt“) oder inaktiv dargestellt wird bzw. es kann ausgelesen werden, welchen Zustand das Objekt gerade besitzt. Im Zusammenspiel mit .picture[enum] kann die jeweils angezeigte Tile-Ressource bestimmt werden.

Bei .style = *menubox* ist das Objekt bei geöffnetem Kontextmenü aktiv.

Menuitem

Bei einem menuitem mit `.style = checkbox` wird mit dem Attribut `active` bestimmt, ob der Menüeintrag aktiviert (Haken ist sichtbar) ist oder nicht. Im `.style = radiobutton` ist nur das Setzen des Wertes `true` für aktiviert erlaubt; ausgelesen werden können aber beide Zustände (aktiviert bzw. nicht aktiviert).

Tablefield

Für die Aktivierung des Eingabemodus durch die Anwendung, wird beim Objekt **Tablefield** das Attribut indiziert (`.active[index]`) und nicht-indiziert (`.active`) zur Verfügung gestellt. Für das Setzen des nicht-indizierten Attributs `.active` ergibt sich folgende Verhaltensweise:

Aktion	aktueller Zustand	Reaktion
<code>active := true</code>	Fokus auf sensitivem Feld	Eingabemodus wird eingeschaltet, falls dies noch nicht der Fall ist. Ist der Eingabemodus bereits aktiv, passiert nichts.
<code>active := false</code>	Fokus auf sensitivem Feld	Eingabemodus wird ausgeschaltet, falls er aktiv ist. Ansonsten passiert nichts
<code>active := true / false</code>	Fokus auf nicht sensitivem Feld	Keine Reaktion durch das Objekt. <code>.active</code> wird vom Dialog Manager auf <code>false</code> zurückgesetzt
<code>active := true / false</code>	Fokus auf keinem Feld im Tablefield sondern in der Ecke oder auf einer der Scrollbars	Keine Reaktion durch das Objekt. <code>.active</code> wird auf <code>false</code> vom Dialog Manager zurückgesetzt
<code>active := true</code>	Tablefield hat nicht den Fokus	<code>.active</code> wird vom Dialog Manager wieder auf <code>false</code> gesetzt. Ansonsten keine Reaktion
Benutzer aktiviert Eingabe		<code>.active</code> wird auf <code>true</code> gesetzt

Siehe auch

Attribut `.activeitem`

2.6 .active[integer]

Mit Hilfe dieses Attributes kann jeder einzelne Eintrag in einer **listbox** mit *.multisel = true* auf aktiv bzw. nicht aktiv gesetzt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_active	Bezeichner: AT-active
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox

Siehe auch

Attribut *.activeitem*

2.7 .active[index]

Mit Hilfe dieses Attributes kann jedes einzelne Textfeld im **tablefield** ([row, column]) inklusive der Überschriften und Zeilenköpfe auf aktiv bzw. nicht aktiv gesetzt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_active	Bezeichner: AT-active
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Siehe auch

Attribute *.activeitem*, *.nextactive[index]*

2.8 .activeitem

Der Index des aktuell angezeigten Textes kann abgefragt und im Zusammenhang mit dem Attribut `.activeitem` eingestellt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer (listbox, poptext, spinbox)</i>	get, set	ja
<i>index (tablefield)</i>		

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_activeitem	Bezeichner: AT-activeitem
Datentyp: DT_integer, DT_index	Datentyp: DT-integer, DT-index

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, poptext, spinbox, tablefield

Ein Setzen des Attributs ändert in Listenobjekten auch den angezeigten Text; bei einer **Spinbox** wird hier der angezeigte Text im Kindobjekt der Spinbox geändert.

Beim poptext sind hier auch die Objektbeschreibung und die Hinweise zum Attribut `.content` zu beachten.

Bei Listenobjekten und **Poptext** kommt es zu einer Fehlermeldung, wenn versucht wird, `.activeitem` auf einen nicht vorhandenen Texteintrag zu setzen. Bei der statischen Objektdefinition muss deshalb die Definition des Texteintrags im Dialogskript vor der Anweisung stehen, mit der `.activeitem` gesetzt wird.

Die Bedeutung von .active und .activeitem bei Listbox und Tablefield

`.activeitem` greift auf den selektierten Eintrag bei Objekten mit Einfach-Selektion (`.multisel = false`) zu.

1. Die Einstellung von Werten: `Object.attrib := value`

	Einfach-Selektion	Mehrfach-Selektion
boolean active[i]	nicht erlaubt	Eintrag i
integer activeitem	Index des aktiven Eintrags	nicht erlaubt

2. Das Abfragen von Werten: `? := Object.attrib`

	Einfach-Selektion	Mehrfach-Selektion
boolean active[i]	Eintrag i in selektiertem Zustand	Eintrag i in selektiertem Zustand
integer activeitem	Index des aktiven Eintrages	nicht erlaubt

Anmerkung

Soll bei einer **Listbox** bzw. einem **Tablefield** kein Eintrag selektiert sein, müssen Sie `.activeitem = 0` setzen.

Siehe auch

Attribute `.active`, `.multisel`

2.9 .activeobject

Die aktive, d.h. die aktuell oben liegende **notepage**.

Mögliche Werte sind die Notepage-Labels des Notebooks. Der Defaultwert 0 gilt nur solange das Notebook noch keine Notepage besitzt, danach bezeichnet es die aktive Notepage.

Definition

<i>Datentyp</i> <i>object</i>	<i>Zugriff</i> get	<i>changed-Ereignis</i> ja
----------------------------------	-----------------------	-------------------------------

<i>C</i> Bezeichner: AT_activeobject Datentyp: DT_instance	<i>COBOL</i> Bezeichner: AT-activeobject Datentyp: DT-instance
--	--

<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> notebook
---	----------------------------

Siehe auch

Objekt notebook

2.10 .alignment

Mit Hilfe dieses Attributes definieren Sie die Ausrichtung von Texten.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_alignment Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-alignment Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> edittext, image, notebook, statictext	

Wertebereich

-1	rechtsbündig
0	zentriert
1	linksbündig

image

Das Attribut bestimmt beim **image**-Objekt wie der Text im für ihn verfügbaren Raum horizontal angebunden wird (Standardwert 0).

Die Wirkung von *.alignment* in Abhängigkeit von den Attributen *.spacing* und *.tilestyle* ist beim Attribut *.tilestyle* beschrieben.

notebook

Beim **notebook** definiert dieses Attribut die Ausrichtung des Statustextes, den jede **notepage** besitzen kann (Standardwert 1).

edittext

Das Attribut wird nur unter MICROSOFT WINDOWS und QT unterstützt. Dabei ist folgendes zu beachten:

- » Zeilen, deren Länge die Breite des **edittextes** übersteigt, werden bei *.hsb_visible = false* automatisch umgebrochen.

WINDOWS-spezifisch:

- » Wenn beide Attribute *.multiline* und *.hsb_visible* den Wert *true* haben, wird *.alignment* nicht unterstützt.

- » Bei RTF-Edittexten (*.options[opt_rtf] = true*) bestimmt nicht das Attribut *.alignment* die Textausrichtung, sondern die RTF-Formatierung.

QT-spezifisch:

- » Bei HTML-Edittexten (*.options[opt_rtf] = true*) bestimmt nicht das Attribut *.alignment* die Textausrichtung, sondern die HTML-Formatierung.

statictext

Der Text ist nach oben ausgerichtet, um kompatibel zu einem insensitive statischen Text zu sein.

2.11 .allowundefined

Dieses Attribut bestimmt, ob der Anwender den Wert des Objekts auf „unbestimmt“ setzen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_allowundefined	Bezeichner: AT-allowundefined
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Standardwert</i>	<i>Vererbung</i>
<i>false</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	datetime

Wenn das Attribut den Wert *true* besitzt, wird im **datetime** eine Checkbox angezeigt. Ausschalten der Checkbox setzt den Wert auf „unbestimmt“. Das Attribut *.value* hat dann den Wert "".

Bei *.allowundefined = true* führen ungültige Werte von *.value* dazu, dass die Checkbox eingeschaltet wird.

Wertebereich

true

Der Wert von *.value* kann vom Anwender auf „unbestimmt“ gesetzt werden. Dafür wird im **datetime** eine Checkbox angezeigt.

false

Der Anwender kann *.value* **nicht** auf „unbestimmt“ ("") setzen.

Hinweis

Das Ändern *.allowundefined* im sichtbaren Zustand kann zu einem Zurücksetzen des Objekts führen.

2.12 .application

Dieses Attribut des Objekts **import** definiert die Applikation, an die die Funktionen gebunden werden sollen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [application]	get, set	ja

C
Bezeichner: AT_application
Datentyp: DT_application

COBOL
Bezeichner: AT-application
Datentyp: DT-application

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	import

Siehe auch

Attribut *.masterapplication[enum]*

Kapitel „Modularisierung“ im Handbuch „Programmiertechniken“

2.13 .arrows

Das Attribut *.arrows* definiert, ob die Scrollbar zwei Pfeile besitzt (*true*) oder nicht (*false*).

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_arrows	Bezeichner: AT-arrows
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	scrollbar

Bei *.arrows = false* wird ein Schieberegler anstatt einer Scrollbar verwendet.

Unter MOTIF werden *.curvalue* angezeigt (digitale Anzeige) und *.linemotion* ignoriert, wenn *.arrows = false* ist.

Unter MICROSOFT WINDOWS verwenden **Scrollbar**-Objekte mit *.arrows = false* die Hintergrundfarbe ihres Vaterobjekts und nicht ihre eigene.

Siehe auch

Attribute *.curvalue*, *.linemotion*, *.pagemotion*

2.14 .attribute

Mit Hilfe dieses Attributes kann beim Ereignisobjekt ***thisevent*** ein geändertes Attribut abgefragt werden (gültig nur beim *changed*-Ereignis).

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>attribute</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_attribute	Bezeichner: AT-attribute
Datentyp: DT_attribute	Datentyp: DT-attribute

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	<i>thisevent</i>

Siehe auch

Kapitel „Ereignisobjekt thisevent“ im Handbuch „Regelsprache“

2.15 .attribute[integer]

Das Attribut *.attribute* dient je nach Indizierung zum Abfragen des Namens oder des Wertes eines Attributs des DOM-Knotens.

Ist der Index eine Zahl, dann wird der Name des entsprechenden Attributs des DOM-Knotens geliefert. Es ist zu beachten, dass Attribute eines DOM-Knotens primär unsortiert sind. Ist der Index ein String, dann stellt der Index den Namen eines Attributes dar und es wird der Wert des Attributes zurückgeliefert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get	nein
<i>string, object [text]</i>	set (nur bei Stringindex)	

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_attribute	Bezeichner: AT-attribute
Datentyp: DT_integer, DT_string	Datentyp: DT-integer, DT-string

Vererbung
nein

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	doccursor

Eine Zuweisung auf das mit einem String indizierte Attribut *.attribute* legt ein entsprechendes Attribut am DOM-Knoten an. Eine Zuweisung eines Leerstrings löscht das entsprechende Attribute des DOM-Knotens. Eine Zuweisung auf ein mit einer Zahl indiziertes Attribut *.attribute* ist nicht erlaubt.

Das Attribut ist für den XML-Cursor verfügbar, es ist nicht vererbbar. Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

Beispiel

Das Attribut „Bestellnummer“ des XML-Elements auf das der XML-Cursor zeigt, wird mit folgender Anweisung auf den Wert *0815* gesetzt:

```
this.attribute["Bestellnummer"] := "0815";
```

Alle Attribute des XML-Elements auf das der XML-Cursor zeigt, können folgendermaßen ausgegeben werden:

```
variable integer I;  
variable string Name;  
  
for I := 1 to this.count[.attribute] do  
    Name := this.attribute[I];
```

```
    print Name + " = " + this.attribute[Name];  
endfor
```

2.16 .autoalign

Dieses Attribut der **toolbar** steuert, ob die **toolbar** in ihrer Zeile oder Spalte automatisch angeordnet wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_autoalign	Bezeichner: AT-autoalign
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	toolbar

Wertebereich

true

Die **toolbar** wird automatisch hinter der letzten **toolbar** einer Zeile oder Spalte bzw. am Anfang von leeren Zeilen und Spalten angeordnet.

false

Die **toolbar** wird nicht automatisch angeordnet.

2.17 .autosize

Dieses Attribut der **toolbar** definiert, ob die **toolbar** automatisch in der entsprechenden Richtung vergrößert werden soll um die Symbolleisten-Zeile bzw. -Spalte komplett auszufüllen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_autosize	Bezeichner: AT-autosize
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	toolbar

Beginnend bei der letzten **toolbar** in einer Zeile oder Spalte werden **toolbars** mit **.autosize = true** – unabhängig von **.sizeable** – vergrößert, wenn freier Platz vorhanden ist,

Die Vergrößerung beschränkt sich auf die Andockrichtung: Bei horizontaler Ausrichtung wird die Breite vergrößert, bei vertikaler Ausrichtung die Höhe.

Wertebereich

true

Die **toolbar** wird automatisch vergrößert und füllt den freien Platz in der Zeile oder Spalte aus.

false

Die **toolbar** wird **nicht** automatisch vergrößert.

2.18 .backpage

Definiert die Ecke, in der die seitlich sichtbaren Seitenkanten zusammentreffen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_binding	Bezeichner: AT-bindinge
Datentyp: DT_enum	Datentyp: DT-enum

Standardwert
bp_bottomright

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	notebook

Wertebereich

- bp_bottomright*** (Standard)
Seitenkanten sind unten und rechts sichtbar.
- bp_bottomleft*
Seitenkanten sind unten und links sichtbar.
- bp_topright*
Seitenkanten sind oben und rechts sichtbar.
- bp_topleft*
Seitenkanten sind oben und links sichtbar.

Im folgenden werden die Auswirkungen der Attribute *.backpage* und *.direction* auf die Positionierung von Majortabs und Minortabs bzw. der Bindung aufgezeigt:

<i>.backpage</i>	<i>.direction</i>	Majortab	Minortab	Bindung
<i>bp_bottomright</i>	1	rechts	unten	links
<i>bp_bottomright</i>	2	unten	rechts	oben
<i>bp_bottomleft</i>	1	links	unten	rechts
<i>bp_bottomleft</i>	2	unten	links	oben
<i>bp_topright</i>	1	rechts	oben	links
<i>bp_topright</i>	2	oben	rechts	unten
<i>bp_topleft</i>	1	links	oben	rechts
<i>bp_topleft</i>	2	oben	links	unten

2.19 .barwidth

Mit diesem Attribut kann die Breite von Bars für interaktive Größenänderungen definiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_barwidth	Bezeichner: AT-barwidth
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	splitbox, window

splitbox

Mit diesem Attribut kann die Breite der Splitbars in Pixeln angegeben werden.

Microsoft Windows

Die Splitbars werden im 3D-Look gezeichnet. Nur wenn die Breite auf 1 gesetzt wird ist der Effekt weg. Die 1 Pixel dünne Linie wird dann aber in der Farbe von *.bordercolor* gezeichnet, falls diese gesetzt ist.

Motif

Die Splitbars verfügen über einen Griff (sash), an dem diese gezogen werden können. Die Größe dieses Griffes wird mit dem Attribut *.barwidth* gesteuert. Die Breite der Trennungslinie ist vom System vorgegeben und bleibt von den Änderungen dieses Attributs unbeeinflusst.

window

Mit diesem Attribut kann die Breite von Bars für interaktive Größenänderungen von Toolbars definiert werden.

Siehe auch

Objekt toolbar

2.20 .bgc

Dieses Attribut definiert die Hintergrundfarbe eines Objekts.

Definition

<i>Datentyp</i> <i>object</i> [color]	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
---	----------------------------	-------------------------------

<i>C</i> Bezeichner: AT_bgc Datentyp: DT_color	<i>COBOL</i> Bezeichner: AT-bgc Datentyp: DT-color
--	--

Klassifizierung
Layoutattribut

Hinweise zum IDM FÜR WINDOWS

- » Beim **poptext** (Combobox) wird *.bgc* wie folgt unterstützt:
 - » Der geschlossene poptext, das heißt der angezeigte Eintrag besitzt Vorder- und Hintergrundfarbe.
 - » Die aufgeklappte Box des poptextes besitzt eine Standardfarbe.
- » Bei **menubox**, **menuitem**, **menusep** und **messagebox** wird *.bgc* ignoriert.
- » Beim **pushbutton** wird immer die systemweite Standardfarbe für den Hintergrund verwendet.
- » Darstellung der **Scrollbars** an IDM-Objekten (z.B. **edittext**, **groupbox**, **listbox**, **treeview**...): Unter Microsoft Windows sind die Scrollbars ein spezieller Rahmenstil und keine eigenständigen Objekte. Hierdurch kann der ISA Dialog Manager in der Regel keinen Einfluss auf die Farbe der Scrollbar-Elemente (Scroll-Leiste, Scroll-Pfeile, Schiebeknopf) nehmen. Dies führt beispielsweise dazu, dass die Scrollbar-Elemente in der vorgesehenen Systemfarbe (meist Grau) dargestellt werden, obwohl für das Objekt eine andere Hintergrundfarbe (*.bgc*) definiert ist.
- » Bei aktiven „Visual Styles“ verwenden die Windows-Objekte **checkbox** und **radiobutton** eine falsche Hintergrundfarbe. Der IDM versucht dies zu korrigieren. Ob das gelingt, hängt aber von verschiedenen Einstellungen der „Visual Styles“ ab.
Um sicher zu gehen, sollte bei den Vaterobjekten, insbesondere wenn es sich um eine **notepage** handelt, eine Hintergrundfarbe gesetzt werden. Allerdings ist bei der **notepage** dann ein zusätzlicher Rand sichtbar.
- » **scrollbar**-Objekte mit *.arrows = false* verwenden die Hintergrundfarbe ihres Vaterobjekts und nicht ihre eigene.

2.21 .bgc[index]

Dieses Attribut definiert die Hintergrundfarbe eines einzelnen Feldes in einem **tablefield**.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_bgc	Bezeichner: AT-bgc
Datentyp: DT_color	Datentyp: DT-color

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.22 .binding

Definiert die Art, wie ein **notebook** gebunden werden soll.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_binding	Bezeichner: AT-binding
Datentyp: DT_enum	Datentyp: DT-enum

Standardwert
bind_solid

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	notebook

Wertebereich

bind_solid (Standard)

Die gebundene Kante sieht aus wie mit einer Klebebindung.

bind_spiral

Die gebundene Kante sieht aus wie mit einer Spiralbindung (nur IDM FÜR MOTIF).

bind_organizer

Die gebundene Kante sieht aus wie bei einem Organizer (nur IDM FÜR WINDOWS).

bind_none

Die gebundene Kante weist keine Bindung auf (nur IDM FÜR MOTIF).

Das Aussehen ist ähnlicher zum Erscheinungsbild unter MICROSOFT WINDOWS.

2.23 .bordercolor

Dieses Attribut definiert die Randfarbe des Objektes.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

C
Bezeichner: AT_bordercolor
Datentyp: DT_color

COBOL
Bezeichner: AT-bordercolor
Datentyp: DT-color

Klassifizierung
Layoutattribut

Siehe auch

Attribute *.borderstyle*, *.borderwidth*

2.24 .borderraster

Mit Hilfe dieses Attributes kann bei einem Objekt mit Rahmen konfiguriert werden, dass die Geometrie so berechnet wird, als ob es keinen Rahmen besäße.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C

Bezeichner: AT_borderraster

Datentyp: DT_boolean

COBOL

Bezeichner: AT-borderraster

Datentyp: DT-boolean

Standardwert

true

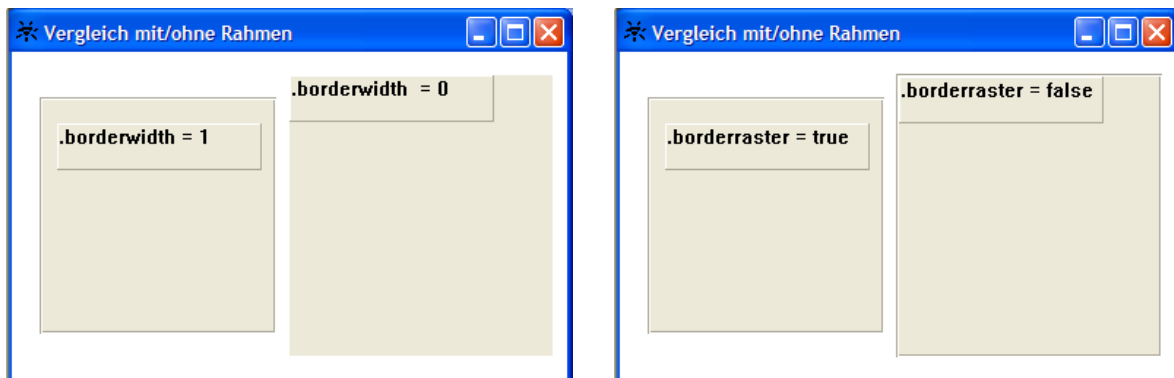
Klassifizierung

Geometrieattribut

Normalerweise werden folgende Anpassungen bei Objekten mit Rahmen ausgeführt:

- » Verschiebung um eine halbe Rastereinheit bei der Berechnung der Position eines Objektes mit Rasterkoordinaten.
- » Verkleinerung um eine Rastereinheit bei der Berechnung der Größe eines Objektes mit Rastergröße.
- » Verschiebung um eine halbe Rastereinheit bei der Berechnung der Position der Kindobjekte mit Rasterkoordinaten, sofern das Objekt Kinder besitzen kann.

Diese Anpassungen lassen sich mit diesem Attribut abschalten. Der Effekt ist analog zum Abschalten des Rahmens an einem groupbox Objekt (*.borderwidth := 0*).



Der Unterschied besteht darin, dass der Rahmen bei *.borderraster = false* weiterhin gezeichnet wird. Das Attribut zeigt nur Wirkung, wenn das Objekt einen Rahmen besitzt und wenn Rasterkoordinaten verwendet werden. Beeinflusst werden dann die Position (*.posraster = true*), die Größe (*.sizeraster =*

true) und gegebenenfalls auch die Position der Kindobjekte, wenn deren Position in Raster angegeben wird.

Die folgende Tabelle gibt einen Überblick über die Unterstützung des Attributs *.borderraster* an den einzelnen Dialog Manager Objekten:

canvas	unterstützt, wenn <i>.borderwidth > 0</i>
checkbox	nicht unterstützt
control	nicht unterstützt
edittext	unterstützt, wenn <i>.multiline = true</i>
groupbox	unterstützt, wenn <i>.borderwidth > 0</i>
image	nicht unterstützt
layoutbox	unterstützt, wenn <i>.borderwidth > 0</i>
listbox	unterstützt
notebook	unterstützt
notepage	unterstützt, wenn <i>.borderwidth > 0</i>
poptext	unterstützt, wenn <i>.style = listbox</i>
progressbar	nicht unterstützt
pushbutton	nicht unterstützt
radiobutton	nicht unterstützt
rectangle	unterstützt, wenn <i>.borderwidth > 0</i>
scrollbar	nicht unterstützt
spinbox	nicht unterstützt
splitbox	unterstützt, wenn <i>.borderwidth > 0</i>
statictext	nicht unterstützt
statusbar	nicht unterstützt
tablefield	unterstützt
toolbar	unterstützt, wenn <i>.docking = dock_window</i> OR <i>.borderwidth > 0</i>
treeview	unterstützt
window	unterstützt

Wertebereich

true

Die Berechnung der Rasterkoordinaten findet wie üblich (analog zu früheren Dialog Manager Versionen) statt.

Dies bedeutet für Objekte mit Rahmen:

- » Das Objekt wird um eine halbe Rastereinheit verschoben, wenn die Position in Rastereinheiten angegeben ist (*.posraster = true*).
- » Das Objekt wird um eine Rastereinheit verkleinert, wenn die Größe in Rastereinheiten angegeben ist (*.sizeraster = true*).
- » Die Kindobjekte werden um eine halbe Rastereinheit verschoben, wenn ihre Position in Rastereinheiten angegeben ist (*.posraster = true*).
Das Objekt muss natürlich Kindobjekte zulassen und besitzen.

false

Die Berechnung der Rasterkoordinaten eines Objektes mit Rahmen findet so statt, als wenn das Objekt keinen Rahmen besäße. Die beim Wert „true“ erwähnten Anpassungen finden nicht statt.

Hinweis

Es werden lediglich die beim Wert „true“ erwähnten Anpassungen nicht durchgeführt; es kann also trotz *.borderraster = false* sein, dass ein Objekt mit Rahmen nicht exakt in derselben Zeile dargestellt wird, wie ein anderes Objekt ohne Rahmen. Dies liegt dann an anderen Faktoren. Die entsprechenden Objekte würden auch nicht exakt in einer Zeile liegen, wenn sie in Pixelkoordinaten positioniert würden.

Referenzen und Abhängigkeiten

Die Einteilung Objekt mit Rahmen bzw. Objekt ohne Rahmen ist eine Dialog Manager Einteilung und betrifft die Art und Weise, wie Rasterkoordinaten in Pixelwerte umgerechnet werden. Diese Einteilung hat nichts damit zu tun, ob vom betreffenden Fenstersystem tatsächlich ein Rahmen um das Objekt gezeichnet wird.

Momentan gilt folgenden Einteilung:

Objekte ohne Rahmen

canvas wenn *.borderwidth = 0*

checkbox

control

edittext wenn *.multiline = false*

groupbox wenn *.borderwidth = 0*

image

layoutbox wenn *.borderwidth = 0*

notepage wenn *.borderwidth = 0*

poptext wenn *.style <> listbox*

progressbar

pushbutton

radiobutton

rectangle wenn *.borderwidth = 0*

scrollbar

spinbox

splitbox wenn *.borderwidth = 0*

statictext

toolbar wenn *.docking <> dock_window AND .borderwidth = 0*

Objekte mit Rahmen

canvas wenn *.borderwidth > 0*

edittext wenn *.multiline = true*

groupbox wenn *.borderwidth > 0*

layoutbox wenn *.borderwidth > 0*

listbox

notebook

notepage wenn *.borderwidth > 0*

poptext wenn *.style = listbox*

rectangle wenn *.borderwidth > 0*

splitbox wenn *.borderwidth > 0*

statusbar wenn *.borderwidth > 0*

tablefield

toolbar wenn *.docking = dock_window OR .borderwidth > 0*

treeview

window

Das *.borderraster*-Attribut wirkt nur auf das **toolbar**-Objekt, wenn es ausgedockt ist (*.docking = dock_window*). Im eingedockten Zustand wirkt das Attribut nur auf die Kindobjekte des **toolbar**-Objekts (vorausgesetzt: *.borderwidth > 0*).

Besitzt ein Objekt keinen Rahmen, dann bleibt eine Änderung von *.borderraster* ohne Wirkung.

Durch das Setzen von *.borderraster* auf den Wert *false* werden Objekte nicht Pixelgenau auf derselben Position erscheinen. Der Effekt wird vielmehr so sein, wie wenn die beteiligten Objekte mit Pixelkoordinaten auf dieselbe Position gesetzt worden wären.

Es ist zu beachten, dass Objekte ohne Rahmen auch einzeilig sein können. Solche Objekte werden in der angegebenen Rasterzeile zentriert, wenn sie Rasterkoordinaten verwenden (*.posraster = true*) und keine Höhe besitzen (*.height = 0*).

Einzeilige Objekte

checkbox

edittext wenn *.multiline = false*

poptext wenn *.style <> listbox*

progressbar wenn *.direction = 2*

pushbutton

radiobutton

scrollbar wenn *.direction = 2*

spinbox

statictext

Andere Anpassungen werden weiterhin durchgeführt, auch wenn das Attribut *.borderraster* den Wert *false* besitzt. Insbesondere werden die Anpassungen zur Kompatibilität mit anderen Versionen, wie *.options[opt_wntsizebug_compat]* und *.options[opt_w2kprefsize_compat]*, weiterhin durchgeführt.

2.25 .borderstyle

Dieses Attribut definiert den Stil, also die Darstellung und Ausprägung des Rahmens eines Objekts.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_borderstyle	Bezeichner: AT-borderstyle
Datentyp: DT_enum	Datentyp: DT-enum

<i>Standardwert</i>	<i>Vererbung</i>
<i>border_compat</i>	ja

Klassifizierung
Layoutattribut

Wertebereich

border_compat

Kompatibilitätsmodus: Rahmendarstellung wie im IDM 5 durch die Attribute *.bordercolor* und *.borderwidth* bestimmt.

Dieser Wert kann nicht aktiv gesetzt werden.

border_none

Es wird kein Rahmen (Begrenzung) um das Objekt gezeichnet.

border_plain

Der Rahmen wird als einfache, flache Linie um das Objekt gezeichnet, in Breite und Farbe beeinflussbar (sofern vom jeweiligen WSI unterstützt).

border_raised

Der Rahmen wird als 3D-Rahmen um das Objekt gezeichnet, der Rahmen ist nach außen erhoben und in der Breite beeinflussbar (sofern vom jeweiligen WSI unterstützt).

border_sunken

Der Rahmen wird als 3D-Rahmen um das Objekt gezeichnet, der Rahmen ist nach innen gedrückt und in der Breite beeinflussbar (sofern vom jeweiligen WSI unterstützt).

border_toolkit

Es wird der Toolkitrahmen gezeichnet, wie vom WSI für das jeweilige Objekt vorgesehen.

Tabelle 1: Unterstützung des Attributs *.borderstyle*

Objekt	Unterstützung von <i>.borderstyle</i>
canvas	Attribut wird unterstützt.
groupbox	Attribut wird unterstützt.
layoutbox	Attribut wird unterstützt.

Objekt	Unterstützung von .borderstyle
rectangle	Attribut wird unterstützt.
splitbox	Attribut wird unterstützt.
edittext	Attribut wird unterstützt, jedoch nur <i>border_none</i> und <i>border_toolkit</i> zulässig. <i>border_plain</i> , <i>border_raised</i> und <i>border_sunken</i> werden als <i>border_toolkit</i> umgesetzt.
image	Attribut wird unterstützt, jedoch nur <i>border_none</i> und <i>border_toolkit</i> zulässig. <i>border_plain</i> , <i>border_raised</i> und <i>border_sunken</i> werden als <i>border_toolkit</i> umgesetzt.
progressbar	Attribut wird unterstützt, jedoch nur <i>border_none</i> und <i>border_toolkit</i> zulässig. <i>border_plain</i> , <i>border_raised</i> und <i>border_sunken</i> werden als <i>border_toolkit</i> umgesetzt.
spinbox	Attribut wird unterstützt, jedoch nur <i>border_none</i> und <i>border_toolkit</i> zulässig. <i>border_plain</i> , <i>border_raised</i> und <i>border_sunken</i> werden als <i>border_toolkit</i> umgesetzt.
statusbar	Attribut wird unterstützt, jedoch nur <i>border_none</i> und <i>border_toolkit</i> zulässig. <i>border_plain</i> , <i>border_raised</i> und <i>border_sunken</i> werden als <i>border_toolkit</i> umgesetzt.
toolbar	Attribut wird unterstützt, jedoch nur <i>border_none</i> und <i>border_toolkit</i> zulässig. <i>border_plain</i> , <i>border_raised</i> und <i>border_sunken</i> werden als <i>border_toolkit</i> umgesetzt.
window	Attribut wird unterstützt, jedoch nur <i>border_none</i> und <i>border_toolkit</i> zulässig. <i>border_plain</i> , <i>border_raised</i> und <i>border_sunken</i> werden als <i>border_toolkit</i> umgesetzt.
checkbox	Attribut wird nicht unterstützt.
control	Attribut wird nicht unterstützt.
listbox	Attribut wird nicht unterstützt.
notebook	Attribut wird nicht unterstützt.
notepage	Attribut wird nicht unterstützt.
poptext	Attribut wird nicht unterstützt.
pushbutton	Attribut wird nicht unterstützt.
radiobutton	Attribut wird nicht unterstützt.

Objekt	Unterstützung von .borderstyle
scrollbar	Attribut wird nicht unterstützt.
statictext	Attribut wird nicht unterstützt.
tablefield	Attribut wird nicht unterstützt.
treeview	Attribut wird nicht unterstützt.

Verfügbarkeit

Ab IDM-Version A.06.01.a

2.26 .borderwidth

Dieses Attribut definiert die Breite eines Objektrandes.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_borderwidth	Bezeichner: AT-borderwidth
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Attribute *.bordercolor*, *.borderstyle*

2.27 .button[integer]

Eine **messagebox** kann maximal drei Buttons anzeigen. Die Belegung jedes dieser Buttons wird mit dem Attribut *.button[integer]* gesetzt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_button	Bezeichner: AT-button
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	messagebox

Wertebereich

button_abort
Zeigt die Schaltfläche „Abbrechen“ („Abort“) an.

button_cancel
Zeigt die Schaltfläche „Abbrechen“ („Cancel“) an.

button_ignore
Zeigt die Schaltfläche „Ignorieren“ an.

button_no
Zeigt die Schaltfläche „Nein“ an.

button_ok
Zeigt die Schaltfläche „OK“ an.

button_retry
Zeigt die Schaltfläche „Wiederholen“ an.

button_yes
Zeigt die Schaltfläche „Ja“ an.

Für *.button[1-3]* sind folgende Button-Kombinationen erlaubt:

<i>.button[1]</i>	<i>.button[2]</i>	<i>.button[3]</i>
<i>button_ok</i>	<i>nobutton</i>	<i>nobutton</i>
<i>button_ok</i>	<i>button_cancel</i>	<i>nobutton</i>
<i>button_cancel</i>	<i>nobutton</i>	<i>nobutton</i>
<i>button_retry</i>	<i>button_abort</i>	<i>nobutton</i>
<i>button_retry</i>	<i>button_abort</i>	<i>button_ignore</i>

.button[1]	.button[2]	.button[3]
<i>button_yes</i>	<i>button_no</i>	<i>nobutton</i>
<i>button_yes</i>	<i>button_no</i>	<i>button_cancel</i>

Wird ein Button als *nobutton* definiert, wird er nicht angezeigt.

Wird eine nicht erlaubte Button-Kombination gesetzt, so kann diese vom Fenstersystem als Fehler behandelt werden, d.h. die **messagebox** erscheint nicht und der Rückgabewert ist *nobutton*.

Anmerkung zum IDM FÜR WINDOWS

Die Kombinationen *button_cancel* – *nobutton* – *nobutton* und *button_retry* – *button_abort* – *nobutton* werden von Microsoft Windows nicht unterstützt.

Achtung

Von der obigen Tabelle abweichende, unter Motif erlaubte Kombinationen erscheinen unter Microsoft Windows nicht!

Anmerkung zum IDM FÜR MOTIF

Normalerweise können Sie mit der **Escape**-Taste einen „Cancel“-Button aktivieren.

Beispiel

```
messagebox MessageBox1
{
    .text      "This text notifies you of an error.";
    .title     "Error Message";
    .icon      icon_exclamation;
    .button[1] button_ok;
    .button[2] button_cancel;
}
```

2.28 .bw

Dieses Attribut bestimmt, wie die **color**-Ressource auf einem Schwarz-Weiß-Monitor (*setup.color_type* = *coltype_bw*) dargestellt wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	nein

C
 Bezeichner: AT_bw
 Datentyp: DT_enum

COBOL
 Bezeichner: AT-bw
 Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	color

Wertebereich

color_black

Wird als Schwarz angezeigt.

color_white

Wird als Weiß angezeigt.

2.29 .calendaralignment

Dieses Attribut definiert die Position des aufklappbaren Kalenders bezüglich des Objekts.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_calendaralignment	Bezeichner: AT-calendaralignment
Datentyp: DT_integer	Datentyp: DT-integer

<i>Standardwert</i>	<i>Vererbung</i>
1	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	datetime

Wertebereich

- 1**
Der Kalender ist **links** ausgerichtet (Standard).
- 0**
Der Kalender ist **zentriert**.
Dieser Wert wird nicht unterstützt; es wird der Standardwert verwendet.
- 1**
Der Kalender ist **rechts** ausgerichtet.

2.30 .canvasfunc

Das Attribut *.canvasfunc* spezifiziert eine Funktion, die bei einem bestimmten Ereignis aufgerufen werden soll. Diese Funktion muss von der Anwendung bereitgestellt werden und muss vom Typ *canvasfunc* sein.

Definition

<i>Datentyp</i> <i>object</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_canvasfunc Datentyp: DT_func		<i>COBOL</i> Bezeichner: AT-canvasfunc Datentyp: DT-func
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> canvas	

2.31 .certificatefile

In diesem Attribut wird die Zertifikatsdatei definiert, die für eine SSL-Verbindung verwendet wird.

Standardmäßig wird die Datei **cert.pem** aus dem Installationsverzeichnis der Anwendung verwendet.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_certificatefile	Bezeichner: AT-certificatefile
Datentyp: DT_string	Datentyp: DT-string

Vererbung
ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Verfügbarkeit

Ab IDM-Version A.06.02.g

2.32 .changedir

Dieses Attribut steuert die Übernahme des Verzeichnisses, in dem sich der Benutzer bei erfolgreicher Selektion in einem Dateiauswahlfenster (**filereq**) befand, in das Attribut *.directory*. Steht es auf *true*, wird der Pfad des Verzeichnisses übernommen. Dadurch wird beim nächsten Öffnen der Dateiauswahl im gleichen Verzeichnis weitergemacht, in dem man sich zuletzt befunden hat.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_changedir	Bezeichner: AT-changedir
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	filereq

Siehe auch

Attribut *.directory*

2.33 .child[integer]

Dieses Attribut wird verwendet, um Kind-Objekte zu definieren. Es setzt das Objekt auf die Position "i" in der Kinder-Liste und kann entsprechend abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_child	Bezeichner: AT-child
Datentyp: DT_object	Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.34 .childcount

Dieses Attribut fragt die Anzahl der Kinder eines Objektes ab.

Definition

Datentyp
integer

Zugriff
get

C
Bezeichner: AT_childcount
Datentyp: DT_integer

COBOL
Bezeichner: AT-childcount
Datentyp: DT-integer

Klassifizierung
Hierarchieattribut

2.35 .class

Mit Hilfe dieses Attributs können Sie die Klasse eines Objekts bzw. einer Ressource, z.B. *pushbutton*, *color*, *function* erfragen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>class</i>	get

C
Bezeichner: AT_class
Datentyp: DT_class

COBOL
Bezeichner: AT-class
Datentyp: DT-class

Klassifizierung
Standardattribut

2.36 .closeable

Dieses Attribut definiert, ob ein Fenster schließbar sein soll bzw. ob es einen entsprechenden Mechanismus (z.B. eine Closebox oder einen Closebutton) besitzen soll.

Das Fenster besitzt einen solchen Schließmechanismus, wenn der Wert dieses Attributes auf *true* gesetzt ist. Ist der Wert auf *false* gesetzt, wird keine Closebox angezeigt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_closeable	Bezeichner: AT-closeable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Besonderheit unter Motif

Das Attribut kann, abhängig vom verwendeten Display- bzw. Desktop Manager, nicht im sichtbaren Zustand umgesetzt werden; unter Umständen ist ein Setzen zur Laufzeit auch nicht möglich. In manchen Fällen kann es helfen, die Sichtbarkeit des Fensters umzuschalten.

Da unter MOTIF das Umsetzen dieses Attributs direkt vom verwendeten Display- bzw. Desktop Manager abhängig ist, wird empfohlen, das Attribut nur statisch bzw. direkt nach Erzeugen einer Instanz mittels *:create(..., true)* im unsichtbaren Zustand zu setzen.

2.37 .codepage

Dieses Attribut definiert die Codepage, mit der Applikationsfunktionen des **application**-Objekts aufgerufen werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
enum	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_codepage	Bezeichner: AT-codepage
Datentyp: DT_enum	Datentyp: DT-enum

Vererbung
ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Durch Setzen dieses Attributs erfolgt eine Redefinition der IDM-Applikations-Codepage für die Zeit des Aufrufs, die ansonsten nur durch Nutzung der IDM-Schnittstellenfunktion DM_Control mit der Aktion *DMF_SetCodePage* erfolgen kann.

Alternativ zur Nutzung dieses Attributs kann auch **DM_Control** genutzt werden, unter Angabe des Applikationsobjektes.

Durch Nutzung einer applikationsspezifischen Codepage kann somit sehr einfach die Aufteilung von Applikationsfunktionen, welche Strings in unterschiedlicher Codierung erfordern, erreicht werden.

Von einer dynamischen Umsetzung dieses Attributs, nicht nur während der Phase der aktivierten Applikation, wird dringend abgeraten. Wenn Applikationsfunktionen mit **Record**-Parametern über eine dynamische Anbindung aufgerufen werden, sollte die Record-Definition (C-Header bzw. COBOL Copy-Strecke) übereinstimmen.

Wertebereich

cp_acp
Aktuell von einer Anwendung verwendete ANSI-Codepage unter MICROSOFT WINDOWS.

Verfügbarkeit
Nur unter MICROSOFT WINDOWS.

cp_ascii
ASCII Zeichencodierung.

cp_cp1252
Westeuropäische Zeichencodierung gemäß MICROSOFT WINDOWS Codepage 1252.

cp_cp437
Englische Zeichencodierung gemäß IBM-Codepage 437 (MS-DOS).

cp_cp850
Westeuropäische Zeichencodierung gemäß IBM-Codepage 850 (MS-DOS).

cp_dec169

Zeichencodierung gemäß DEC-Codepage 169.

cp_euc

Zeichencodierung gemäß „Extended UNIX Code“ (EUC).

cp_hp15

Westeuropäische 16-Bit Zeichencodierung von HP-Systemen.

cp_iso6937

Westeuropäische Codierung mit variabler Länge nach ISO 6937.

cp_iso8859

Westeuropäische Zeichencodierung Latin-1 nach ISO 8859-1.

cp_jap15

Japanische 16-Bit Zeichencodierung von HP-Systemen.

cp_kor15

Koreanische 16-Bit Zeichencodierung von HP-Systemen.

cp_prc15

Traditionelle chinesische 16-Bit Zeichencodierung (Langzeichen) von HP-Systemen.

cp_roc15

Vereinfachte chinesische 16-Bit Zeichencodierung (Kurzzeichen) von HP-Systemen.

cp_roman8

8-Bit Zeichencodierung gemäß HP-Codepage Roman-8.

cp_ucp

Benutzerdefinierte Codepage („User Code Page“); Konvertierung in eine beliebige Codepage mit **iconv()** durch DM_ControlEx mit der Aktion *DMF_SetUserCodePage*.

Verfügbarkeit

Nur auf UNIX/LINUX-Systemen. Unter MICROSOFT WINDOWS werden nicht darstellbare Zeichen in „?“ konvertiert.

cp_utf16

16-Bit Unicode-Codierung mit variabler Länge von 2 bis 4 Byte.

Es gibt zwei Varianten:

- » BE – Big Endian: höherwertige(s) Byte(s) zuerst.
- » LE – Little Endian: niederwertige(s) Byte(s) zuerst.

UTF-16 ohne Angabe der Byte-Reihenfolge entspricht unter MICROSOFT WINDOWS der LE-Variante, auf UNIX/LINUX-Systemen dagegen der BE-Variante.

cp_utf16b

16-Bit Unicode-Codierung mit variabler Länge von 2 bis 4 Byte in der BE-Variante (Big Endian: höherwertige(s) Byte(s) zuerst).

Dies ist der Standard für UTF-16 auf UNIX/LINUX-Systemen.

cp_utf16l

16-Bit Unicode-Codierung mit variabler Länge von 2 bis 4 Byte in der LE-Variante (Little Endian: niederwertige(s) Byte(s) zuerst).

Dies ist der Standard für UTF-16 unter MICROSOFT WINDOWS.

cp_utf8

8-Bit Unicode-Codierung mit variabler Länge, entspricht im Bereich 0 – 127 der ASCII-Codierung.

cp_utfwin

16-Bit Unicode-Codierung wie *cp_utf16l* mit Konvertierung von Zeilenumbrüchen `\r\n` → `\n`.

cp_wcs

„Wide Character String“ (Datentyp *wchar_t**) Zeichencodierung, abhängig vom System und den Regions- und Spracheinstellungen.

cp_winansi

MICROSOFT WINDOWS Zeichencodierung.

Anmerkung

Es ist zu beachten, dass die Codepage nicht an eine DDM-Server-Applikation weitergegeben wird. Auf der DDM-Serverseite konnte bisher schon immer die Codepage über **DM_Control** umgesetzt werden.

Verfügbarkeit

Ab IDM-Version A.06.01.d

Siehe auch

Kapitel „Dynamische Anbindung von Record-Funktionen“ im Handbuch „C-Schnittstelle - Grundlagen“

Kapitel „Dynamische Anbindung von Record-Funktionen“ im Handbuch „COBOL-Schnittstelle“

Startoption `+writeheader`

Setzen der Codepage über `DM_Control` im Handbuch „C-Schnittstelle - Funktionen“

Setzen einer benutzerdefinierten Codepage über `DM_ControlEx` im Handbuch „C-Schnittstelle - Funktionen“

2.38 .colalignment[integer]

Mit Hilfe dieses Attributes kann die Ausrichtung des Inhalts bezüglich seines Feldes (links, zentriert, rechts) beschrieben werden. *.colalignment[0]* ist dabei der Defaultwert, der für alle anderen genommen wird, wenn nichts weiter angegeben ist.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
-----------------------------------	----------------------------	-------------------------------

C
Bezeichner: AT_colalignment
Datentyp: DT_integer

COBOL
Bezeichner: AT-colalignment
Datentyp: DT-integer

<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield
---	------------------------------

Wertebereich

-1	rechtsbündig
0	horizontal zentriert
1	linksbündig

2.39 .colcount

Das Attribut definiert die Anzahl der Spalten.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_colcount	Bezeichner: AT-colcount
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listview, tablefield

tablefield

Mit Hilfe dieses Attributes kann die Anzahl der insgesamt im **tablefield** vorhandenen Spalten definiert werden. Der Wertebereich reicht von 0 bis 65535.

Diese Anzahl muss nicht unbedingt wirklich in dem Tablefield vorhanden sein, sondern soll dem Tablefield viel mehr dazu dienen, die horizontale Scrollbar richtig zu berechnen.

listview

Das Attribut definiert die Anzahl der Spalten in der Detailansicht.

Wertebereich

- 0 Leert das **listview**-Objekt.
- > 0 Anzahl der Spalten in der Detailansicht.

Der Wert von **.colcount** kann implizit erhöht werden, indem **.content** um jeweils eine neue Spalte erweitert wird.

Spalte 1 wird in allen Ansichten angezeigt.

Bei ungültigen Werten wird der Standardwert 1 verwendet. Der Attributwert wird allerdings nicht geändert.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

Siehe auch

Attribut **.rowcount**

2.40 .colfirst

Mit Hilfe dieses Attributes kann die Nummer der neben den Zeilenköpfen sichtbaren Spalte definiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_colfirst	Bezeichner: AT-colfirst
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.41 .colheader

Mit Hilfe dieses Attributes kann die Anzahl der Spalten (im Sinne des Tablefields, also keine Buchstaben) definiert werden, die als Zeilenbeschriftung dienen und damit nicht horizontal gescrollt werden sollen. Wertebereich 0 - 255 .

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_colheader Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-colheader Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield	

2.42 .colheadfgc

Mit Hilfe dieses Attributes kann die Farbe definiert werden, mit der die Spaltenköpfe geschrieben werden sollen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

C
Bezeichner: AT_colheadfgc
Datentyp: DT_color

COBOL
Bezeichner: AT-colheadfgc
Datentyp: DT-color

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.43 .colheadfont

Mit Hilfe dieses Attributes kann der Zeichensatz definiert werden, in dem die Spaltenköpfe geschrieben werden sollen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [font]	get, set	ja

C
Bezeichner: AT_colheadfont
Datentyp: DT_font

COBOL
Bezeichner: AT-colheadfont
Datentyp: DT-font

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.44 .colheadshadow

Mit Hilfe dieses Attributes kann definiert werden, in welcher Form die Spaltenköpfe gestaltet werden. Dabei bedeutet *true* Darstellung mit Schatten wie bei einem Button.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_colheadshadow	Bezeichner: AT-colheadshadow
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Anmerkung für DM mit Motif

Das Attribut *.colheadshadow* wird nicht unterstützt.

2.45 .colheadvisible

Mit Hilfe dieses Attributes kann angegeben werden, ob die Spaltenköpfe angezeigt werden sollen oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C
Bezeichner: AT_colheadvisible
Datentyp: DT_boolean

COBOL
Bezeichner: AT-colheadvisible
Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.46 .collinewidth[integer]

Mit Hilfe dieses Attributes kann die Breite der horizontalen Linien definiert werden, die im **tablefield** gezeichnet werden. *.collinewidth[0]* ist dabei der Defaultwert, der für alle anderen genommen wird, wenn nichts weiter angegeben ist.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_collinewidth Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-collinewidth Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield	

2.47 .color

Mit Hilfe dieses Setup-Objekt-Attributes kann die Farbvariante abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_color	Bezeichner: AT-color
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

2.48 .color_type

Mit Hilfe dieses Setup-Objekt-Attributes kann abgefragt werden, ob ein Schwarz-Weiß-, Graustufen- oder Farb-Bildschirm benutzt wird.

Es wird bei Multiscreendialogen (nur IDM FÜR MOTIF) der Wert für den Default-Screen zurückgegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>enum</i>	get
 <i>C</i>	 <i>COBOL</i>
Bezeichner: AT_color_type	Bezeichner: AT-color-type
Datentyp: DT_enum	Datentyp: DT-enum
 <i>Klassifizierung</i>	 <i>Objekte</i>
Objektspezifisches Attribut	setup

Wertebereich

coltype_bw

Der verwendete Bildschirm ist ein Schwarz-Weiß-Monitor.

coltype_color

Der verwendete Bildschirm ist ein Farb-Monitor.

coltype_grey

Der verwendete Bildschirm ist ein Graustufen-Monitor.

2.49 .color_type[integer]

Mit Hilfe dieses Setup-Objekt-Attributes kann abgefragt werden, ob Bildschirm I ein Schwarz-Weiß-, Graustufen- oder Farb-Bildschirm ist.

Das indizierte Attribut ist nur bei Multiscreendialogen verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>enum</i>	get
 <i>C</i>	 <i>COBOL</i>
Bezeichner: AT_color_type	Bezeichner: AT-color-type
Datentyp: DT_enum	Datentyp: DT-enum
 <i>Klassifizierung</i>	 <i>Objekte</i>
Objektspezifisches Attribut	setup

Wertebereich

<i>coltype_bw</i>
Der verwendete Bildschirm ist ein Schwarz-Weiß-Monitor.
<i>coltype_color</i>
Der verwendete Bildschirm ist ein Farb-Monitor.
<i>coltype_grey</i>
Der verwendete Bildschirm ist ein Graustufen-Monitor.

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung. Es ist zu beachten, dass mit Screen-Index nicht die Screen-Nummer gemeint ist, die z.B. mit dem Programm **xdpyinfo** erhalten wird.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

Siehe auch

Ressource display

2.50 .colorcount

Mit Hilfe dieses Setup-Objekt-Attributes kann die Anzahl der Farben abgefragt werden.

Es wird bei Multiscreendialogen (nur Motif) der Wert für den Default-Screen zurückgegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_colorcount	Bezeichner: AT-colorcount
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkung

Unter MICROSOFT WINDOWS kann der Wert -1 vorkommen, wenn mehr als 256 Farben unterstützt werden.

2.51 .colorcount[integer]

Mit Hilfe dieses Setup-Objekt-Attributes kann die Anzahl der Farben abgefragt werden.

Das indizierte Attribut ist nur bei Multiscreendialogen (nur Motif) verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>
Bezeichner: AT_colorcount
Datentyp: DT_integer

<i>COBOL</i>
Bezeichner: AT-colorcount
Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung. Es ist zu beachten, dass mit Screen-Index nicht die Screen-Nummer gemeint ist, die z.B. mit dem Programm **xdpyinfo** erhalten wird.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

Siehe auch

Ressource display

2.52 .colorname[integer]

Attribut des **Setup**-Objekts, das eine Liste mit den Namen der im WSI verfügbaren Farben enthält.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_colorname	Bezeichner: AT-colorname
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Verfügbarkeit

Nur IDM FÜR QT, ab IDM-Version A.06.01.a.

IDM FÜR MOTIF & WINDOWS, Ab IDM-Version A.06.03.a

2.53 .colsizeable[integer]

Dieses einfach indizierte Attribut steuert die interaktive Vergrößerbarkeit von Spalten.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_colsizeable	Bezeichner: AT-colsizeable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Wird die Maus über den Rand eines manipulierbaren Feldes bewegt, so ändert sich der Mauszeiger und zeigt ein Größenänderungssymbol. Das Aussehen des Cursors ist vom Anwendungsprogrammierer nicht spezifizierbar. Bei den Spalten kann die Größenänderung jeweils am rechten Rand einer Spalte durchgeführt werden.

Der Benutzer kann mit der linken Maustaste die Größenänderung einleiten. Es erscheint eine graue schraffierte Linie, die die Position der Maus anzeigt (Die Breite der schraffierten Linie ist abhängig von dem Attribut *.collinewidth[integer]*). Wird die Maus (mit gedrückter linker Maustaste) bewegt, wandert die schraffierte Linie mit. Die Größenänderung wird abgeschlossen, indem die Maustaste losgelassen wird.

Dieses Attribut wird wie andere einfach indizierte Attribute beim Tablefield gehandhabt, d.h. in *.colsizeable[0]* steht der Default, der genommen werden soll, wenn für die Spalte kein Wert definiert ist. Der Default von *.colsizeable[0]* ist *false*.

2.54 .coltitle[integer]

Mit diesem Attribut werden die Spaltenüberschriften in der Detailansicht festgelegt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string, object [text]</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_coltitle	Bezeichner: AT-coltitle
Datentyp: DT_string, DT_text	Datentyp: DT-string, DT-text

<i>Standardwert</i>	<i>Vererbung</i>
""	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listview

Beim Schreibzugriff („set“) können Werte vom Datentyp *string* und **text**-Ressourcen angegeben werden. **text**-Ressourcen werden automatisch zu *string* gewandelt.

Lesezugriff („get“) liefert immer einen Wert vom Datentyp *string*.

Der Wertebereich des Index ist *0colcount*, wobei der Wert mit Index *0* als Standardwert für nicht gesetzte Werte im Bereich *1colcount* dient.

Wenn für eine Spalte keine Überschrift definiert wird und auch kein Standardwert *.coltitle[0]* definiert ist, dann wird diese Spalte ohne Spaltenüberschrift angezeigt.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

2.55 .colvisible[integer]

Dieses einfach indizierte Attribut steuert die Sichtbarkeit von Spalten, so dass sie je nach Anforderung ausgeblendet werden können.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_colvisible	Bezeichner: AT-colvisible
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

In manchen Anwendungen werden Informationen im Tablefield gehalten, die ausschließlich für die Anwendung und nicht für den Benutzer bestimmt sind. Diese Spalten bzw. Zeilen sollen dabei immer nur in speziellen Situationen angezeigt werden.

Dieses Attribut wird wie andere einfach indizierte Attribute beim Tablefield gehandhabt, d.h. in *.colvisible[0]* steht der Default, der genommen werden soll, wenn für die Spalte kein Wert definiert ist. Der Default ist *.colvisible[0]* auf *true*.

2.56 .colwidth[integer]

Mit Hilfe dieses Attributes kann die Breite der einzelnen Spalten definiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_colwidth	Bezeichner: AT-colwidth
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listview, tablefield

tablefield

Mit Hilfe dieses Attributes kann die Breite der einzelnen Spalten in Koordinateneinheiten definiert werden. (Pixel, falls *.sizeraster* nicht gesetzt ist; Raster, falls *.sizeraster* gesetzt ist). *.colwidth[0]* ist dabei der Defaultwert für alle nachfolgenden Spalten, falls dort kein Wert angegeben ist.

Anmerkung

Die Breite von Spalten im Tablefield (*.colwidth*) wurde in älteren Versionen des DM für Motif zu breit gewählt. Um die bisherige Spaltenbreite zu erhalten, kann das Attribut

```
.options[opt_old_colwidth] := true;
```

gesetzt werden.

Zusätzlich bietet das **Tablefield** die Option *opt_new_colwidth*. Wenn diese gesetzt ist, wird zur Berechnung der Spaltenbreite die korrekte Breite (Rastergrößen wie beim Layout der anderen Objekte) verwendet. Ein *true* hier hat Vorrang vor *.options[opt_old_colwidth] = true!*

listview

Dieses Attribut definiert die Breite der Spalten in der Detailansicht.

Der Wertebereich des Index ist *0colcount*, wobei der Wert mit Index *0* als Standardwert für nicht gesetzte Werte im Bereich *1colcount* dient.

Wertebereich

- 0** Die Spaltenbreite wird vom **listview** berechnet.
Der gesetzte Wert *0* bleibt erhalten, bis der Anwender die Spaltenbreite interaktiv ändert.
- > 0** Breite der jeweiligen Spalte in der Detailansicht.

Bei ungültigen Werten wird der Standardwert *0* verwendet. Der Attributwert wird allerdings nicht geändert.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

2.57 .configurable

Mit diesem Attribut können Sie definieren, ob ein **record** oder eine **globale Variable** konfigurierbar ist, d.h. in der mit **DM_LoadProfile()** geladenen Konfigurationsdatei gesetzt werden darf.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_configurable	Bezeichner: AT-configurable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	record, variable

Wertebereich

true

Der Wert kann aus einer Konfigurationsdatei geladen werden.

false

Das Attribut bzw. die globale Variable ist **nicht** konfigurierbar.

Bei **globale Variablen** ist die Konfigurierbarkeit durch das vorangestellte Keyword **config** gekennzeichnet.

Siehe auch

Objekt record

Kapitel „Konfigurierbare Variablen“ im Handbuch „Regelsprache“

Kapitel „Konfigurationsdatei“ im Handbuch „Entwicklungsumgebung“

C-Funktion DM_LoadProfile

2.58 .connect

Dieses Attribut definiert beim Objekt **application** den Zustand und die Eigenschaften der Verbindung zu einem Server-Prozess. Bei den Objekten **control** und **subcontrol** definiert es den Zustand der OLE-Verbindung.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string (application)	get, set	ja
boolean (control , subcontrol)		
<i>C</i>	<i>COBOL</i>	
Bezeichner: AT_connect	Bezeichner: AT-connect	
Datentyp: DT_string (application)	Datentyp: DT-string (application)	
Datentyp: DT_boolean (control , subcontrol)	Datentyp: DT-boolean (control , subcontrol)	
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	application, control, subcontrol	

Application

Dieses Attribut definiert, dass sich die Anwendung mit einem laufenden Server-Prozess in Verbindung setzen soll, der auf dem Host gestartet worden ist.

Beim TCP/IP-Protokoll gilt für den Attributwert die Syntax

```
"<host>:<portnumber>"
```

wobei entweder der Name des Hosts oder seine IP-Adresse angegeben werden kann.

Um die Verbindung über **SSL** durchzuführen kann bei der Spezifikation der Verbindung über **.connect** das Schema "ssl://" vorangestellt werden (Beispiel **.connect "ssl://myserver:0815";**).

Beispiel

Ohne SSL	Mit SSL
<pre>application Appl { .connect "localhost:4711"; }</pre>	<pre>application Appl { .connect "ssl://localhost:4711"; }</pre>

Anmerkungen

- » Die Attribute **.transport**, **.connect** und **.exec** können nur geändert werden, wenn **.active = false** ist.
- » Die Attribute **.connect** und **.exec** sind vom verwendeten Transport abhängig, d.h. zukünftige neue Arten einer Transportschicht können andere Arten des Verbindungsaufbaus enthalten.

- » Auch beim Attribut *.transport* kann das Schema "ssl://" angegeben werden. Wenn bei beiden Attributen ein Schema angegeben ist, muss es identisch sein. Ein einmal angegebenes Schema "ssl://" lässt sich nicht mehr abstellen.
- » Ab IDM-Version A.05.02.i unterstützt der DISTRIBUTED DIALOG MANAGER (DDM) das IPv6-Protokoll auf allen Architekturen, die IPv6 **nativ** unterstützen.

Control

Dieses Attribut definiert den Zustand der Verbindung zum OLE-Client bzw. zum OLE-Server. Nur wenn eine Verbindung existiert, können die Dienstleistungen des OLE-Servers in Anspruch genommen werden.

Bei der Programmierung eines Controls im Client-Modus ist zu berücksichtigen das mit dem Setzen von *.connect := true* keine Verbindung zum Server geschaffen wird und das auch mit der Abfrage auf *.connect = true* nicht überprüft werden kann. Das Verbinden und Überprüfen zu einem Server ist erst möglich, wenn das Objekt sichtbar ist. Das Setzen/Erfragen einer Property oder Aufruf von Methoden ist damit erst möglich wenn das Objekt sichtbar ist. Dieses kann mit Hilfe des Attributs *.real_visible = true* überprüft werden.

Subcontrol

Der Vorgabewert für das Attribut ist *false*, d.h. es besteht keine Verbindung zu einem OLE-Server. Wird das Attribut auf *true* umgesetzt, versucht das Subcontrol, eine Verbindung zu einem OLE-Server (normalerweise eine Collection) des zum Vaterobjekt gehörenden OLE-Servers zu bekommen. Der Bezeichner des Subcontrols hat dabei dem Attributnamen des zuerst genannten OLE-Servers zu entsprechen.

Im folgenden Beispiel wird die Documents-Collection des Word-OCX konnektiert:

```
dialog WordClient

window WnFenster
{
    .title "WordClient";

    on close { exit; }

    control CoWord
    {
        .mode      mode_client;
        .name      "Word.Application";
        .visible true;
        .connect true;

        child subcontrol Documents { }
    }
}
```

```
on dialog start
{
  CoWord.Visible := true;
  Documents.connect := true;
  print Documents.connect; // Sollwert true
}
```

Eine Verbindung kann nur dann erfolgreich aufgebaut werden, wenn der Vater realisiert und konnektiert ist und der Bezeichner des Subcontrols mit dem Namen des OLE-Server-Attributs übereinstimmt.

Wird das Attribut *.connect* auf *false* gesetzt, bricht die Verbindung zu dem konnektierten OLE-Server ab. Bei der gegenwärtigen Implementierung bleiben die Kinder eines Subcontrols dann jedoch trotzdem konnektiert!

Dies bedeutet auch: verliert das Vaterobjekt (egal ob Control oder Subcontrol) seine Verbindung wenn das Attribut bereits den Wert *true* hat, bleibt die Verbindung des Subcontrols trotzdem erhalten.

2.59 .constant

Mit Setzen dieses Attributes auf *true* kann eine Variable „schreibgeschützt“ werden um so eine konstante Variable zu bekommen. Ein Aufheben des Schreibschutzes ist danach nicht mehr möglich. Enthält die Variable allerdings eine Objekt-Referenz, so wird beim expliziten Zerstören (z.B. über die eingebaute Funktion **destroy()**) der Variablenwert zu *null*.

Definition

<i>Datentyp</i> <i>boolean</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_constant Datentyp: DT_boolean		<i>COBOL</i> Bezeichner: AT-constant Datentyp: DT-boolean
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> variable	

Das Attribut gibt es nur für das Objekt *variable* (siehe auch Handbuch „Regelsprache“).

Eleganter als die Benutzung dieses Attributes ist die Verwendung des Schlüsselwortes *constant* anstatt *variable* im statischen Definitionsteil eines Dialogs oder Moduls um direkt eine konstante Variable zu definieren.

Beispiel

```
dialog D
variable integer V := 123;
constant integer C := 456;
on dialog start
{
  C := V;    // Evaluationsfehler da C nicht veränderbar
  V := 234;  // OK
  V.constant := true;
  V := 345;  // Fehler - Variable nun schreibgeschützt
}
```

2.60 .content

Dieses Attribut beinhaltet einen String, der vom Benutzer in einen editierbaren Text eingegeben wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string, object [text]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_content	Bezeichner: AT-content
Datentyp: DT_string, DT_text	Datentyp: DT-string, DT-text

Klassifizierung
Textattribut

Ausnahme

Ist ein Edittext ein Kindobjekt eines Spinbox-Objekts, darf das Attribut *.content* nicht genutzt werden. Die Wertsetzung erfolgt ausschließlich über die entsprechenden Attribute des Spinbox-Objekts selbst.

Anmerkungen zum IDM für Motif

- » Das Umsetzen von *.content* oder *.navigable* an einem **Edittext** kann Einfluss darauf haben, ob in einem anderen Edittext der Cursor (Caret) durchgezogen gezeichnet wird. Zu beachten ist, dass bei einem Edittext mit *.navigable = false* unter Motif keine Eingaben möglich sind.
- » Der durchgezogene Cursor erlaubt keinen Rückschluss auf Navigationsreihenfolge oder Fokussierung eines **Edittexts**.

Hinweis für das Objekt Poptext/Combobox

Das Attribut *.content* ist am Poptext (Combobox) ausschließlich als skalaras Attribut verfügbar. Zum Zugriff auf den Inhaltsvektor des Poptextes ist *.text[integer]* zu benutzen!

Es dient folgenden zwei Funktionalitäten:

1. Vereinfachung des Zugriffs auf den Text des aktiven Eintrags
2. Zugriff auf den tatsächlichen Text im Eingabefeld

Eine Abfrage von *.content* liefert im Style Poptext (Attribut *.style*) immer den Text des aktiven Eintrags oder einen Fehler wenn kein Eintrag aktiv bzw. der Poptext leer ist. In den anderen Styles wird der aktuelle Inhalt des Editierbereichs zurückgegeben.

Eine Setzung von *.content* in den Styles listbox oder edittext bewirkt zusätzlich eine Setzung des *.activeitem*-Attributes auf den im *.text[]* gefundenen Eintrag oder auf 0.

Im Style poptext ist eine Setzung von *.content* auf jeden Fall zu vermeiden.

Siehe auch

Attribute `.content[integer]`, `.text[integer]`

2.61 .content[integer]

Mit dem Attribut *.content[integer]* kann die **listbox** aus den Regeln gefüllt werden. Mit Hilfe dieses Attributes ist es möglich, einen Text an die Stelle in der **listbox** zu setzen, die im Index angegeben wurde.

Hinweis

Dieses Attribut wird von Modellen oder Defaults nicht geerbt!

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string, object [text]</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_content	Bezeichner: AT-content
Datentyp: DT_string, DT_text	Datentyp: DT-string, DT-text

Klassifizierung
Textattribut

Siehe auch

Objekt *listbox*

Attribut *.content*

2.62 .content[index]

Mit Hilfe dieses Attributs kann jedes Feld [row, column] im **tablefield** verändert werden.

Beim **listview** definiert das Attribut die Listeneinträge.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string, object [text]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_content	Bezeichner: AT-content
Datentyp: DT_string, DT_text	Datentyp: DT-string, DT-text

Klassifizierung
Textattribut

tablefield

Mit Hilfe dieses Attributs kann jedes Feld [row, column] im Tablefield verändert werden. Mit diesem Attribut können auch die Überschriften und Zeilenköpfe verändert bzw. erfragt werden. Damit umfasst **.content** auch das Attribut **.field**.

Dieses Attribut wird nicht vom Modell auf seine Instanzen vererbt.

Anmerkung zum IDM für Motif

Das Motif-Tablefield ignoriert in den Tabellenzellen führende Zeilenumbrüche (ein oder mehrere \n am Anfang der Texte). Um dennoch Leerzeilen vor einem Text anzuzeigen, kann vor den Zeilenumbrüchen ein Leerzeichen in den Text eingefügt werden.

Beispiel: " \nXYZ" statt "\nXYZ".

listview

Das Attribut definiert die Listeneinträge des **listviews**.

Beim Schreibzugriff („set“) können Werte vom Datentyp **string** und **text**-Ressourcen angegeben werden. **text**-Ressourcen werden automatisch zu **string** gewandelt.

Lesezugriff („get“) liefert immer einen Wert vom Datentyp **string**.

Der Wertebereich des Index ist $[0, 1] \dots [rowcount, colcount]$, wobei die Werte mit Index $[0, C]$ als Standardwerte für nicht gesetzte Werte im Bereich $[1, C] \dots [rowcount, C]$ dienen.

Spalte 1 ist die Beschriftung des Listeneintrags, die in allen Ansichten angezeigt wird. Die anderen Spalten werden nur in der Detailansicht angezeigt.

Wenn für **.content[R, C]** kein Inhalt definiert wird und auch kein Standardwert **.content[0, C]** definiert ist, dann wird zwar nichts angezeigt, dennoch ist der Eintrag vorhanden.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

Siehe auch

Objekte listview, tablefield

2.63 .contentfunc

Dieses Attribut definiert die Funktion, die für das dynamische Nachladen des **Tablefields** aufgerufen werden soll.

Diese Funktion wird immer dann vom IDM aufgerufen, wenn neue Zeilen oder Spalten angezeigt werden sollen und diese nicht mehr im IDM geladen sind.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_contentfunc	Bezeichner: AT-contentfunc
Datentyp: DT_func	Datentyp: DT-func

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Siehe auch

Kapitel „Nachlade-Funktionen“ im Handbuch „C-Schnittstelle - Grundlagen“

2.64 .control

Mit Hilfe dieses Attributes kann das hierarchisch übergeordnete **control**-Objekt erfragt werden.

Definition

Datentyp
object

Zugriff
get

C
Bezeichner: AT_control
Datentyp: DT_object

COBOL
Bezeichner: AT-control
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.65 .count

Allgemein liefert dieses Attribut, die Anzahl von Elementen in Vektoren.

Beim **timer** definiert das Attribut wie oft er wiederholt werden soll. Beim Ereignisobjekt **thisevent** liefert es die Anzahl der *select*-Ereignisse des timers.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set get (function , rule , thisevent)	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_count Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-count Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> function , rule , thisevent , timer	

Mit Hilfe dieses Attributes wird definiert, wie oft der **timer** wiederholt werden soll. Ist hier 0 angegeben, wird er unendlich oft wiederholt.

Beim Ereignisobjekt **thisevent** wird mit *.count* die Anzahl der Timer-Ereignisse definiert (nur gültig bei *select*-Ereignis beim Timer).

Bei **benutzerdefinierten Attributen** beinhaltet dieses Attribut bei indizierten Attributen die aktuelle Größe des gegebenen Attributes (= 0 bei nicht-indizierten Attributen und bei *.shadow*-Attributen).

Die Abfrage bei benutzerdefinierten Attributen erfolgt durch Angabe des zu erfragenden Attributes als Index, z.B. *.count[<benutzerdefiniertes Attribut>]*.

Des weiteren kann mit *.count* ganz allgemein die **Länge von Vektoren** erfragt werden, z.B. *.count[record]*, *.count[child]*.

Bei **Funktionen** und **Regeln** dient dieses Attribut zum Abfragen der Anzahl der Parameter.

Siehe auch

Attribute *.shadowattr*, *.shadowindex*, *.shadowobject*, *.type*

Objekt timer

Kapitel „Ereignisobjekt thisevent“, „Funktionen (functions)“ und „Benannte Regeln (Unterprogramme)“ im Handbuch „Regelsprache“

2.66 .cursor

Dieses Attribut weist einem Objekt einen Cursor zu.

Beim **setup**-Objekt kann mit *.cursor* die Cursor-Variante (Nummer) erfragt und gesetzt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [cursor]	get, set	ja
<i>integer</i> (setup)		nein (setup)

C

Bezeichner: AT_cursor

Datentyp: DT_cursor

Datentyp: DT_integer (**setup**)

COBOL

Bezeichner: AT-cursor

Datentyp: DT-cursor

Datentyp: DT-integer (**setup**)

Klassifizierung

Standardattribut

Hinweis

In der MOTIF-Version ist *.cursor* für ein **Rechteck** nicht verfügbar.

2.67 .cursorname[integer]

Attribut des **Setup**-Objekts, das eine Liste mit den Namen der im WSI verfügbaren Cursor enthält.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_cursorname	Bezeichner: AT-cursorname
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Verfügbarkeit

Nur IDM FÜR QT, ab IDM-Version A.06.01.a.

IDM FÜR MOTIF & WINDOWS, Ab IDM-Version A.06.03.a

2.68 .curvalue

Dieses Attribut definiert die aktuelle Position bzw. den aktuell angezeigten Wert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_curvalue	Bezeichner: AT-curvalue
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	progressbar, scrollbar, spinbox

progressbar

Die aktuelle Position des Fortschritts.

Das Attribut *.curvalue* gibt die Position des Fortschrittsbalkens an. Außerdem wird aus diesem Wert die Prozentangabe berechnet, die als optionale Beschriftung angezeigt wird.

scrollbar

Die Position des Scrollbarsliders kann durch das Attribut *.curvalue* erfragt und gesetzt werden.

spinbox

Gibt den momentan angezeigten Wert der **spinbox** (eigentlich den im Kindobjekt angezeigten Wert) an.

2.69 .cut_pending

Dieses Attribut zeigt an, dass eine Cut-Operation noch nicht abgeschlossen ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_cut_pending	Bezeichner: AT-cut-pending
Datentyp: DT_boolean	Datentyp: DT-boolean

Klassifizierung
Standardattribut

Der Wert wird zwischen dem Auslösen der Operation und der Bearbeitung des Cut-Events auf *true* gesetzt. Solange *.cut_pending = true* ist, werden Veränderungen am Objekt abgelehnt. Damit wird verhindert, dass während einer laufenden Cut-Operation das betreffende Objekt manipuliert wird.

Soll in einer Regel ein gesperrtes Objekt verändert werden, kann man das Attribut auf *false* setzen. Dabei geht das Attribut *.cut_pending_changed* auf *true*, damit nachfolgende Regeln erkennen können, dass möglicherweise Attribute nicht mehr in dem Zustand sind, wie zum Beginn der Cut-Operation.

2.70 .cut_pending_changed

Dieses Attribut wird beim Auslösen der Cut-Operation auf *false* gesetzt und zeigt damit an, dass während der laufenden Cut-Operation das betreffende Objekt nicht manipuliert wurde. Dieses Attribut wird automatisch auf *true* gesetzt, sobald in einer Regel *.cut_pending := false* gesetzt wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>boolean</i>	get

C
Bezeichner: AT_cut_pending_changed
Datentyp: DT_boolean

COBOL
Bezeichner: AT-cut-pending-changed
Datentyp: DT-boolean

Klassifizierung
Standardattribut

2.71 .data

Dient zum Setzen und Abfragen der Daten eines DOM-Knotens. Das Attribut ist nur verfügbar, wenn der nodetype entweder *nodetype_cdata_section* oder *nodetype_processing_instruction* ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string, object [text]	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_data	Bezeichner: AT-data
Datentyp: DT_string, DT_text	Datentyp: DT-string, DT-text

Vererbung
nein

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	doccursor

Das Attribut ist für den XML-Cursor verfügbar, es ist nicht vererbbar. Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.72 .dataget[attribute]

Mit diesem Attribut wird die Kopplung eines View-Attributs, anzugeben als Index, zu einem Model-Attribut, anzugeben als Wert, für das Holen des Wertes von der Model-Komponente und das Repräsentieren durch die View-Komponente definiert.

Definition

<i>Datentyp</i> <i>attribute</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
-------------------------------------	----------------------------	---------------------------------

C
Bezeichner: AT_dataget
Datentyp: DT_attribute

COBOL
Bezeichner: AT-dataget
Datentyp: DT-attribute

Vererbung
ja

Klassifizierung
Datenmodellattribut

Ohne Index wird das klassenspezifische Standardattribut verwendet. Das Attribut ist an der View-Komponente zu definieren.

Der Datenaustausch erfolgt entsprechend den Synchronisationsregeln der beteiligten Model-Komponente und View-Komponente und ist nur aktiv, wenn auch ein Datenmodell im [.datamodel](#)-Attribut gesetzt ist.

Verfügbarkeit

Ab IDM-Version A.06.01.a

Siehe auch

Kapitel „Kopplung zwischen Model und View“ im Handbuch „Programmiertechniken“

2.73 .dataindex[attribute]

Dieses Attribut definiert zusätzlich zur Kopplung von Attributen zwischen Model-Komponente und View-Komponente einen Index der den Zugriff auf das Attribut weiter verfeinert um so die unterschiedlichen Relationsarten zu ermöglichen.

Als Index kann ein beliebiges View- oder Model-Attribut verwendet werden. Dieses Attribut ist an der View-Komponente zu setzen.

Definition

<i>Datentyp</i> <i>attribute</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
-------------------------------------	----------------------------	---------------------------------

<i>C</i> Bezeichner: AT_dataindex Datentyp: DT_attribute	<i>COBOL</i> Bezeichner: AT-dataindex Datentyp: DT-attribute
--	--

Vererbung
ja

Klassifizierung
Datenmodellattribut

Verfügbarkeit

Ab IDM-Version A.06.01.a

Siehe auch

Kapitel „Reihenfolge und Werteaggregation“ im Handbuch „Programmiertechniken“

2.74 .datamap[attribute]

Dieses Attribut definiert eine zusätzliche Abbildung von Attributen bei der Kopplung von Attributen zwischen View- und Model-Komponente. Diese Definition kann notwendig werden, wenn mehrere Datenmodelle ihre Datenwerte in ein einziges Attribut der View-Komponente schreiben sollen um so die Relationsart „Mischung“ zu erlauben.

Als Index kann ein beliebiges View- oder Model-Attribut verwendet werden. Dieses Attribut ist an der View-Komponente zu setzen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>attribute</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_datamap	Bezeichner: AT-datamap
Datentyp: DT_attribute	Datentyp: DT-attribute

Vererbung
ja

Klassifizierung
Datenmodellattribut

Verfügbarkeit

Ab IDM-Version A.06.01.a

Siehe auch

Kapitel „Reihenfolge und Werteaggregation“ im Handbuch „Programmiertechniken“

2.75 .datamodel[attribute]

Mit diesem Attribut kann die Kopplung des Objekts (View-Komponente) zu einem Datenmodell (Model-Komponente) definiert werden. Das Attribut ist an der View-Komponente zu definieren.

Als Index kann ein Attribut verwendet werden, welches das View-Attribut bezeichnet für das das Datenmodell verwendet werden soll. Ohne Index gilt das Datenmodell für alle View-Attribute.

Für eine aktive Kopplung ist mindesten die Definition eines Datenmodells ohne Index sowie eine Kopplung des View-Attributs mit einem Model-Attribut über *.dataget* oder *.dataset* nötig.

Vorrang vor der normalen Vererbung des Attributwertes hat noch die Überdeckung durch einen gesetzten Wert an dem am nächsten liegenden Vaterobjekt (ohne Dialog bzw. Modul).

Definition

<i>Datentyp</i> <i>object</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
----------------------------------	----------------------------	---------------------------------

C
Bezeichner: AT_datamodel
Datentyp: DT_object

COBOL
Bezeichner: AT-datamodel
Datentyp: DT-object

Vererbung
ja

Klassifizierung
Datenmodellattribut

Verfügbarkeit

Ab IDM-Version A.06.01.a

2.76 .dataoptions[enum]

Dieses Attribut definiert die automatische Synchronisation zwischen Model- und View-Komponente.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_dataoptions	Bezeichner: AT-dataoptions
Datentyp: DT_boolean	Datentyp: DT-boolean

Vererbung
ja

Klassifizierung
Datenmodellattribut

Attribut-Index	Standard	Komponente	Bedeutung
<i>dopt_represent_on_map</i>	true	View	Direkt vor der Sichtbarschaltung werden von den Model-Komponenten die Datenwerte geholt und dem View-Objekt gesetzt.
<i>dopt_represent_on_init</i>	false	View	In der Objektinitialisierung (:init-Methode) werden die Datenwerte geholt und am View-Objekt gesetzt.
<i>dopt_apply_on_unmap</i>	false	View	Direkt vor der Unsichtbarschaltung werden die Datenwerte vom View-Objekt geholt und den gekoppelten Model-Komponenten zugewiesen.
<i>dopt_apply_on_event</i>	false	View	Wird durch eine Benutzerinteraktion ein Dialogereignis erzeugt, welches auf eine mögliche Änderung eines View-Attributs hinweist, so wird dieses den gekoppelten Model-Komponenten zugewiesen.
<i>dopt_propagate_on_start</i>	true	Model	Beim Start des Dialogs bzw. Moduls werden die Daten der Model-Objekte an die gekoppelten View-Komponenten weitergegeben.

Attribut-Index	Standard	Komponente	Bedeutung
<i>dopt_propagate_on_changed</i>	true	Model	Änderungen an einem Model-Attribut werden an die gekoppelten View-Komponenten weitergereicht.
<i>dopt_cache_data</i>	true	Model	Dieser Indexwert ist nur für den doccursor verfügbar. true Die vom doccursor selektierten Daten werden für weitere Zugriffe zwischengespeichert („Caching“). false Der doccursor selektiert die Daten bei jedem Zugriff neu aus dem XML-Dokument.

Dieses Attribut ist an allen Objektklassen vorhanden, die benutzerdefinierte Attribute erlauben, aber unter Umständen nicht mit allen Indizes. Nur für visuelle Objekte stehen die Indizes *dopt_apply_on_unmap*, *dopt_represent_on_map* und *dopt_apply_on_event* zur Verfügung.

Verfügbarkeit

Ab IDM-Version A.06.01.a

Siehe auch

Kapitel „Synchronisierung zwischen Model und View“ im Handbuch „Programmiertechniken“

2.77 .dataselect[attribute]

Mit diesem Attribut werden gleichzeitig das als Index angegebene Datamodel-Attribut und ein als Wert zugewiesenes Selektionsmuster für Knoten eines XML-Dokuments definiert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_dataselect	Bezeichner: AT-dataselect
Datentyp: DT_string	Datentyp: DT-string

Vererbung
ja

<i>Klassifizierung</i>	<i>Objekte</i>
Datenmodellattribut	doccursor

Das definierte Datamodel-Attribut kann dann zur Kopplung mit einer View verwendet werden. Datenänderungen werden über das im Index angegebene Attribut des **Doccursors** weitergegeben. Das Selektionsmuster beschreibt die Knoten des XML-Dokuments aus denen die Daten geholt bzw. in denen sie gespeichert werden.

Die Syntax des Selektionsmusters entspricht der Musterdefinition für die **:select**-Methode des **Doccursors**.

Das Selektionsmuster des *.dataselect*-Attributs ohne Index definiert Knoten als Ausgangspunkte für die indizierten *.dataselect*-Attribute. Die Selektionsmuster der indizierten *.dataselect*-Attribute werden dann als relative IDM-Pfade interpretiert. Sie referenzieren Knoten innerhalb der Teilbäume, deren Wurzeln durch das nicht indizierte *.dataselect*-Attribut selektiert werden. Die indizierten *.dataselect*-Attribute greifen in diesem Fall also nicht auf das gesamte XML-Dokument zu, sondern nur auf die Unterknoten jener Knoten, die vom nicht indizierten *.dataselect*-Attribut selektiert worden sind.

Um eine konsistente Behandlung von optionalen XML-Elementen und -Attributen zu ermöglichen, gibt die Selektion eines Datamodel-Attributs einen Leerstring zurück, wenn ein Wert nicht vorhanden ist oder sich nicht innerhalb der vorselektierten Teilbäume befindet.

Mit dem Attribut *.dataselectattr* kann festgelegt werden, ob das Datamodel-Attribut mit dem Knoteninhalt oder einem Knotenattribut verknüpft ist. Datentyp und Kardinalität des Datamodel-Attributs können mit den Attributen *.dataselecttype* und *.dataselectcount* gesteuert werden.

Beispiel

In diesem Beispiel stammen die Daten für eine Liste mit Nobelpreisträgern aus einem XML-Dokument, in dem die Informationen zu jedem Preisträger in einem Knoten „prize“ stehen.

```

<?xml version="1.0"?>
<nobelprizes>
  <category id="p">Physics</category>
  <category id="c">Chemistry</category>
  <prize year="1" category="p">
    <merit>discovery of x-rays</merit>
    <winner>Wilhelm Conrad Röntgen</winner>
  </prize>
  <prize year="11" category="c">
    <merit>discovery of radioactivity</merit>
    <winner>Marie Curie</winner>
  </prize>
  <prize year="18" category="p">
    <merit>development of the concept of quanta</merit>
    <winner>Max Planck</winner>
  </prize>
  <prize year="70" category="c">
    <merit>discovery of sugar nucleotides and their role
      in the biosynthesis of carbohydrates</merit>
    <winner>Luis Leloir</winner>
  </prize>
</nobelprizes>

```

Die XML-Knoten können dann beispielsweise wie folgt über Datamodel-Attribute verknüpft werden:

```

...
document Doc
{
  doccursor DocCur
  {
    .dataselect "..prize";
    .dataselect[.Winner]   ".winner";
    .dataselect[.Year]     ".";
    .dataselect[.Discovery] "merit";
    .dataselectattr[.Year]  "year";
    .dataselecttype[.Year]  integer;
    .dataselectcount[.Year] integer;
  }
}
...

```

Verfügbarkeit

Ab IDM-Version A.06.01.b

Siehe auch

Attribute *.dataselectattr[attribute]*, *.dataselectcount[attribute]* und *.dataselecttype[attribute]*

2.78 .dataselectattr[attribute]

Dieses Attribut definiert, mit welchem Knotenattribut das als Index angegebene Datamodel-Attribut verknüpft ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_dataselectattr	Bezeichner: AT-dataselectattr
Datentyp: DT_string	Datentyp: DT-string

Vererbung
ja

<i>Klassifizierung</i>	<i>Objekte</i>
Datenmodellattribut	doccursor

Wenn *.dataselectattr* nicht gesetzt ist, dann ist das Datamodel-Attribut an den Knoteninhalt gekoppelt. Die Daten werden über das Attribut *.text* des ***Doccursors*** geholt und gesetzt.

Wenn *.dataselectattr* auf den Namen eines Knotenattributs gesetzt wird, dann ist das Datamodel-Attribut an die Attributwerte dieses Knotenattributs gekoppelt. Die Daten werden über das Attribut *.attribute* des ***Doccursors***, mit dem Namen des Knotenattributs als Index, geholt und gesetzt.

Verfügbarkeit

Ab IDM-Version A.06.01.b

Siehe auch

Attribut *.dataselect[attribute]*

2.79 .dataselectcount[attribute]

Dieses Attribut definiert die Kardinalität des als Index angegebenen Datamodel-Attributs.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>datatype</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_dataselectcount	Bezeichner: AT-dataselectcount
Datentyp: DT_type	Datentyp: DT-type

<i>Standardwert</i>	<i>Vererbung</i>
<i>integer</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Datenmodellattribut	doccursor

Wertebereich

integer (Standard)

Das Datamodel-Attribut ist ein *vector* und enthält die Inhalte bzw. Attributwerte aller Knoten, die dem Selektionsmuster des Attributs *.dataselect* entsprechen.

void

Das Datamodel-Attribut ist ein Skalar und enthält nur den Inhalt bzw. Attributwert des ersten Knotens, der dem Selektionsmuster des Attributs *.dataselect* entspricht.

Verfügbarkeit

Ab IDM-Version A.06.01.b

Siehe auch

Attribut *.dataselecttype[attribute]*

2.80 .dataselecttype[attribute]

Dieses Attribut definiert den Datentyp, in den die Werte des als Index angegebenen Datamodel-Attributs konvertiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>datatype</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_dataselecttype	Bezeichner: AT-dataselecttype
Datentyp: DT_type	Datentyp: DT-type

Vererbung
ja

<i>Klassifizierung</i>	<i>Objekte</i>
Datenmodellattribut	doccursor

Wenn *.dataselecttype* nicht angegeben ist, dann enthält das Datamodel-Attribut einen Vektor mit Strings (Datentyp *vector[string]*).

Schlägt die Konvertierung fehl, wird das Holen der Werte mit einem Fehler abgebrochen.

Beispiel

Im folgenden Dialogausschnitt werden die drei Datamodel-Attribute „.Name“, „.Female“ und „.Name3“ definiert.

```
dialog D
...
document Doc
{
  doccursor DocCur
  {
    .dataselect[.Name] "..person";

    .dataselect[.Name3]      "..person[.birthyear=\"1978\"] [3]";
    .dataselecttype[.Name3] string;

    .dataselect[.Female]      "..person";
    .dataselectattr[.Female]  "female";
    .dataselecttype[.Female]  boolean;
    .dataselectcount[.Female] integer;
  }
}
...
```

Das Attribut „.Name“ sammelt die Texte von allen „person“-Knoten in einem String-Vektor, während das Attribut „.Name3“ nur den Namen der 3. Person mit Geburtsjahr 1978 als Skalar vom Typ *string* enthält.

Beim Attribut „.Female“ werden über den Zugriff *DocCur.attribute["female"]* die entsprechenden Knotenattribute aller „person“-Knoten geholt, in *boolean* konvertiert und in einem *vector* gespeichert.

Verfügbarkeit

Ab IDM-Version A.06.01.b

Siehe auch

Attribut *.dataselectcount[attribute]*

2.81 .dataset[attribute]

Mit diesem Attribut wird die Kopplung eines View-Attributs, anzugeben als Index, zu einem Model-Attribut, anzugeben als Wert, für das Holen des Wertes von der View-Komponente und das Zuweisen an der Model-Komponente definiert.

Definition

<i>Datentyp</i> <i>attribute</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
-------------------------------------	----------------------------	---------------------------------

C
Bezeichner: AT_dataset
Datentyp: DT_attribute

COBOL
Bezeichner: AT-dataset
Datentyp: DT-attribute

Vererbung
ja

Klassifizierung
Datenmodellattribut

Ohne Index wird das klassenspezifische Standardattribut verwendet. Das Attribut ist an der View-Komponente zu definieren.

Der Datenaustausch erfolgt entsprechend den Synchronisationsregeln der beteiligten Model-Komponente und View-Komponente und ist nur aktiv, wenn auch ein Datenmodell im [.datamodel](#)-Attribut gesetzt ist.

Verfügbarkeit

Ab IDM-Version A.06.01.a

Siehe auch

Kapitel „Kopplung zwischen Model und View“ im Handbuch „Programmiertechniken“

2.82 .defbutton

Dieses Attribut definiert einen **pushbutton** als Default-Pushbutton.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
boolean (pushbutton)	get, set	ja
integer (messagebox)		
<i>C</i>	<i>COBOL</i>	
Bezeichner: AT_defbutton	Bezeichner: AT-defbutton	
Datentyp: DT_boolean (pushbutton)	Datentyp: DT-boolean (pushbutton)	
Datentyp: DT_integer (messagebox)	Datentyp: DT-integer (messagebox)	
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	messagebox, pushbutton	

Die Aufgabe dieses Default-Pushbuttons ist es, das entsprechende Objekt zu selektieren, wenn der Benutzer die **Return**-Taste drückt und sich der Fokus nicht auf einem mehrzeiligen editierbaren Text befindet.

Hinweis für Motif

Der **pushbutton** muss in einer Dialogbox liegen!

messagebox

Das Attribut **.defbutton** legt fest, welcher der drei Buttons der Defaultbutton ist. Ist der Wert ungültig, weil der Button nicht sichtbar ist, wird vom Fenstersystem ein Button als Defaultbutton ausgewählt.

Wenn dieses Attribut mehr als einem sichtbaren Pushbutton zugewiesen wurde, wählt der Dialog Manager selbsttätig den Default-Pushbutton aus. Definieren Sie deshalb bitte nur **einen** Pushbutton als Default.

Siehe auch

Attribut **.dialogbox**

2.83 .deltavalue

Dieses Attribut bestimmt den Differenzwert, um den der aktuelle Wert *.curvalue* der **spinbox** pro Pfeilselektion erhöht bzw. verringert werden soll.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_deltavalue
Datentyp: DT_integer

COBOL
Bezeichner: AT-deltavalue
Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	spinbox

Siehe auch

Attribut *.curvalue*

2.84 .depth

Über dieses Attribut wird das Aussehen des statischen Textes in einer Statusbar gesteuert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_depth	Bezeichner: AT-depth
Datentyp: DT_integer	Datentyp: DT-integer

Standardwert
0

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	statictext

Negative Werte bedeuten, dass der statische Text gegenüber der Statusbar vertieft dargestellt wird. Positive Werte bedeuten, dass er hervorgehoben dargestellt wird. Die Höhe des Wertes entspricht der Hervorhebung entsprechend für negative Werte.

Dieses Attribut hat nur Auswirkung, wenn der statische Text ein Kind einer Statusbar ist.

2.85 .dialog

Dieses Attribut definiert den Dialog zu einem Objekt.

Definition

Datentyp
object

Zugriff
get

C
Bezeichner: AT_dialog
Datentyp: DT_object

COBOL
Bezeichner: AT-dialog
Datentyp: DT-object

Klassifizierung
Standardattribut

2.86 .dialogbox

Das Attribut wandelt ein Fenster in eine sogenannte **Dialogbox**. Ist diese sichtbar, werden Eingaben für andere Fenster über Tastatur oder Maus deaktiviert. Erst wenn die Dialogbox geschlossen wurde, sind Einträge für andere Fenster möglich.

Wenn der Wert auf *true* gesetzt ist, so ist das angebundene Fenster eine Dialogbox. Dies bedeutet, dass alle anderen Fenster deaktiviert werden, sobald die Dialogbox auf dem Bildschirm erscheint. Erst wenn die Dialogbox geschlossen wurde, kann das andere aktive Fenster wieder bearbeitet werden.

Definition

<i>Datentyp</i> <i>boolean</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_dialogbox Datentyp: DT_boolean		<i>COBOL</i> Bezeichner: AT-dialogbox Datentyp: DT-boolean
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> window	

Hinweis zum IDM für Motif

Von MOTIF wird eine applikationsmodale Dialogbox erzeugt.

Hinweis zum IDM für Windows

Nur Toplevel-Fenster können Dialogboxen werden!

Dialogboxen werden – wie unter MICROSOFT WINDOWS üblich – mit einem breiten Rahmen dargestellt, falls sie nicht vergrößerbar sind (*.sizeable = false*).

Damit Fenster als Dialogboxen richtig funktionieren benötigen sie ein anderes Fenster der Anwendung, dem sie zugeordnet werden können. Aus diesem Grund sollte beim Öffnen einer Dialogbox immer ein anderes, sichtbares und sensitives, Fenster der Anwendung existieren. Dieses Fenster darf auch erst nach der Dialogbox geschlossen werden. Auch auf die richtige Schachtelung mehrerer Dialogboxen ist zu achten, das heißt, die zuletzt geöffnete Dialogbox muss als erste wieder geschlossen werden. Werden diese Regeln nicht beachtet kann es vorkommen, dass Fenster der Anwendung nicht automatisch wieder aktiviert werden oder dass Dialogboxen sogar hinter anderen Fenstern der Anwendung liegen.

Empfehlung

Wenn Fenster immer als direkte Folge von Benutzeraktionen geöffnet werden und das Schließen eines Fensters nur durch eine direkte Benutzeraktion an dem zu schließenden Fenster ausgelöst wird, werden die obigen Regeln nicht verletzt.

Siehe auch

Attribut *.defbutton*

2.87 .direction

Dieses Attribut bestimmt die Ausrichtung (Orientierung) des Objekts.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_direction Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-direction Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> layoutbox, notebook, progressbar, scrollbar, spinbox, splitbox, tablefield	

Wertebereich

- 1
Vertikale bzw. spaltenweise Ausrichtung.
- 2
Horizontale bzw. zeilenweise Ausrichtung.

layoutbox

Das *.direction*-Attribut steuert, wie die Kindobjekte in der **layoutbox** angeordnet werden.

Wertebereich

- 1
Die Kinder der **layoutbox** werden in der Reihenfolge, wie sie im Kindvektor stehen, **spaltenweise** angeordnet. Wenn in einer Spalte kein Platz mehr ist, d.h. das Objekt nicht mehr sichtbar wäre wenn es in derselben Spalte angeordnet werden würde, wird das nächste Kind in der nächsten Spalte positioniert. In jeder Spalte wird die Maximalbreite bestimmt. Anhand dieser werden die Kinder der **layoutbox** dann ausgerichtet.
- 2 (Standard)
Die Kinder der **layoutbox** werden in der Reihenfolge, wie sie im Kindvektor stehen, **zeilenweise** angeordnet. Wenn in einer Zeile kein Platz mehr ist, d.h. das Objekt nicht mehr sichtbar wäre wenn es in derselben Zeile angeordnet werden würde, wird das nächste Kind in die nächste Zeile positioniert. In jeder Zeile wird die Maximalhöhe bestimmt. Anhand dieser werden die Kinder der **layoutbox** dann ausgerichtet.

notebook (nur Motif)

Dieses Attribut definiert die Orientierung des **notebooks**, d.h. die Richtung, in der die Bindung verlaufen soll.

Wertebereich

1 (Standard)

Die Bindung verläuft **vertikal**. Abhängig vom *.backpage*-Attribut ist das **notebook** links oder rechts gebunden.

2

Die Bindung verläuft **horizontal**. Abhängig vom *.backpage*-Attribut ist das **notebook** oben oder unten gebunden.

Im folgenden werden die Auswirkungen der Attribute *.backpage* und *.direction* auf die Positionierung von Majortabs und Minortabs bzw. der Bindung aufgezeigt:

<i>.backpage</i>	<i>.direction</i>	Majortab	Minortab	Bindung
<i>bp_bottomright</i>	1	rechts	unten	links
<i>bp_bottomright</i>	2	unten	rechts	oben
<i>bp_bottomleft</i>	1	links	unten	rechts
<i>bp_bottomleft</i>	2	unten	links	oben
<i>bp_topright</i>	1	rechts	oben	links
<i>bp_topright</i>	2	oben	rechts	unten
<i>bp_topleft</i>	1	links	oben	rechts
<i>bp_topleft</i>	2	oben	links	unten

progressbar

Das *.direction*-Attribut definiert die Ausrichtung der Fortschrittsanzeige.

Wertebereich

1

Die Fortschrittsanzeige verläuft in vertikaler Richtung **von unten nach oben**.

2 (Standard)

Die Fortschrittsanzeige verläuft in horizontaler Richtung **von links nach rechts**.

scrollbar

Mit dem *.direction*-Attribut wird die Ausrichtung der **scrollbar** festgelegt.

Wertebereich

1 (Standard)

Vertikale **scrollbar**, bei der sich *.minvalue* oben und *.maxvalue* unten befindet.

2

Horizontale **scrollbar**, bei der sich *.minvalue* links und *.maxvalue* rechts befindet.

spinbox

Bei der **spinbox** bestimmt das *.direction*-Attribut, ob die Pfeile der Schaltflächen nach oben und unten oder nach links und rechts zeigen.

Wertebereich

1 (Standard)

Vertikale Anordnung der Schaltflächen mit Pfeilen nach oben und unten.

2

Horizontale Anordnung der Schaltflächen mit Pfeilen nach links und rechts.

Hinweis zum IDM FÜR QT

Das Attribut *.direction* und die horizontale Ausrichtung der Pfeile wird nicht unterstützt.

splitbox

Das *.direction*-Attribut bestimmt, ob die **splitbox** horizontal oder vertikal in Split-Bereiche unterteilt wird.

Wertebereich

1 (Standard)

Die Splitbars verlaufen vertikal, d.h. die **splitbox** ist horizontal in Split-Bereiche unterteilt und durch Verschieben der Splitbars kann die Breite der Split-Bereiche geändert werden.

2

Die Splitbars verlaufen horizontal, d.h. die **splitbox** ist vertikal in Split-Bereiche unterteilt und durch Verschieben der Splitbars kann die Höhe der Split-Bereiche geändert werden.

tablefield

Das *.direction*-Attribut definiert die Ausrichtung des **tablefields** und steuert ob Zeilen oder Spalten Vorrang haben. Das Attribut beeinflusst die Übernahme von Zeilen- und Spalten-Standardwerten, die Selektion und Navigation sowie bestimmte Methoden.

Wertebereich

1 (Standard)

Vertikale Ausrichtung mit Spaltenpriorität.

2

Horizontale Ausrichtung mit Zeilenpriorität.

Auswirkungen

Standardwerte

Bei *.direction* = 1 werden für nicht gesetzte Zellwerte (*.content[index]*, *.bgc[index]*, *.fgc[index]*...) die Spalten-Standardwerte übernommen, d.h. die Attributwerte mit den Indizes *[0,<C>]*. Bei

.*direction* = 2 werden die Zeilen-Standardwerte, d.h. die Attributwerte mit den Indizes [*<R>*, 0] übernommen.

Selektion

Wenn sowohl *.selection[sel_column]* als auch *.selection[sel_row]* auf *true* gesetzt sind und die Zelle [*<R>*, *<C>*] ausgewählt wird, dann wird bei *.direction* = 1 die Spalte *<C>* und bei *.direction* = 2 die Zeile *<R>* selektiert.

In einem **tablefield** mit Mehrfachselektion sucht *.nextactive[index]* bei *.direction* = 1 zeilenweise und bei *.direction* = 2 spaltenweise nach der nächsten selektierten Zelle. Sind beispielsweise die Zellen [4,2] und [3,5] selektiert, liefert *.nextactive[1,1]* bei *.direction* = 1 als Ergebnis [3,5] und bei *.direction* = 2 als Ergebnis [4,2].

Navigation

Bei *.direction* = 1 wird der Eingabefokus beim Drücken von **Return** in die nächste selektierbare Zelle rechts von der aktuellen Zelle bewegt. **Pos 1** und **Ende** springen an den Anfang bzw. das Ende der Zeile mit der aktuell fokussierten Zelle.

Bei *.direction* = 2 wird der Eingabefokus beim Drücken von **Return** in die nächste selektierbare Zelle unter der aktuellen Zelle bewegt. **Pos 1** und **Ende** springen an den Anfang bzw. das Ende der Spalte mit der aktuell fokussierten Zelle.

:find()-Methode

Das *.direction*-Attribut beeinflusst die Suchrichtung der **:find()**-Methode. Bei *.direction* = 1 wird das **tablefield** zeilenweise durchsucht, bei *.direction* = 2 spaltenweise. Befindet sich der gesuchte Wert beispielsweise in den Zellen [4,2] und [3,5], dann liefert **:find()** ohne Angabe eines Suchbereichs bei *.direction* = 1 die Fundstelle [3,5] und bei *.direction* = 2 die Fundstelle [4,2].

Methoden :clear(), :delete(), :exchange(), :insert() und :move()

Das Attribut *.direction* steuert, zusammen mit dem optionalen *Direction*-Parameter der Methoden, ob die Methoden auf Zeilen oder Spalten angewendet werden. Wenn der *Direction*-Parameter nicht angegeben ist, werden die Methoden bei *.direction* = 1 auf Zeilen und bei *.direction* = 2 auf Spalten angewendet.

2.88 .directory

Dieses Attribut beinhaltet den Pfad des initialen Verzeichnisses für die Datei- oder Verzeichnisauswahl (**filereq**), wenn diese mit der eingebauten Funktion **querybox()** geöffnet wird.

Hat das Attribut den Wert "" oder einen ungültigen Pfadnamen, so wird das Arbeitsverzeichnis der Anwendung als initiales Verzeichnis beim Aufruf mit **querybox()** erscheinen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_directory	Bezeichner: AT-directory
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	filereq

Es ist zu beachten, dass die Pfadangaben der Notation des Systems zu entsprechen haben. Für MICROSOFT WINDOWS ist dies <drive>:\<dir 1>\...\<dir n> und für UNIX /<dir 1>/../<dir n>. Dabei ist zu beachten, dass „\“ innerhalb von Strings in IDM-Dateien als „\\“ zu schreiben sind.

Bei gesetztem *.changedir*-Attribute enthält dieses Attribut nach erfolgreicher Selektion das Verzeichnis in dem eine Datei bzw. ein Verzeichnis selektiert wurde.

Besonderheiten Microsoft Windows

Ist am **filereq**-Objekt das Attribute *.style* auf *fr_directory* gesetzt, definiert das *.directory*-Attribut die oberste Pfadangabe. Das Attribut *.directory* beschränkt die Auswahl bei *.style = fr_directory* allerdings nur dann, wenn das Attribut *.changedir* den Wert *false* besitzt. Ansonsten (*.changedir = true*) ist die Verzeichnisauswahl unbeschränkt und das Verzeichnis, welches im Attribut *.directory* angegeben ist, wird vorselektiert.

Der Attributwert "" wird als „Arbeitsplatz“ interpretiert und nicht als „Arbeitsverzeichnis“.

Siehe auch

Attribut *.changedir*

Eingebaute Funktion **querybox()**

2.89 .display

Dieses Attribut definiert den Screen in welchem das Fenster darzustellen ist.

Bei Angabe von *null* oder einer **display**-Ressource mit „ungültiger“ Screen-Nummer bzw. bei IDM-Versionen ohne Multiscreen-Support erscheint das Fenster im Default-Screen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object (display)	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_display	Bezeichner: AT-display
Datentyp: DT_display	Datentyp: DT-display

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Eine **messagebox** oder ein **filereq**, die über `querybox()` mit einem Fenster als Vater geöffnet werden, erscheinen im Screen des Vater-Fensters, ansonsten im Default-Screen.

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

2.90 .doccursor[integer]

Über das *.doccursor*-Attribut kann auf den XML-Cursor des XML-Dokuments zugegriffen werden. Das Attribut wird mit dem Objektindex indiziert. Der erste Kindcursor besitzt den Index 1.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> (<i>doccursor</i>)	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_doccursor	Bezeichner: AT-doccursor
Datentyp: DT_doccursor	Datentyp: DT-doccursor

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	document

2.91 .dock_line

Mit Hilfe dieses Attributs kann die Reihenfolge mehrerer Toolbars in einem Fenster angegeben werden. Diese Toolbars werden dann aufsteigend sortiert angelegt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_dock_line	Bezeichner: AT-dock-line
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	toolbar

2.92 .dock_offset

Das Attribut gibt bei horizontal angedockten Toolbars den Abstand zum linken Rand des Vaterfensters an, bei vertikal angedockten Toolbars den Abstand zum oberen Fensterrand.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_dock_offset	Bezeichner: AT-dock-offset
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	toolbar

Es können auch negative Werte verwendet werden, was zum Beispiel bei eingeschaltetem Positionsraster auch notwendig ist, um die Toolbar genau an den Fensterrand zu platzieren (*.dock_offset* = -1).

Bei Toolwindows wird das Attribut ignoriert.

2.93 .dockable[enum]

Mit diesem Attribut kann gesteuert werden, an welchen Stellen sich eine **toolbar** befinden darf.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_dockable	Bezeichner: AT-dockable
Datentyp: DT_boolean	Datentyp: DT-boolean

Standardwert
true

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	toolbar

Wertebereich

true (Standard)

Die **toolbar** kann die als Index angegebene Andockposition einnehmen.

false

Die **toolbar** kann die als Index angegebene Andockposition nicht einnehmen.

Indexbereich

dock_window

Abdocken der **toolbar** als frei bewegliches Tool-Window.

dock_up

Eindocken der **toolbar** im Symbolleistenbereich über dem Client-Bereich des Fensters.

dock_down

Eindocken der **toolbar** im Symbolleistenbereich unter dem Client-Bereich des Fensters.

dock_left

Eindocken der **toolbar** im Symbolleistenbereich links vom Client-Bereich des Fensters.

dock_right

Eindocken der **toolbar** im Symbolleistenbereich rechts vom Client-Bereich des Fensters.

Es muss immer mindestens ein Element auf **true** gesetzt sein. Außerdem kann das dem aktuellen Zustand der **Toolbar** entsprechende Element bei Instanzen nicht auf **false** umgesetzt werden. Die folgende Zuweisung würde bei einer Instanz also eine Fehlermeldung im Tracefile erzeugen:

```
this.dockable[this.docking] := false;
```

Ebenfalls erlaubt sind Zuweisungen ohne Indexangabe:

```
this.dockable := true;
```

In diesem Fall werden alle Elemente auf **true** gesetzt.

```
this.dockable := false;
```

Dies setzt das Attribut für alle Indizes mit Ausnahme von *this.dockable[this.docking]* auf *false*.

Siehe auch

Attribut *.docking*

2.94 .docking

Mit diesem Attribut wird die Toolbar umgedockt.

Definition

<i>Datentyp</i> <i>enum</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
--------------------------------	----------------------------	---------------------------------

<i>C</i> Bezeichner: AT_docking Datentyp: DT_enum	<i>COBOL</i> Bezeichner: AT-docking Datentyp: DT-enum
---	---

Standardwert
dock_window

<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> toolbar
---	---------------------------

Wertebereich

dock_window
Abdocken der **toolbar** als frei bewegliches Tool-Window.

dock_up
Eindocken der **toolbar** im Symbolleistenbereich über dem Client-Bereich des Fensters.

dock_down
Eindocken der **toolbar** im Symbolleistenbereich unter dem Client-Bereich des Fensters.

dock_left
Eindocken der **toolbar** im Symbolleistenbereich links vom Client-Bereich des Fensters.

dock_right
Eindocken der **toolbar** im Symbolleistenbereich rechts vom Client-Bereich des Fensters.

Bei einer Instanz kann dem Attribut nur ein Wert zugewiesen werden, für den das *.dockable[enum]*-Attribut *true* ist.

Ist das Vaterfenster einer Toolbar-Instanz sichtbar und nicht sizeable, kann das Attribut nur geändert werden, wenn sich das *.height*-Attribut des Vaterfensters dadurch nicht ändern würde, d.h. es darf dann nur von oben nach unten oder von links nach rechts und umgekehrt umgedockt werden.

Bei einem unsichtbaren Vaterfenster ist diese Einschränkung nicht gegeben, da der Anwendungsprogrammierer auf eine Größenänderung reagieren kann, ohne dass es dabei zu unschönen optischen Effekten kommt.

Siehe auch

Attribut *.dockable[enum]*

2.95 .document[integer]

Über das *.document*-Attribut kann auf die XML-Dokumente zugegriffen werden. Das Attribut wird mit dem Objektindex indiziert. Das erste Kinddokument besitzt den Index *1*.

Das Attribut ist für alle Objekte verfügbar, bei denen das *.record*-Attribut verfügbar ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object (document)	get, set	nein

C
Bezeichner: AT_document
Datentyp: DT_document

COBOL
Bezeichner: AT-document
Datentyp: DT-document

Klassifizierung
Standardattribut

2.96 .editable

Für die Objekte **tablefield** und **edittext** sind Attribute verfügbar, die dem Dialogdesigner die Möglichkeit zu geben, das Aussehen und Eingabeverhalten beeinflussen zu können. Dies betrifft die Darstellung von nicht selektierbaren Edittexten sowie die Definition, ob der Benutzer mit diesen Objekten interagieren können soll oder nicht:

» **Bedienbar- oder Selektierbarkeit**

Attribut: *sensitive*

Die Eigenschaft, ob ein Objekt vom Benutzer selektiert werden kann oder nicht. Nur wenn ein Objekt selektierbar ist, kann der Benutzer die für das Objekt üblichen Aktionen auf das Objekt anwenden, z.B. Daten eingeben und Scrollen.

» **Editierbarkeit**

Attribut: *editable*

Die Eigenschaft, dass der Benutzer den Inhalt des Objektes ändern kann; er ist also in der Lage, den Inhalt zu editieren und zu verändern.

» **Fokussierbarkeit**

Attribut: *navigable*

Die Eigenschaft, dass das Objekt in der Tastatursteuerung erfasst wird und damit den Eingabefokus erhalten kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_editable	Bezeichner: AT-editable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	edittext, tablefield

Diese drei Attribute beeinflussen nun das Verhalten des Objektes in folgender Weise:

Attribut	Wert	Auswirkung
<i>.sensitive</i>	<i>true</i>	Objekt kann vom Benutzer selektiert werden. Nur wenn das Attribut <i>.sensitive</i> den Wert <i>true</i> hat, können die Attribute <i>.navigable</i> und <i>.editable</i> wirksam werden, ansonsten werden sie ignoriert.
<i>.sensitive</i>	<i>false</i>	Objekt kann nicht vom Benutzer selektiert werden. Es sind also keinerlei Aktionen auf dem Objekt möglich. Der Inhalt des Objekts wird wie für die Fenstersysteme üblich gegraut dargestellt.

Attribut	Wert	Auswirkung
<i>.editable</i>	<i>true</i>	Benutzer kann den Inhalt des Objektes ändern, falls er das Objekt selektieren kann.
<i>.editable</i>	<i>false</i>	Benutzer kann den Inhalt des Objektes nicht verändern.
<i>.navigable</i>	<i>true</i>	Objekt ist in der normalen Tastatursteuerung enthalten, d.h., der Benutzer kann das Objekt durch Navigation im zugehörigen Fenster erreichen.
<i>.navigable</i>	<i>false</i>	Das Objekt kann nicht über die Tastatursteuerung erreicht werden. Das Objekt kann jedoch trotzdem über die Maus den Fokus erhalten.

2.97 .editable[index]

Für das Objekt **tablefield** ist das Attribut *.editable* indiziert für jede Zelle verfügbar. Damit kann die Editierbarkeit jeder Zelle in der Tabelle einzeln gesteuert werden. Dabei überschreibt das Attribut an der Zelle den Wert an der gesamten Tabelle.

Definition

<i>Datentyp</i> <i>boolean</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_editable Datentyp: DT_boolean		<i>COBOL</i> Bezeichner: AT-editable Datentyp: DT-boolean
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield	

2.98 .editpos

Mit Hilfe dieses Attributes kann definiert werden, wie und wo das **tablefield** editiert werden kann.

Ist *.editpos true*, soll immer auf der Stelle des Feldes editiert werden, das gerade bearbeitet wird. Dabei wird die Breite des assoziierten editierbaren Textes immer automatisch der Spaltenbreite angepasst.

Ist *.editpos false*, soll die im assoziierten editierbaren Text definierte Position immer beibehalten werden.

Definition

<i>Datentyp</i> <i>boolean</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_editpos Datentyp: DT_boolean		<i>COBOL</i> Bezeichner: AT-editpos Datentyp: DT-boolean
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield	

2.99 .edittext

Mit Hilfe dieses Attributes kann der Identifikator des zu dem Tablefield zugehörigen editierbaren Textes definiert werden. Dieser editierbare Text muss denselben Vater wie das Tablefield haben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [<i>edittext</i>]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_edittext	Bezeichner: AT-edittext
Datentyp: DT_object	Datentyp: DT-object

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.100 .edittype

Mit Hilfe dieses Attributes wird beschrieben, wie der zu dem Tablefield gehörende editierbare Text und der Inhalt des Tablefields zusammenarbeiten.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
enum	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_edittype	Bezeichner: AT-edittype
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Wertebereich

edit_locking

Die Eingabe in den editierbaren Text wird erst nach dem Betätigen der **Return**-Taste in das Tablefield übernommen. In diesem Fall ist jede weitere Selektion im Tablefield verboten, wenn ein Element gerade bearbeitet wird und noch nicht mit der **Return**-Taste abgeschlossen wurde. Die Sperrung beginnt dabei nach der ersten Veränderung des Inhaltes des editierbaren Textfeldes. Mit Eingabe von **Escape** kann diese Sperre ohne jede Änderung des Tablefieldinhaltes gelöst werden.

edit_offline

Die Eingabe in den editierbaren Text wird erst nach der Deselektion des editierbaren Textes übernommen. Dabei wird das Tablefield aber nicht gesperrt. Wird in dem Tablefield direkt auf ein anderes Feld geklickt, so wird der editierbare Text deselektiert.

edit_online

Jede Eingabe in den editierbaren Text wird sofort in dem zugehörigen Element des Tablefields angezeigt.

2.101 .endsel

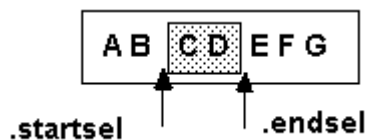
Das Attribut *.endsel* definiert das Ende der Markierung (selektierter Bereich) im Eingabefeld.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_endsel	Bezeichner: AT-endsel
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	edittext, poptext



Beim Verändern von *.endsel* bzw. *.startsel* muss *.focus = true* sein, damit die neue Selektion dargestellt werden kann. Dabei darf *.endsel* auch kleiner als *.startsel* sein.

Besondere Werte

Folgende Werte, die insbesondere bei gesetzten Formaten sinnvoll sind, haben eine Sonderstellung:

- 1
Markierung einschließlich aller Nachkommastellen (auch per Format eingefügter Nullen, die nicht im Inhaltsstring enthalten sind)
- 2
Markierung aller Vorkommastellen bis zum rechten Ende (Komma)
- 3
Markierung aller Vorkommastellen einschließlich per Format gesetzter führender Nullen

Eine Markierung des Vorzeichens ist in keinem Fall möglich.

Anmerkung zum IDM für Windows

Beim Abfragen von *.startsel* und *.endsel* des **edittextes** kann *.endsel* ab IDM-Version A.05.02.I auch kleiner sein als *.startsel*. Der Cursor befindet sich in diesem Fall links von der Markierung. Wenn allerdings der Anwender Text mit der Maus markiert, ist *.startsel* immer kleiner als *.endsel* und es kann nicht erkannt werden, wo sich der Cursor befindet. In vorherigen IDM-Versionen war *.startsel* immer kleiner oder gleich *.endsel*, außer es wurde von der Regelsprache aus anders gesetzt.

Siehe auch

Attribut *.startsel*

2.102 .env[string]

Über das Setup-Objekt lassen sich die Umgebungsvariablen des Programms setzen. Dazu existieren zwei Attribute: `.env[]` und `.envvar[]`.

In `.env[]` sind alle Umgebungsvariablen enthalten, wobei die über die Option **-IDMenv** gesetzten Werte die in der Umgebung gesetzten Werte überschreiben.

Die mittels der Option **-IDMenv** gesetzten Variablen sind nur für IDM-Funktionen gültig. Über das Attribut `.envvar[]` hingegen erhält man nur die Umgebungsvariablen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<code>string</code>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_env	Bezeichner: AT-env
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Die Attribute `.env[]` und `.envvar[]` sind setzbar und abfragbar. Die Indizierung erfolgt über den Namen der Umgebungsvariablen. Beim Erfragen erhält man ein `fail` wenn die Umgebungsvariable nicht gesetzt ist.

Weiter ist es möglich sich alle mittels **-IDMenv** gesetzten Variablen anzusehen. Dazu dient das Attribut `.count[]` indiziert über `.env`, welches als Rückgabewert die Anzahl der IDM-Umgebungsvariablen liefert. Nun kann `.env[]` über den Zahlenbereich `1 ... .count[.env]` indiziert werden, um die Namen der Umgebungsvariablen zu ermitteln.

2.103 .envvar[string]

Über das Setup-Objekt lassen sich die Umgebungsvariablen des Programms setzen. Dazu existieren zwei Attribute: *.env[]* und *.envvar[]*.

In *.env[]* sind alle Umgebungsvariablen enthalten, wobei die über die Option **-IDMenv** gesetzten Werte die in der Umgebung gesetzten Werte überschreiben.

Die mittels der Option **-IDMenv** gesetzten Variablen sind nur für IDM-Funktionen gültig. Über das Attribut *.envvar[]* hingegen erhält man nur die Umgebungsvariablen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_envvar	Bezeichner: AT-envvar
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Die Attribute *.env[]* und *.envvar[]* sind setzbar und abfragbar. Die Indizierung erfolgt über den Namen der Umgebungsvariablen. Beim Erfragen erhält man ein *fail* wenn die Umgebungsvariable nicht gesetzt ist.

Weiter ist es möglich sich alle mittels **-IDMenv** gesetzten Variablen anzusehen. Dazu dient das Attribut *.count[]* indiziert über *.env*, welches als Rückgabewert die Anzahl der IDM-Umgebungsvariablen liefert. Nun kann *.env[]* über den Zahlenbereich *1count[.env]* indiziert werden, um die Namen der Umgebungsvariablen zu ermitteln.

2.104 .errfile

Mit Hilfe dieses Attributs kann während der Laufzeit über das **Setup**-Objekt der absolute Pfad der **Fehlerdatei** (Umleitung von IDM-Fehlermeldungen in eine Datei) abgefragt werden.

Definition

<i>Datentyp</i> <i>string</i>	<i>Zugriff</i> get	
<i>C</i> Bezeichner: AT_errfile Datentyp: DT_string		<i>COBOL</i> Bezeichner: AT-errfile Datentyp: DT-string
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> setup	

2.105 .errorcode

Das Attribut zeigt die Fehlerart an, welche bei der Aktivierung oder Deaktivierung bzw. der Nutzung von Applikationsfunktionalitäten aufgetreten ist. Zurückgesetzt wird dieses Attribut intern bei der Aktivierung einer Applikation über das *.active*-Attribut.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>enum</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_errorcode	Bezeichner: AT-errorcode
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Wertebereich

error_file

Datei-Fehler, der z.B. beim Laden einer DYNLIB-Bibliothek auftritt.

error_network

Fehler ist beim Aufruf von Netzwerk-Systemroutinen aufgetreten. Insofern ist eine klare Unterscheidung zu DDM-Protokoll-Fehlern möglich.

error_none

Aktivierung oder Deaktivierung der Applikation hat fehlerlos funktioniert.

error_protocol

Nicht reparabler Fehler im DDM-Protokoll ist aufgetreten. Sinnvollerweise sollte ein Kunde in diesem Fall den IDM-Support benachrichtigen. Dieser Fehler wird nicht zwangsläufig von beiden Seiten signalisiert sondern nur von der Seite, die ihn zuerst bemerkt. Da bei einem Protokollfehler die Netzwerkverbindung geschlossen wird, bekommt die andere Seite dies normalerweise als Netzwerkfehler gemeldet.

error_unavail

Applikations-Funktionalität nicht verfügbar – d.h. NDX- oder DYNLIB-Erweiterung ist nicht eingebunden.

error_version

DDM-Protokoll-Version zwischen Client- und Server ist inkompatibel.

Zu beachten ist hierzu auch das Attribut *.systemerror*. Grundsätzlich ist *.systemerror* nur bei den Fehlerarten *error_network* und *error_file* belegt.

Siehe auch

Attribut *.systemerror*

2.106 .event[integer]

Eine Regel kann von mehreren Ereignissen gleichzeitig ausgelöst werden. Alle diese Ereignisse sind im (indizierten) Attribut *.event[integer]* enthalten.

Mit *.eventcount* kann die Anzahl der Ereignisse ermittelt werden.

Wird *.event* mit einem Integer-Wert indiziert, erhält man ein gleichzeitig aufgetretenes Ereignis.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	
<i>anyvalue</i>	get	
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_event		Bezeichner: AT-event
Datentyp: DT_event		Datentyp: DT-event
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	<i>thisevent</i>	

.event[event]

Wird *.event* mit einem Wert vom Typ *event* indiziert, ist das Ergebnis vom Typ *boolean* und *true*, falls das Index-Ereignis eines der aufgetretenen Ereignisse ist.

Als Index kann jegliches Ereignis eingetragen werden, so kann z.B. erfragt werden, ob ein spezielles Ereignis vorliegt, z.B. *.event[select]*.

Siehe auch

Kapitel „Ereignisobjekt *thisevent*“ im Handbuch „Regelsprache“

2.107 .event_code

Mit Hilfe dieses Attributes kann beim Ereignisobjekt **thisevent** der Code des externen Ereignisses abgefragt werden (gültig nur beim *extevent*-Ereignis).

Definition

Datentyp
anyvalue (definiert durch den
Identifikator des externen Ereignisses)

Zugriff
get

C
Bezeichner: AT_event_code
Datentyp: DT_anyvalue

COBOL
Bezeichner: AT-event-code
Datentyp: DT-anyvalue

Klassifizierung
Objektspezifisches Attribut

Objekte
thisevent

Siehe auch

Kapitel „Externe Ereignisse“ und „Ereignisobjekt thisevent“ im Handbuch „Regelsprache“

2.108 .eventcount

Eine Regel kann von mehreren Ereignissen gleichzeitig ausgelöst werden. Alle diese Ereignisse sind im (indizierten) Attribut *.event[integer]* enthalten.

Mit *.eventcount* kann die Anzahl der Ereignisse ermittelt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_eventcount	Bezeichner: AT-eventcount
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	<i>thisevent</i>

Siehe auch

Kapitel „Ereignisobjekt thisevent“ im Handbuch „Regelsprache“

2.109 .exec

Dieses Attribut des **application**-Objekts definiert, dass die Anwendung einen Prozess auf dem Host starten soll, der durch den Pfad gegeben ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string	get, set	ja

C	COBOL
Bezeichner: AT_exec	Bezeichner: AT-exec
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Beim TCP/IP-Protokoll gilt für den Attributwert die Syntax

```
"<host>[%<username>[%<password>]]:<path>"
```

wobei entweder der Name des Hosts oder seine IP-Adresse angegeben werden kann.

Um die Anwendungsseite über **SSH** an Stelle von RSH zu starten, kann bei der Spezifikation des Kommandos das Schema "ssh://" vorangestellt werden (Beispiel .exec "ssh://myserver:list";).

Standardmäßig wird das RSH-Protokoll verwendet, wenn kein Sicherheitsschema beim .transport-Attribut angegeben wurde. Andernfalls wird das SSH-Protokoll verwendet. Soll von diesem Standard abgewichen werden, muss das gewünschte Schema beim .exec-Attribut angegeben werden. Hierbei wird mit "ssh://" das SSH-Protokoll und mit "rsh://" das RSH-Protokoll ausgewählt.

Das Schema "ssh://" unterstützt auch das OpenSSH Kommando. Es wird zuerst gesucht, ob die libssh DLL verfügbar ist. Ist diese nicht verfügbar wird überprüft, ob ein ssh Kommando aufrufbar ist. Zusätzlich gibt es das Schema "sshlib://" um nur libssh zu verwenden und "sshcmd://" um nur OpenSSH zu verwenden. Der Nachteil des Kommandos ist, dass kein Passwort verwendet werden kann. OpenSSH muss so konfiguriert sein, dass eine Verbindung ohne Passwort möglich ist (siehe Manpages von ssh).

Beispiel

Ohne SSH	Mit SSH
<pre>application App11 { .exec "host%account%passwd:list"; }</pre>	<pre>application App11 { .exec "ssh://host%account%passwd:list"; }</pre>

Ohne SSH	Mit SSH und SSL
<pre>application App12 { .exec "host%account%passwd:list"; }</pre>	<pre>application App12 { .transport "ssl"; .exec "host%account%passwd:list"; }</pre>

Anmerkungen

- » Die Attribute *.transport*, *.connect* und *.exec* können nur geändert werden, wenn *.active = false* ist.
- » Die Attribute *.connect* und *.exec* sind vom verwendeten Transport abhängig, d.h. zukünftige neue Arten einer Transportschicht können andere Arten des Verbindungsaufbaues enthalten.
- » Um einen Doppelpunkt vor dem syntaktisch vorgeschriebenen Doppelpunkt zu verwenden (im Name oder Passwort) muss dieser verdoppelt werden. Im Kommandoteil des *.exec*-Attributes (nach dem syntaktisch vorgeschriebenen Doppelpunkt) werden Doppelpunkte direkt übernommen.
- » Die für SSL notwendigen zusätzlichen Startoptionen müssen beim Kommando angegeben werden. Nur das Sicherheitsschema „ssl“ wird – falls verwendet – dem Kommando als zusätzliche Startoption `-IDMtransport ssl` hinzugefügt.
- » Die Kommandozeile erhält automatisch die Startoption `-IDMtel1port`. Wird als Kommando ein Skript oder ähnliches an Stelle einer IDM-Serveranwendung angegeben, dann muss von der Ausgabe der IDM-Serveranwendung zumindest die Zeile, die mit `-IDMport` beginnt, komplett (inklusive führender und folgender Zeilenumbrüche „\n“) weitergeleitet werden.
- » Ab IDM-Version A.05.02.i unterstützt der DISTRIBUTED DIALOG MANAGER (DDM) das IPv6-Protokoll auf allen Architekturen, die IPv6 **nativ** unterstützen.

Siehe auch

Objekt application

Handbuch „Distributed Dialog Manager (DDM)“

2.110 export

Mit *export* werden Objekte und benannte Regeln eines Moduls nach außen bekannt gemacht, sodass in einem importierenden Modul bzw. Dialog darauf zugegriffen werden kann.

Im Gegensatz zu den anderen Attributen steht die Eigenschaft *export* am Beginn der eigentlichen Objektdefinition.

Definition

```
export <Objektklasse> <Bezeichner>
{
  <weitere Objektdefinitionen wie Attribute, Kindobjekte...>
}
```

Anmerkung

An Kindobjekten, die ihre Export-Eigenschaft durch *reexport* geerbt haben, wird *export* ignoriert.

Siehe auch

Attribut *reexport*

Kapitel „Modularisierung“ im Handbuch „Programmiertechniken“

2.111 .extension

Dieses Attribut erlaubt die Definition eines Dateisuffixes beim Dateiauswahldialog (**Filerequestor**).

Bei erfolgter Selektion durch den **querybox()**-Aufruf wird, wenn die Datei noch keinen Suffix besitzt, ein „.“ und dieser Attributwert angehängt. Bei einem Attributwert von "" wird kein Suffix angehängt.

Definition

<i>Datentyp</i> string	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_extension Datentyp: DT_string		<i>COBOL</i> Bezeichner: AT-extension Datentyp: DT-string
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> filereq	

Besonderheiten Microsoft Windows

Extensions werden nur auf Dateien angewendet.

Siehe auch

Attribute *.style*, [value](#)

Eingebaute Funktion `querybox()`

2.112 .external

Liefert *true* falls die Klasse des Objekts eine „externe“ USW-Klasse ist, andernfalls *false*.

Das Attribut ist an allen Objektklassen verfügbar.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>boolean</i>	get

C
Bezeichner: AT_external
Datentyp: DT_boolean

COBOL
Bezeichner: AT-external
Datentyp: DT-boolean

Klassifizierung
Standardattribut

2.113 .external[integer]

Liefert die registrierte „externe“ USW-Objektklasse an der Position / zurück. Durch Iteration zwischen dem Index 1 ... *count[.external]* können alle externen Klassen ermittelt werden.

Das Attribut ist an allen Objektklassen verfügbar.

Definition

Datentyp

class, void

Zugriff

get

C

Bezeichner: AT_external

Datentyp: DT_class

COBOL

Bezeichner: AT-external

Datentyp: DT-class

Klassifizierung

Standardattribut

2.114 .extevent

Mit Hilfe dieses Attributes kann ein Applikations-Objekt so eingestellt werden, dass im Netz auch unter MICROSOFT WINDOWS asynchrone Ereignisse an den Client geschickt werden können. Wird es auf *true* gesetzt, wird zwischen Applikations- und Bildschirmseite ein zweiter Kommunikationskanal aufgebaut, der zum Transport asynchroner externer Ereignisse über das Netz verwendet wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_extevent	Bezeichner: AT-extevent
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Es sollte beachtet werden, dass der Aufbau eines solchen zweiten Kommunikationskanals ein „teurer“ Vorgang ist, der sowohl Zeit als auch Systemressourcen verbraucht. Aus diesem Grund ist das *.extevent*-Attribut defaultmäßig mit *false* belegt.

Siehe auch

Handbuch „Distributed Dialog Manager (DDM)“

2.115 .face

Mit Hilfe dieses Attributes kann die Schriftbild / Schrägstellung eines Fonts definiert werden.

This attribute can be used to define the typeface / slant of a font.

Anmerkung:

Es ist zu beachten, dass es vom gewählten Zeichensatz abhängt ob und in welcher Umfang die ausgewählten Modifikatoren angewendet werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
enum	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_face	Bezeichner: AT-face
Datentyp: DT_enum	Datentyp: DT-enum

Klassifizierung
Layoutattribut

Wertebereich

face_default
Normale, unveränderte Zeichendarstellung.

face_italic
Kursive Zeichendarstellung.
Im Unterschied zu *face_oblique* werden meist spezielle, kursive Zeichen verwendet.

face_roman
Aufrechte, gerade Zeichendarstellung (wird unter MICROSOFT WINDOWS ignoriert).

face_oblique
Geneigte, schräge Zeichendarstellung.
Im Unterschied zu *face_italic* werden die geneigten Zeichendarstellungen meist aus den regulären Zeichen abgeleitet.
face_oblique entspricht unter MICROSOFT WINDOWS *face_italic*.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.116 .fgc

Das Attribut *.fgc* definiert die Vordergrundfarbe eines Objektes.

Definition

<i>Datentyp</i> <i>object</i> [color]	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
---	----------------------------	-------------------------------

<i>C</i> Bezeichner: AT_fg Datentyp: DT_color	<i>COBOL</i> Bezeichner: AT-fgc Datentyp: DT-color
---	--

Klassifizierung
Layoutattribut

Anmerkungen

- » *.fgc* wird beim **poptext** (Combobox) in der Version für Microsoft Windows folgendermaßen unterstützt:
 - » Der geschlossene poptext, d.h. der entsprechende angezeigte Eintrag, besitzt Vorder- und Hintergrundfarbe.
 - » Der aufgeklappte poptext, d.h. die aufgeklappte Box besitzt eine Defaultfarbe.
- » *.fgc* wird bei **menubox** in der Version für Microsoft Windows ignoriert.
- » *.fgc* wird bei **menuitem**, **menusep**, **messagebox** und **scrollbar** in der Version für Microsoft Windows ignoriert.
- » Für *.fgc* wird bei **pushbuttons** in der Version für Microsoft Windows immer die systemweite Defaultfarbe verwendet.

2.117 .fgc[index]

Dieses Attribut definiert die Vordergrundfarbe eines einzelnen Feldes.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_fg	Bezeichner: AT-fgc
Datentyp: DT_color	Datentyp: DT-color

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Anmerkung

Das Attribut *.fgc[index]* wird im Header ignoriert, wenn *.colheadfg* und/oder *.rowheadfg* gesetzt sind.

2.118 .field[index]

Mit Hilfe dieses Attributs kann jedes einzelne Textfeld [row, column] im Inneren des **Tablefields** gesetzt werden. Damit können aber keine Überschriften und Zeilenköpfe geändert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string, object [text]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_field	Bezeichner: AT-field
Datentyp: DT_string, DT_text	Datentyp: DT-string, DT-text

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Hinweis zum IDM für Motif

Das Motif-Tablefield ignoriert in den Tabellenzellen führende Zeilenumbrüche (ein oder mehrere \n am Anfang der Texte). Um dennoch Leerzeilen vor einem Text anzuzeigen, kann vor den Zeilenumbrüchen ein Leerzeichen in den Text eingefügt werden.

Beispiel

" \nXYZ" statt "\nXYZ".

Siehe auch

Attribut .content[index]

2.119 .fieldactive[index]

Mit Hilfe dieses Attributes kann jedes einzelne Textfeld [row, column] im Tablefieldinneren auf aktiv bzw. nicht aktiv gesetzt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_fieldactive	Bezeichner: AT-fieldactive
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.120 .fieldfocus

Mit Hilfe dieses Attributes kann im **Tablefield**-Inneren (d.h. ohne Header) ein Feld (*[row,column]*), das gerade den Fokus hat, definiert bzw. abgefragt werden (im Gegensatz zu *.focus[l,J]* – hier werden auch die Header beachtet).

Definition

<i>Datentyp</i> <i>index</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_fieldfocus Datentyp: DT_index		<i>COBOL</i> Bezeichner: AT-fieldfocus Datentyp: DT-index
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield	

Beispiel

```
Var := Tf.fieldfocus;  
// Var => index
```

2.121 .fieldfocus[index]

Mit Hilfe dieses Attributes kann im **tablefield**-Inneren (d.h. ohne Header) ein Feld (*[row,column]*), das gerade den Fokus hat, definiert bzw. abgefragt werden (im Gegensatz zu *.focus[index]* – hier werden auch die Header beachtet).

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_fieldfocus	Bezeichner: AT-fieldfocus
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Beispiel

```
Var := Tf.fieldfocus[2,2];  
// Var => boolean
```

2.122 .fieldfocusable

Mit Hilfe dieses Attributes kann definiert werden, ob die Cursorsteuerung auch nicht sensitive Elemente im Tablefield berücksichtigen soll oder nicht. Dieses Attribut wird nur dann ausgewertet, wenn *.sensitive* von diesem Feld *false* ist.

Definition

<i>Datentyp</i> <i>boolean</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_fieldfocusable Datentyp: DT_boolean		<i>COBOL</i> Bezeichner: AT-fieldfocusable Datentyp: DT-boolean
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield	

2.123 .fieldshadow

Dieses Attribut definiert, ob die sensitiven Felder eines **Tablefields** mit einem Schatten dargestellt werden sollen oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_fieldshadow	Bezeichner: AT-fieldshadow
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Anmerkung

Bitte beachten Sie, dass dieses Attribut die interne Größe des Tablefields beeinflusst!

2.124 .filled

Das Attribut *.filled* definiert, ob das Rechteck mit der Hintergrundfarbe des Objektes ausgefüllt wird oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_filled	Bezeichner: AT-filled
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	rectangle

Anmerkung zum IDM für Windows

Rechtecke mit dem Attribut *.filled = false* werden nicht unterstützt. Diese werden mit der Hintergrundfarbe des Vaters gefüllt und sind nur am Rand selektierbar.

2.125 .firstchar

Das Attribut *.firstchar* bestimmt das erste angezeigte Zeichen und damit die horizontale Scroll-Position.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_firstchar	Bezeichner: AT-firstchar
Datentyp: DT_integer	Datentyp: DT-integer

<i>Standardwert</i>	<i>Vererbung</i>
1	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, treeview

Das Attribut *.firstchar* gibt die Nummer des ersten dargestellten Zeichens zurück oder setzt diese. Dadurch kann mit dem Attribut die horizontale Scroll-Position der **Listbox** bzw. des **Treeviews** abgefragt und festgelegt werden.

Der Wertebereich umfasst Integer-Werte > 0 .

Beim **Treeview** bezieht sich *.firstchar* auf Einträge der ersten Hierarchieebene.

Hinweis

Das tatsächlich als Erstes angezeigte Zeichen hängt von mehreren Faktoren ab, unter anderem ob eine proportionale oder nichtproportionale („monospaced“) Schriftart verwendet wird, ob und in welcher Größe Bilder zu den Einträgen angezeigt werden (*.picture[integer]*-Attribut) sowie beim **Treeview** zusätzlich vom *.style[enum]*-Attribut.

2.126 .firstchild

Das Attribut *.firstchild* greift auf das erste Kind eines Objektes zu.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

C
Bezeichner: AT_firstchild
Datentyp: DT_object

COBOL
Bezeichner: AT-firstchild
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.127 .firstmenu

Dieses Attribut greift auf das erste Kind in der Menühierarchie zu.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_firstmenu	Bezeichner: AT-firstmenu
Datentyp: DT_object	Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.128 .firstrecord

Dieses Attribut greift auf das erste Kind in der Record-Hierarchie zu.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_firstrecord	Bezeichner: AT-firstrecord
Datentyp: DT_record	Datentyp: DT-record

Klassifizierung
Hierarchieattribut

2.129 .firstsubcontrol

Dieses Attribut greift auf das erste **Subcontrol** eines **Control**- oder **Subcontrol**-Objekts zu.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

C
Bezeichner: AT_firstsubcontrol
Datentyp: DT_object

COBOL
Bezeichner: AT-firstsubcontrol
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.130 .firsttoolbar

Dieses Attribut greift auf die erste Toolbar im Fenster zu.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

C
Bezeichner: AT_firsttoolbar
Datentyp: DT_object

COBOL
Bezeichner: AT-firsttoolbar
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.131 .focus

In einem Dialog gibt es meistens genau ein Objekt, das den Tastatur-Fokus haben kann. Es wird in der gewohnten Weise vom darunterliegenden Fenstersystem angezeigt.

.focus ist ein Attribut mit einigen speziellen Eigenschaften:

- » Es kann nicht von einer Vorlage oder einem Default vererbt werden.
- » Es kann beim Definieren des Objektes nicht angezeigt werden.
- » Der Datentyp des Attributes hängt von der Objektart und der Zugriffsweise ab.
- » SetVal: Nur das Setzen des Fokus ist möglich, die Zurückstellung wird vom IDM automatisch ausgeführt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_focus	Bezeichner: AT-focus
Datentyp: DT_boolean	Datentyp: DT-boolean

Vererbung
nein

Klassifizierung
Standardattribut

SetValue

Das Objekt muss *real_visible* und *real_sensitive* sein.

Wenn das zugehörige Fenster das aktive Fenster der Anwendung ist, wird der Fokus gesetzt und angezeigt.

Wenn das zugehörige Fenster nicht das aktive Fenster der Anwendung ist, wird der Fokus nur intern gesetzt und als „savedfocus“ gespeichert.

Wenn ein Fenster deaktiviert oder wieder aktiviert wird, wird der Fokus wieder auf das Objekt gesetzt, das den Fokus zuletzt hatte, z.B. wird der Fokus in Bezugnahme auf das Fenster immer als „saved-focus“ gespeichert. Dies gilt jedoch nur solange das Fenster im Fenstersystem existiert.

Der Fokus kann nur gesetzt werden, das Zurücksetzen wird vom Dialog Manager automatisch ausgeführt.

Der Datentyp für normale Objekte ist *boolean*. Es wird nur *true* akzeptiert, *false* wird als Fehler zurückgewiesen.

Aus diesem Grund sind die folgenden Ausdrücke in der Regelsyntax zugelassen:

```
Lb.focus := true;
Pb.focus := true;
Wn.focus := true;
```

Objekt	Verhalten für .focus
Canvas	Der Fokus wird auf die entsprechende Canvas gesetzt und die zugehörige Canvas-Funktion CCR_focus aufgerufen. Wenn die Canvas gegenwärtig nicht im sichtbaren Bereich ihrer Eltern ist, werden die notwendigen Scroll-Operationen ausgeführt. Siehe auch Kapitel „Bedeutung des Attribut .options“ beim Objekt Canvas in der „Objektreferenz“
Fenster	Der Fokus wird auf das erste Objekt gesetzt (im Fenster), das <i>real_sensitive</i> und <i>real_visible</i> ist. Die Suche beginnt beim ersten Kind des Fensters, geht weiter zu den Kindern des ersten Kindes und schließlich zu den Brüdern des ersten Kindes. Wenn ein Objekt gefunden wird, das den Fokus haben kann, wird der Fokus auf dieses Objekt gesetzt. Wenn kein Objekt gefunden wurde, wird der Fokus auf das Fenster gesetzt.
Groupbox	Der Fokus wird auf das erste Objekt (in der Groupbox), das <i>real_sensitive</i> und <i>real_visible</i> ist gesetzt. Wenn ein Objekt gefunden wird, das den Fokus haben kann, wird der Fokus auf dieses Objekt gesetzt. Wenn kein Objekt gefunden wurde, wird der Fokus nicht vom aktuellen Fokus-Objekt übernommen, d.h. der Fokus bleibt unverändert.
Checkbox	Der Fokus wird auf das gegebene Objekt gesetzt und in der dem Objekt spezifischen Weise angezeigt. Wenn das Objekt gegenwärtig im Bereich seiner Eltern nicht sichtbar ist, werden die Scroll-Operationen ausgeführt, die notwendig sind, um das Objekt sichtbar zu machen.
Editierbarer Text	
Listbox	
Poptext	
Pushbutton	
Radiobutton	
Rechteck*	
Scrollbar	
Statischer Text	
Menübox	ungültig
Menüeintrag	ungültig

Objekt	Verhalten für .focus
Tablefield	Der Fokus wird auf das entsprechende Tablefield gesetzt. Um den Fokus auf bestimmte Felder zu setzen, muss für das Attribut .focus ein doppelter Index angegeben werden ([row, column]), der die Koordinaten des Feldes bestimmt: .focus [I,J]

*** Anmerkung**

Das DM-Objekt **Rechteck** kann unter Motif keinen Fokus erhalten, daher kann der Fokus nicht auf dieses Objekt gesetzt werden. Beim Erfragen des Fokus für ein Rechteck wird false zurückgegeben.

GetValue

.focus kann für ein beliebiges Objekt angefordert werden. Die Rückgabe ist jedoch von der Objektklasse abhängig.

Objekt	Datentyp	Rückgabewert
Groupbox	DM_ID	Kind (von gegebenem Objekt), das den Fokus hat
Fenster		
Canvas	boolean	<i>true</i> , wenn das angeforderte Objekt den Fokus hat. <i>false</i> , wenn das angeforderte Objekt den Fokus nicht hat.
Checkbox		
Editierbarer Text		
Listbox		
Poptext		
Pushbutton		
Radiobutton*		
Rechteck		
Scrollbar		
Statischer Text		
Tablefield	index	Index (s.o.), falls das Tablefield den Fokus hat; dann wird auf das durch den Index bestimmte Feld zugegriffen.

*** Anmerkung zum IDM für Windows**

Ein Radiobutton wird aktiviert, wenn er den Fokus erhält.

Siehe auch

Attribut .focusitem

2.132 .focus_on_click

Dieses Attribut legt fest, ob ein Mausklick in den Clientbereich das Objekt aktiviert, also den Fokus auf dieses Objekt setzt, oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_focus_on_click	Bezeichner: AT-focus-on-click
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	groupbox, image, rectangle, statictext, toolbar

Wertebereich

true

Ein Mausklick aktiviert das Objekt und fokussiert das Objekt oder eines seiner Kinder.

false

Ein Mausklick aktiviert das Objekt nicht und der Fokus bleibt unverändert.

Das Attribut dient dazu eine Toolbar definieren zu können, welche beim Anklicken per Maus den Fokus nicht aus dem zugehörigen Fenster herausnimmt. Allerdings ist dabei zu beachten, dass eine Groupbox mit *.focus_on_click false* keine Elemente enthält, welche einen Fokus bekommen können oder sogar zu Eingabezwecken benötigen (z.B. editierbarer Text).

Das Attribut kann nur unter MICROSOFT WINDOWS genutzt werden.

2.133 .focusable

Gibt an, ob ein Objekt den Fokus erhalten kann oder nicht.

Achtung

Veraltet, wird nicht mehr unterstützt. Zugriff (lesend oder schreibend) erzeugt ein `fail`.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_focusable	Bezeichner: AT-focusable
Datentyp: DT_boolean	Datentyp: DT-boolean

Klassifizierung
Objektspezifisches Attribut

2.134 .focusitem

Zusätzlich zu *.focus* existiert für das Objekt **listbox** ein spezielles Element, das *.focusitem*. Mit Hilfe dieses Attributes kann der Fokus auf einen bestimmten Eintrag (z.B. den vierten), positioniert werden. Die Verfügbarkeit hängt dabei in erster Linie von dem jeweilig darunterliegenden Fenstersystem ab.

Achtung

Veraltet, wird nicht mehr unterstützt. Zugriff (lesend oder schreibend) erzeugt ein `fail`.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_focusitem	Bezeichner: AT-focusitem
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox

SetValue

Objekt	Verhalten von .focusitem
listbox	Der Fokus wird auf den gegebenen Eintrag gesetzt, wenn die Listbox den Fokus hat. Falls ein nicht vorhandener Index angegeben ist, wird der Fokus auf den letzten Eintrag gesetzt.

GetValue

Objekt	Datentyp	Rückgabewert
listbox	<i>integer</i>	Besitzt ein Objekt den Fokus, wird das aktuelle Fokusitem zurückgegeben; im anderen Fall wird 0 zurückgegeben.

Siehe auch

Attribut *.focus*

2.135 .font

Mit Hilfe dieses Attributes wird der zu dem Objekt gehörende Zeichensatz festgelegt.

Mit *.font* kann beim **setup**-Objekt die Zeichensatz-Variante (Nummer) erfragt und gesetzt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [font]	get, set	ja
<i>integer</i> (setup)		nein (setup)

C

Bezeichner: AT_font

Datentyp: DT_font

Datentyp: DT_integer (**setup**)

COBOL

Bezeichner: AT-font

Datentyp: DT-font

Datentyp: DT-integer (**setup**)

Klassifizierung

Standardattribut

Hinweis

.font wird bei **menubox**, **menuitem** und **messagebox** in der Version für Microsoft Windows ignoriert.

2.136 .font[index]

Dieses Attribut definiert den Zeichensatz eines einzelnen Feldes.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [font]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_font	Bezeichner: AT-font
Datentyp: DT_font	Datentyp: DT-font

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Anmerkung

Das Attribut *.font[index]* wird im Header ignoriert wenn *.colheadfont* und/oder *.rowheadfont* gesetzt sind.

2.137 .fontname[integer]

Attribut des **Setup**-Objekts, das eine Liste mit den Namen der im WSI verfügbaren Schriftarten enthält.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_fontname	Bezeichner: AT-fontname
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Verfügbarkeit

Nur IDM FÜR QT, ab IDM-Version A.06.01.a.

IDM FÜR MOTIF & WINDOWS, Ab IDM-Version A.06.03.a

2.138 .format

Dieses Attribut weist einem Objekt ein Format zu.

Beim **datetime**-Objekt definiert dieses Attribut das Anzeigeformat.

Beim **setup**-Objekt kann mit *.format* die Format-Variante (Nummer) erfragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string, object [format]</i>	get, set	ja
<i>string, object [text] (datetime)</i>	get, set (setup)	
<i>integer (setup)</i>		
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_format		Bezeichner: AT-format
Datentyp: DT_string, DT_format		Datentyp: DT-string, DT-format
Datentyp: DT_string , DT_text (datetime)		Datentyp: DT-string, Dt-text (datetime)
Datentyp: DT_integer (setup)		Datentyp: DT-integer (setup)

Klassifizierung

Textattribut

Als Format ist entweder ein String oder ein Identifikator erlaubt:

Bei Identifikator muss es eine Format-Ressource sein.

Eine Format-Ressource darf nur angegeben werden, wenn nicht schon das Attribut *.formatfunc* gesetzt ist (vgl. hierzu auch das Kapitel „Format-Funktionen“ im Handbuch „C-Schnittstelle - Grundlagen“).

datetime

Dieses Attribut definiert das Anzeigeformat des **datetime**-Objekts.

Beim Schreibzugriff („set“) können Werte vom Datentyp *string* und **text**-Ressourcen angegeben werden. **text**-Ressourcen werden automatisch zu *string* gewandelt.

Lesezugriff („get“) liefert immer einen Wert vom Datentyp *string*.

Wertebereich

<i>""</i>	kurzes Datumsformat (wie " <i>widget:d</i> ")
<i>widget:d</i>	Systemformat: kurzes Datumsformat
<i>widget:dd</i>	Systemformat: kurzes Datumsformat mit 4-stelliger Jahreszahl
<i>widget:ddd</i>	Systemformat: langes Datumsformat
<i>widget:t</i>	Systemformat: Zeitformat
<i><format_string></i>	Formatstring zur Definition des Anzeigeformats

Systemformate

Die **Systemformate** sind abhängig von den Sprach- und Regionseinstellungen des Betriebssystems. Sie bestimmen das Anzeigeformat jeweils vollständig und sind weder miteinander noch mit Formatstrings kombinierbar. Es kann zum Beispiel keine Beschriftung zu einem Systemformat hinzugefügt werden.

Formatstrings

Formatstrings bestehen aus Formatierungszeichen, die definieren, welche Teile eines Datums angezeigt werden und wie diese Teile angezeigt werden.

Tabelle 2: *Formatierungszeichen des datetime-Objekts*

Formatierungszeichen	Anzeige
<i>d</i>	Tag mit so vielen Stellen wie notwendig
<i>dd</i>	Tag mit 2 Stellen, gegebenenfalls mit führender „0“
<i>ddd</i>	Name des Wochentags (sprachabhängig), abgekürzt auf 3 Zeichen
<i>dddd</i>	vollständiger Name des Wochentags (sprachabhängig)
<i>h</i>	Stunde im 12-Stundenformat mit so vielen Stellen wie notwendig
<i>hh</i>	Stunde im 12-Stundenformat mit 2 Stellen, gegebenenfalls mit führender „0“
<i>H</i>	Stunde im 24-Stundenformat mit so vielen Stellen wie notwendig
<i>HH</i>	Stunde im 24-Stundenformat mit 2 Stellen, gegebenenfalls mit führender „0“
<i>m</i>	Minute mit so vielen Stellen wie notwendig
<i>mm</i>	Minute mit 2 Stellen, gegebenenfalls mit führender „0“
<i>M</i>	Monat mit so vielen Stellen wie notwendig
<i>MM</i>	Monat mit 2 Stellen, gegebenenfalls mit führender „0“
<i>MMM</i>	Monatsname (sprachabhängig), abgekürzt auf 3 Zeichen
<i>MMMM</i>	vollständiger Monatsname (sprachabhängig)
<i>s</i>	Sekunde mit so vielen Stellen wie notwendig
<i>ss</i>	Sekunde mit 2 Stellen, gegebenenfalls mit führender „0“

Formatierungszeichen	Anzeige
<i>t</i>	Vormittags- bzw. Nachmittagskennzeichen (sprachabhängig, im Englischen „AM“ und „PM“) mit einem Zeichen; Verwendung wird nicht empfohlen
<i>tt</i>	Vormittags- bzw. Nachmittagskennzeichen (sprachabhängig, im Englischen „AM“ und „PM“) mit 2 Zeichen
<i>yy</i>	Jahr mit 2 Stellen
<i>yyyy</i>	Jahr mit 4 Stellen

Andere Zeichen als die genannten Formatierungszeichen werden in die Anzeige übernommen. Zeichen, die angezeigt werden sollen, können optional in einfache Anführungszeichen (') eingeschlossen werden. Sie müssen in einfache Anführungszeichen eingeschlossen werden, wenn sie einem Formatierungszeichen entsprechen. Um ein einfaches Anführungszeichen anzuzeigen, muss es im Formatstring verdoppelt werden (").

Beispiel

Der Formatstring *'Heute ist 'dddd', der 'dd'. 'MMM yyyy* ergibt folgende Anzeige: „Heute ist Freitag, der 04. Jul 2014“.

Hinweise

- » Bei einem fehlerhaften Formatstring wird das systemabhängige Standardformat verwendet (entspricht *.format = ""*).
- » Die für *ddd*, *dddd*, *MMM*, *MMMM*, *t* und *tt* angezeigten Namen und Kennzeichen werden durch die Sprach- und Regionseinstellungen des Systems bestimmt.
- » Bei langen Anzeigetexten kann es sein, dass auf der Schaltfläche zum Öffnen des Kalenders kein Kalendersymbol mehr dargestellt wird.

setup

Mit *.format* kann beim **Setup**-Objekt die Format-Variante (Nummer) erfragt werden.

Hinweis zur Abfrage des Attributs *.format* mit *DM_GetVectorValue*

Es ist zu beachten, dass **DM_GetVectorValue()** bei der Abfrage von *.format* die Formatdefinition nur bei Verwendung von Format-Ressourcen zurückliefern kann. Bei Formatdefinitionen per Formatstring oder Formatfunktion wird Null zurückgeliefert. Daher wird die Nutzung von Formatressourcen empfohlen.

Mit den Optionen *DMF_GetMasterString* und *DMF_GetLocalString* erhält man einen Formatstring wie bei einer Formatdefinition per Formatstring.

Damit entsprechen die zurück gelieferten Werte der Funktion **DM_GetVectorValue** denen der Funktion **DM_GetValueIndex** bzw. eines **getvalue** in der Regelsprache.

2.139 .format[index]

Mit Hilfe dieses Attributes kann das Format für die einzelnen Felder [row, column] definiert werden.

.format[0,0] ist dabei der Defaultwert, der für alle anderen genommen wird, wenn nichts weiter angegeben ist.

Definition

<i>Datentyp</i> string, object [format]	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
---	----------------------------	-------------------------------

<i>C</i> Bezeichner: AT_format Datentyp: DT_string, DT_format	<i>COBOL</i> Bezeichner: AT-format Datentyp: DT-string, DT-format
---	---

<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield
---	------------------------------

Siehe auch

Attribut *.format*

2.140 .formatfunc

Die Format-Funktion interpretiert den Formatstring im Attribut *.format*. Ist bei einem editierbaren Text oder im Tablefield das Attribut *.format* auf eine Format-Ressource gesetzt, kann *.formatfunc* nicht mehr gesetzt werden.

Definition

<i>Datentyp</i> <i>object</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
----------------------------------	----------------------------	-------------------------------

<i>C</i> Bezeichner: AT_formatfunc Datentyp: DT_func	<i>COBOL</i> Bezeichner: AT-formatfunc Datentyp: DT-func
--	--

<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> edittext, tablefield
---	--

Hinweis

.formatfunc kann nur in einem einzeiligen editierbaren Text und im Tablefield verwendet werden, d.h. im anderen Fall wird das Attribut ignoriert!

Siehe auch

Kapitel „Format-Funktionen“ im Handbuch „C-Schnittstelle - Grundlagen“

C-Funktion DM_FmtDefaultProc

2.141 .function

Mit diesem Attribut wird die zu dem Objekt gehörende Funktion definiert. Die Funktion muss vom Anwendungsprogrammierer definiert werden. Sie wird dann zu den bei der Funktionsdefinition angegebenen Zeitpunkten vom DM aufgerufen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_function	Bezeichner: AT-function
Datentyp: DT_func	Datentyp: DT-func

Klassifizierung
Standardattribut

Siehe auch

Kapitel „Callback-Funktion“ im Handbuch „Regelsprache“

Kapitel „Objektcallback-Funktionen“ im Handbuch „C-Schnittstelle - Grundlagen“

2.142 .gradient

Dieses Attribut bestimmt Art und Parameter eines Farbverlaufs bei einer **color**-Ressource.

Verfügbarkeit

Nur IDM FÜR QT

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_gradient	Bezeichner: AT-gradient
Datentyp: DT_string	Datentyp: DT-string

Klassifizierung
Objektspezifisches Attribut

Objekte
color

Der Wert des Attributs ist ein String in der Form "<kind> [, <arg>]".

Der Parameter <kind> definiert die Verlaufsart. Es darf nur eine Verlaufsart angegeben werden.

Die weiteren Parameter können Folgendes enthalten:

- » Ergänzungsparameter zur Verlaufsart
- » Stopp-Punkt-Definitionen
- » Farbdefinitionen

Diese **Verlaufsarten** stehen zur Verfügung:

Tabelle 3: Arten von Farbverläufen

Verlauf	Definition	Erklärung
linear	"Linear, ..." "LinearV, ..."	vertikaler Verlauf
	"LinearH, ..."	horizontaler Verlauf
radial	"Radial, ..." "Radial, <R>, ..."	radialer Verlauf mit Standardradius 50% radialer Verlauf mit Ergänzungsparameter Radius R% Radius wird in Prozent des verfügbaren Platzes angegeben
konisch	"Conical, ..." "Conical, <S>, ..."	konischer Verlauf mit Start bei 90° konischer Verlauf mit Ergänzungsparameter S°, der den Startwinkel angibt

Die **Farbdefinitionen** stehen immer nach der Verlaufsart und eventuellen Ergnzungsparametern. Es knnen Farbnamen, HTML-Notation sowie die fr Farbressourcen bekannten Notationen *rgb(...)*, *hls(...)* und *grey(...)* verwendet werden.

Zusätzlich kann zu jeder Farbdefinition ein **Stopp-Punkt** angegeben werden, mit dem die Gewichtung der Farbe festgelegt wird. Ein Stopp-Punkt ist ein Prozentwert mit optionalem Prozentzeichen, der immer vor der jeweiligen Farbe steht und beeinflusst, wie viel Raum diese Farbe im Farbverlauf einnimmt. Ein Farbverlauf startet bei 0% und endet bei 100%, bezogen auf den Bereich, den er ausfüllt. Die Angabe ... , 20%, green, ... bedeutet zum Beispiel, dass nach 20% des zu füllenden Bereichs die Farbe Grün gesetzt wird. Wenn für den Bereich 0–20% keine andere Farbe gesetzt ist, werden die ersten 20% des Bereichs grün eingefärbt. Falls vor 20% bereits eine andere Farbe gesetzt ist, dann wird ein Übergang zwischen dieser Farbe und Grün dargestellt.

Stopp-Punkte sollten immer in aufsteigender Reihenfolge gesetzt werden. Sind mehrere Farben mit demselben Stopp-Punkt definiert, gilt die Farbe, die in der Parameterliste am weitesten hinten steht. Die Definition ... , 20%, *green*, 40%, *blue*, 20%, *red*, ... erzeugt beispielsweise einen Farbverlauf mit einem Rotton bei 20%, der in einen Blauton übergeht, welcher ab 40% satt dargestellt wird.

Beispiele

[illegible]

"Linear,red, #00FF00, rgb(0,0,255)"	gemischte Notationen, Farben gleichmäßig verteilt
"Linear, 20%, green, 60%, yellow, 80%, red");	benannte Farben mit prozentualen Stopp-Punkten
"Linear, 20, green, 60, yellow, 80 ,red"	das %-Zeichen bei den Stopp- Punkten
	ist optional

Hinweise

Qt lässt das Setzen von Farbverläufen an den meisten Stellen zu, wendet sie aber nicht unbedingt an. Ob ein Farbverlauf angezeigt wird, hängt sehr stark vom Objekt und UI-Stil ab. Dabei sind Flächen (z. B. Hintergründe) meist unproblematisch, bei filigranen Strukturen (z. B. Texten) wird der Verlauf oft durch eine einzelne Farbe ersetzt. Bei Gruppierungsobjekten werden Gradienten als Hintergrund in der Regel dargestellt.

2.143 .grey

Dieses Attribut definiert die Graustufe (0 ... 255) einer **color**-Ressource, die bei der Anzeige auf einem Graustufen-Monitor (*setup.color_type = coltype_grey*) verwendet wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_grey		Bezeichner: AT-grey
Datentyp: DT_integer		Datentyp: DT-integer
<i>Wertebereich</i>		
0 ... 255		
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	color	

2.144 .groupbox

Mit Hilfe dieses Attributes kann die hierarchisch übergeordnete Groupbox erfragt werden.

Definition

Datentyp
object

Zugriff
get

C
Bezeichner: AT_groupbox
Datentyp: DT_object

COBOL
Bezeichner: AT-groupbox
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.145 .height

Dieses Attribut gibt die aktuelle Höhe des Objektes an.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i> (kann 0 sein)	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_height	Bezeichner: AT-height
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

.height definiert die Höhe des gesamten **notebook**-Objekts mit all seinen Elementen. Wird die Höhe zu klein definiert, wird ein vom Fenstersystem abhängiger Minimalwert genommen. Die Höhe darf nicht 0 sein, da die benötigte Höhe nicht berechnet werden kann.

Beim **tablefield** kann das Attribut auch den Wert 0 haben. Dies bedeutet, dass der Dialog Manager die entsprechende Größe so berechnen soll, dass alle Elemente in der entsprechenden Richtung dargestellt werden können, ohne dass eine Scrollbar notwendig ist. Wird z.B. die Höhe des **tablefields** auf 0 gesetzt, wird es so breit, dass alle Zeilen vollständig in dem Objekt dargestellt werden können.

Bei der **toolbar** definiert *.height* (ohne Index) den Standardwert für die nicht mit *.height[class]* gesetzten Werte.

Anmerkung zum IDM für Windows

Die Höhe eines **poptextes** (Combobox) kann nicht gesetzt werden; das jeweilige Fenstersystem wählt automatisch einen Wert, der vom Zeichensatz abhängt.

Siehe auch

Attribute *.real_height*, *.width*, *.yauto*, *.ybottom*, *.ytop*

2.146 .height[class]

Höhe einer Toolbar im angedockten Zustand oder als Toolwindow.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_height	Bezeichner: AT-height
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	toolbar

Indexbereich

<i>toolbar</i>
Höhe, wenn die Toolbar angedockt ist
<i>window</i>
Höhe, wenn die Toolbar abgedockt ist (Toolwindow)

Ohne einen Index gibt das Attribut den Standardwert für beide Andockzustände zurück oder legt ihn fest.

Siehe auch

Attribut *.width[class]*

2.147 .height[enum]

Mit diesem Attribut können Korrekturfaktoren für die Berechnung der Rasterhöhe aus dem **font** definiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_height	Bezeichner: AT-height
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	font

Indexbereich

scale_factor

Mit diesem Indexwert definiert das Attribut einen Prozentsatz, der als Skalierungsfaktor mit dem der Basiswert multipliziert wird.

Der Basiswert ist die Höhe eines einzeiligen **edittextes** mit diesem **font**.

scale_offset

Mit diesem Indexwert definiert das Attribut einen Pixelwert, der als Konstante zum skalierten Basiswert addiert wird.

Die Höhe des Rasters wird also nach folgender Formel berechnet:

Rasterhöhe =
(<Höhe des edittextes mit diesem Font> * .height[scale_factor]) / 100 +
.height[scale_offset]

Siehe auch

Attribute *.real_yraster*, *.width[enum]*, *.yraster*

Kapitel „Berechnung der Rastergröße aus einem Referenzfont“ im Handbuch „Ressourcenreferenz“

2.148 .help

In diesem Attribut können beliebige Texte zu dem Objekt abgelegt werden, die z.B. zu Hilfezwecken verwendet werden können.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string, object [text]</i>	get, set	ja

C

Bezeichner: AT_help

Datentyp: DT_string, DT_text

COBOL

Bezeichner: AT-help

Datentyp: DT-string, DT-text

Klassifizierung

Standardattribut

2.149 .helpmenu

Mit Hilfe dieses Attributes wird eine nach rechts ausgerichtete Menübox in der Menüleiste definiert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_helpmenu	Bezeichner: AT-helpmenu
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	menubox

Hinweis

Ist das Attribut für mehr als eine Menübox in der Menüleiste gesetzt, entscheidet der DM, welche davon das Helpmenü ist. Definieren Sie daher bitte nur eine rechts ausgerichtete Menübox.

Anmerkung zum IDM für Windows

Ist das Attribut gesetzt, werden alle Menüboxes, die danach definiert werden, rechts angehängt.

2.150 .helppos

Dieses Attribut definiert die Position des Hilfetextes in der Statusbar. Es gibt an, in welcher Position der Hilfetext *.statushelp* darzustellen ist. Die Angabe von 0 (Defaultwert) gibt an, dass die Hilfeinformation ganzzeilig dargestellt wird. Der Wert *n* gibt an, dass der Hilfetext über alle Kinder bis zu dem *n*-ten dargestellt wird. Ist der Wert <0 , wird keine Hilfeinformation dargestellt. Dieses Attribut kann von Fenstersystem zu Fenstersystem variieren. Die Werte 0 und -1 sind aber immer setzbar und werden entsprechend obiger Beschreibung angezeigt.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_helppos Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-helppos Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> statusbar	

2.151 .hinttext

Mit Hilfe dieses Attributes kann einem **edittext** oder **poptext**-Objekt ein Platzhalter- bzw. Hinweis-Text hinzugefügt werden, der angezeigt wird, wenn das Objekt noch keinen Inhalt gesetzt hat. Am **poptext**-Objekt muss dabei der Style **edittext** gesetzt sein.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string	get, set	ja

C

Bezeichner: AT_hinttext

Datentyp: DT_string

COBOL

Bezeichner: AT-hinttext

Datentyp: DT-string

Klassifizierung

Standardattribut

Achtung:

Bei einem Leerzeichen als Inhalt wird der Platzhalter - oder Hinweis-Text nicht angezeigt, dies betrifft ebenso einige Formate.

Besonderheiten:

Dieses Attribut wird nur von WINDOWS und QT unterstützt.

Anmerkung für Windows:

Unter MICROSOFT WINDOWS wird das Attribut im Moment an folgenden Objekten unterstützt:

- » **edittext**-Objekt, wenn die Attribute *.multiline* und *.options[opt_rtf]* den Wert *false* besitzen
- » **poptext**-Objekt, wenn das Attribut *.style* den Wert *edittext* oder *listbox* besitzt

Der Hinttext wird angezeigt, wenn das Objekt den Fokus nicht besitzt und der Inhalt leer ist.

Anmerkung für Qt:

Wenn beim **edittext**-Objekt das Attribut *.multiline* den Wert *true* besitzt und Layoutattribute (z.B. *.alignment*, *.xmargin*, *.ymargin* etc.) gesetzt sind, so werden diese bei der Anzeige des Hinweis- oder Platzhaltertexts nicht beachtet. Der Hinweis- oder Platzhaltertext ist immer links oben angebunden.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.152 .hls[enum]

Mit diesem Attribut werden Farbton, Helligkeit und Farbsättigung einer HLS-Farbe bei einer **color**-Ressource definiert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein
C	COBOL	
Bezeichner: AT_hls	Bezeichner: AT-hls	
Datentyp: DT_integer	Datentyp: DT-integer	
<i>Wertebereich</i>		
0 ... 255		
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	color	

Indexbereich

- color_hue*
Bestimmt den Farbton.
- color_light*
Bestimmt die Helligkeit.
- color_sat*
Bestimmt die Farbsättigung.

Im HLS-Farbmodell sind Farben durch Farbton **H** (hue), Helligkeit **L** (lightness, luminance) und Farbsättigung **S** (saturation) definiert. Der Wertebereich für alle drei Werte ist 0 bis 255.

Die Farben sind in einem Zylinder angeordnet, bei dem die obere Fläche Weiß und die untere Schwarz ist. Die Helligkeit L steigt von 0 (dunkel) in der unteren bis 255 (hell) in der oberen Fläche. Bei den Farben in der Kreisfläche, die den Mittelpunkt des Zylinders schneidet, ist $L = 127$. Diese Farben sind weder aufgehellt (durch Mischung mit Weiß) noch abgedunkelt (durch Mischung mit Schwarz).

Der Farbton H wird durch den Winkel auf der Kreisfläche festgelegt. Der Wertebereich für H von 0 bis 255 und der Winkel W im Farbkreis (eigentlich von 0 bis 359 Grad) hängen wie folgt zusammen:

- » $H = W / 2$.
- » $W = (H \bmod 180) * 2$; das heißt, Werte von 180 bis 255 entsprechen den Farbtönen von 0 bis 75.

Rot hat den Wert $H = 0$ (oder 180), $H = 60$ (oder 240) ergibt Grün und $H = 120$ Blau.

Auf der senkrechten Geraden im Zentrum des Zylinders, liegen die Grautöne mit der Farbsättigung $S = 0$. Die Farben auf dem Zylindermantel haben die höchste Farbsättigung $S = 255$. Auf dem Umfang der mittleren Kreisfläche, liegen die sogenannten reinen Farben mit $L = 127$ und $S = 255$.

Hinweis

Die Attribute *.rgb[enum]*, *.hls[enum]* und *.name* schließen sich gegenseitig aus, sodass „get“ zu einer Fehlermeldung „can't get value“ führen kann.

2.153 .hsb_arrows

Das Attribut *.hsb_arrows* definiert, ob die horizontale Scrollbar Pfeile an ihren Enden hat.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_hsb_arrows	Bezeichner: AT-hsb-arrows
Datentyp: DT_boolean	Datentyp: DT-boolean

Klassifizierung
Scrollbarattribut

Wertebereich

<i>true</i>
Scrollbar mit Pfeilen
<i>false</i>
Scrollbar ohne Pfeile

Verfügbarkeit

Das Attribut wird nur unter Motif und erst ab IDM-Version A.05.02.h unterstützt.

Siehe auch

Attribut *.vsb_arrows*

2.154 .hsb_linemotion

Dieses Attribut definiert, um wie viele Einheiten sich der Wert von *.xorigin* ändert, wenn der Benutzer zeilenweise horizontal scrollt. Einheiten entsprechen Pixeln, wenn *.sizeraster = false* ist, und Rasterereinheiten bei *.sizeraster = true*.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
-----------------------------------	----------------------------	-------------------------------

C
Bezeichner: AT_hsb_linemotion
Datentyp: DT_integer

COBOL
Bezeichner: AT-hsb-linemotion
Datentyp: DT-integer

Klassifizierung
Scrollbarattribut

Siehe auch

Kapitel „Scrollbarattribute“

2.155 .hsb_optional

Über dieses Attribut wird die horizontale Scrollbar nur angezeigt, wenn sie tatsächlich benötigt wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_hsb_optional	Bezeichner: AT-hsb-optional
Datentyp: DT_boolean	Datentyp: DT-boolean

Klassifizierung
Scrollbarattribut

tablefield

Wird das Attribut auf *true* gesetzt, entscheidet der Dialog Manager, ob die horizontale Scrollbar aktuell benötigt wird oder nicht. Können z.B. alle Spalten eines Tablefields komplett in dem zur Verfügung stehenden Raum dargestellt werden und ist die horizontale Scrollbar optional gesetzt, wird diese Scrollbar nicht angezeigt.

Hinweis

.hsb_optional wird in der Motif Version ignoriert, d.h. *.hsb_optional* wird implizit immer auf *true* gesetzt.

Siehe auch

Kapitel „Scrollbarattribute“

2.156 .hsb_pagemotion

Dieses Attribut definiert, um wie viele Einheiten sich der Wert von *.xorigin* ändert, wenn der Benutzer seitenweise horizontal scrollt. Einheiten entsprechen Pixeln, wenn *.sizeraster = false* ist, und Rasterereinheiten bei *.sizeraster = true*.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_hsb_pagemotion
Datentyp: DT_integer

COBOL
Bezeichner: AT-hsb-pagemotion
Datentyp: DT-integer

Klassifizierung
Scrollbarattribut

Siehe auch

Kapitel „Scrollbarattribute“

2.157 .hsb_visible

Mit Hilfe dieses Attributs wird die Sichtbarkeit einer horizontalen Scrollbar definiert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C
Bezeichner: AT_hsb_visible
Datentyp: DT_boolean

COBOL
Bezeichner: AT-hsb-visible
Datentyp: DT-boolean

Klassifizierung
Scrollbarattribut

Siehe auch

Kapitel „Scrollbarattribute“

2.158 .icon

Das Attribut *.icon* weist einem Objekt ein Bild zu.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [<i>tile</i>] (<i>window</i>)	get, set	ja
enum (<i>messagebox</i>)		

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_icon	Bezeichner: AT-icon
Datentyp: DT_tile (<i>window</i>)	Datentyp: DT-tile (<i>window</i>)
Datentyp: DT_enum (<i>messagebox</i>)	Datentyp: DT-enum (<i>messagebox</i>)

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	messagebox, window

window

Beim Objekt **Fenster** wird mit *.icon* eine **tile**-Ressource zugewiesen, die als Symbol in der Titelleiste des Fensters angezeigt wird. Wenn *.icon* nicht gesetzt ist, verwendet der IDM die in den IDM-Bibliotheken vordefinierten Grafikressourcen „IDM_Applcon“ oder „IDM_Deflcon“.

Die Ressourcen „IDM_Applcon“ und „IDM_Deflcon“ sind vom IDM für das Symbol in der Titelleiste von Fenstern und unter MICROSOFT WINDOWS zusätzlich als Anwendungssymbol vorgesehen. Dieses Symbol kann geändert werden, indem in der Anwendung eine eigene Grafikressource mit dem Bezeichner „IDM_Applcon“ definiert wird. Das vom IDM bereitgestellte „IDM_Deflcon“ wird verwendet, wenn kein „IDM_Applcon“ vorhanden ist. Dies hängt unter anderem davon ab, wie die Anwendung gelinkt wird. Informationen zur Definition und Anbindung von Grafikressourcen können der Entwicklerdokumentation des Systems entnommen werden.

Das Attribut wird vom IDM FÜR MOTIF nicht unterstützt.

messagebox

Bei einer **messagebox** hat das *.icon*-Attribut den Datentyp *enum*.

Wertebereich

<i>icon_asterisk</i>	Zeigt das Fenstersystem-spezifische Informationssymbol an.
<i>icon_error</i>	Zeigt das Fenstersystem-spezifische Fehlersymbol an.
<i>icon_exclamation</i>	Zeigt das Fenstersystem-spezifische Warnungssymbol an.
<i>icon_hand</i>	Zeigt das Fenstersystem-spezifische Fehlersymbol an.
<i>icon_information</i>	Zeigt das Fenstersystem-spezifische Informationssymbol an.

icon_query

Zeigt das Fenstersystem-spezifische Fragesymbol an.

icon_question

Zeigt das Fenstersystem-spezifische Fragesymbol an.

icon_warning

Zeigt das Fenstersystem-spezifische Warnungssymbol an.

noicon

Zeigt kein Symbol an.

Die Repräsentation von *.icon* ist abhängig vom Fenstersystem, dessen Version und – auf QT – dem UI-Stil. Es kann auch Mehrfachbelegung geben.

Wert	WINDOWS	MOTIF	QT
<i>icon_asterisk</i>	wie <i>icon_information</i>		
<i>icon_error</i>	Stopp-Zeichen	Fehler-Zeichen (durchgestrichener Kreis)	kleines „x“
<i>icon_exclamation</i>	Ausrufezeichen „!“		
<i>icon_hand</i>	wie <i>icon_error</i>		
<i>icon_information</i>	kleines „i“ in einem Kreis	kleines „i“	
<i>icon_query</i>	wie <i>icon_question</i>		
<i>icon_question</i>	Fragezeichen „?“	Fragezeichen „?“ (in einem Kopf)	Fragezeichen „?“
<i>icon_warning</i>	wie <i>icon_exclamation</i>		
<i>noicon</i>	keins		

2.159 .iconic

Dieses Attribut spezifiziert, ob ein Fenster ikonisch (minimiert in der Taskbar) ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_iconic	Bezeichner: AT-iconic
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Siehe auch (nur IDM für Windows)

Kapitel „Fensterzustand bei ungewöhnlichen Attributwerten“ in der „Objektreferenz“

2.160 .iconifyable

Dieses Attribut definiert, ob ein Fenster iconifizierbar sein soll oder nicht bzw. ob ein Iconifiziermechanismus im Fenstertitel gesetzt sein soll oder nicht.

Das zugehörige Fenster erhält einen Iconifiziermechanismus zum Iconifizieren des Fensters, wenn der Wert des Argumentes auf true gesetzt ist. Ist dieser auf false gesetzt, so wird kein Iconifiziermechanismus gezeichnet.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_iconifyable	Bezeichner: AT-iconifyable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Besonderheit unter Motif

Das Attribut kann, abhängig vom verwendeten Display- bzw. Desktopmanager, nicht im sichtbaren Zustand umgesetzt werden; u.U. ist ein Setzen zur Laufzeit auch nicht möglich. In manchen Fällen kann es helfen, die Sichtbarkeit des Fensters umzuschalten.

Da unter Motif das Umsetzen des Attributs direkt vom verwendeten Display- bzw. Desktopmanager abhängig ist wird empfohlen, dieses Attribut unter Motif nur statisch bzw. direkt nach Erzeugen einer Instanz mittels `:create(..., true)` im unsichtbaren Zustand zu setzen.

2.161 .idispach

Über dieses Attribut kann unter Microsoft Windows auf den IDispatch COM Interface Pointer des Objekts zugegriffen werden. Wenn beim Objekt ein neuer Wert gesetzt wird, wird das gespeicherte COM Objekt gelöscht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>pointer</i>	get set (document , subcontrol)	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_idispach	Bezeichner: AT-idispach
Datentyp: DT_pointer	Datentyp: DT-pointer

Vererbung
nein

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	control, doccursor, document, subcontrol

In der Regelsprache darf das Attribut nur demselben Attribut eines anderen Dialog Manager Objektes zugewiesen werden. In den Programmierschnittstellen ist zu beachten, dass das COM Objekt nur so lange gültig ist, wie der Dialog Manager dieses verwendet. Eine Anwendung sollte deshalb nach einer Abfrage den Referenzzähler sofort erhöhen (COM Methode: IUnknown->AddRef). Wenn das Objekt nicht mehr gebraucht wird, muss der Zähler wieder heruntergezählt werden (COM Methode: IUnknown->Release). Es darf aber auf gar keinen Fall der Zähler öfter erniedrigt als erhöht werden, da sonst das COM Objekt freigegeben wird. Der Dialog Manager kann diese Situation nicht erkennen und wird abstürzen. Der Dialog Manager wird ebenso abstürzen, wenn der angegebene Zeiger nicht auf ein COM Interface zeigt.

Das Attribut ist für das Control im mode_client, den XML-Cursor, das XML-Dokument und das Subcontrol verfügbar, es ist nicht vererbbar. Das XML-Dokument überprüft, ob das IXMLDOMDocument2 COM Interface implementiert ist und der XML-Cursor überprüft, ob das IXMLDOMNode COM Interface implementiert ist. Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.162 .ignorecursor

Dieses Attribut ist definiert für Dialoge und Fenster und dient dazu einem beim **Setup**-Objekt gesetzten temporären Override-Cursor für spezielle Fenster oder Dialoge und deren Kinder zu ignorieren, um so den eingestellten oder Default-Cursor wieder zu erhalten.

Definition

<i>Datentyp</i> <i>boolean</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_ignorecursor Datentyp: DT_boolean		<i>COBOL</i> Bezeichner: AT-ignorecursor Datentyp: DT-boolean
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> dialog, window	

Anmerkungen

- » Wenn *.ignorecursor* bei einem **Fenster** oder **Dialog** auf *true* gesetzt ist, wird der temporäre Override-Cursor von diesem bestimmten Objekt und jedem seiner Kinder ignoriert.
- » Normalerweise wird der Cursor vom Vater-Objekt an das Kind-Objekt vererbt, wenn das Kind keinen eigenen Cursor definiert hat. Die einzige Ausnahme bildet das Objekt Fenster, welches nicht den Cursor des Vaters ererbt.

Beispiel

Bis auf ein Fenster wird für den gesamten Dialog ein temporärer Wait-Cursor gesetzt.

```
window Wi
{
  .ignorecursor true;
  .cursor CursorCross;
}
on Pb select
{
  setup.overridecursor := CursorWait;
}
```

Siehe auch

Attribut *.overridecursor*

2.163 .imagebgc

Mit Hilfe dieses Attributes wird die Hintergrundfarbe eines Bildes definiert. Dieses Attribut wird jedoch nur dann ausgewertet, wenn das Bild nicht aus einer Datei geladen wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [color]	get, set	ja

C
Bezeichner: AT_imagebgc
Datentyp: DT_color

COBOL
Bezeichner: AT-imagebgc
Datentyp: DT-color

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	image

2.164 .imagefgc

Mit Hilfe dieses Attributes wird die Vordergrundfarbe eines Bildes definiert. Dieses Attribut wird jedoch nur dann ausgewertet, wenn das Bild nicht aus einer Datei geladen wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_imagefgc	Bezeichner: AT-imagefgc
Datentyp: DT_color	Datentyp: DT-color

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	image

2.165 .incertime

Mit Hilfe dieses Attributes wird definiert, in welchen Abständen („incremental time“) der Timer jeweils aktiviert werden soll. *.incertime* kann nur beim Zeitabstand angegeben werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_incertime	Bezeichner: AT-incertime
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	timer

2.166 .index

Mit Hilfe dieses Attributes kann beim Ereignisobjekt **thisevent** die Selektion eines bestimmten Listbox- Poptext- oder Tablefeldeintrags abgefragt werden (gültig nur beim *select*-Ereignis).

Bei auf dem Bildschirm darstellbaren Objekten, kann mit Hilfe des Attributs *.index* erfragt werden, das wievielte Kind das angegebene Objekt ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer, index</i>	<i>get</i>

C
Bezeichner: AT_index
Datentyp: DT_integer, DT_index

COBOL
Bezeichner: AT-index
Datentyp: DT-integer, DT-index

Klassifizierung
Standardattribut

Bedeutung beim Objekt thisevent

Mit Hilfe dieses Attributes kann beim Ereignisobjekt **thisevent** die Selektion eines bestimmten Listbox- bzw. Poptexteintrags abgefragt werden (gültig nur beim *select*-Ereignis bei Listbox und Poptext (Combobox); Datentyp *integer*).

Ebenso kann mit Hilfe dieses Attributes beim Ereignisobjekt **thisevent** die Selektion eines bestimmten Tablefields abgefragt werden (gültig nur beim *select*-Ereignis bei Tablefield; Datentyp *index*).

Siehe auch

Kapitel „Ereignisobjekt thisevent“ im Handbuch „Regelsprache“

Bedeutung bei „normalen“ Objekten

Bei „normalen“, d.h. auf dem Bildschirm darstellbaren Objekten, kann mit Hilfe des Attributs *.index* erfragt werden, das wievielte Kind das angegebene Objekt ist. Damit kann man sich also die Suche nach dem Objekt in der Liste der Kinder eines Vater sparen, man kann direkt auf den entsprechenden Index zugreifen. Dieses Attribut gibt einen integer Wert zurück, der anzeigt, das wievielte Kind dieses Objekts bei seinem Vater ist. Über dieses Attribut lässt sich dann sehr einfach der Nachfolger bzw. Vorgänger des Objekts beim Vater-Objekt berechnen.

Damit sind die nachfolgenden Konstrukte möglich:

```
this.parent.child[this.index] = this
NextChild      := this.parent.child[this.index + 1];
PreviousChild := this.parent.child[this.index - 1];
```

Beispiel

```
variable object Next;  
variable object Previous;  
Next      := this.parent.child[this.index+1];  
Previous  := this.parent.child[this.index-1];
```

Wie im Dialog Manager üblich, muss man bei diesen Abfragen auf den Wertebereich des Indexes achten. Er ist definiert durch

$1 \leq \text{index} \leq \text{Anzahl der Kinder}$

Ein `child[0]` existiert im Dialog Manager nicht und erzeugt einen Index-Range Fehler, ebenso wenn der Index größer als `.childcount` ist.

Das Attribut `.index` ist ebenfalls für Menüs und Records zulässig. Bei Menüs kann damit abgefragt werden, das wievielte Kind ein Menüeintrag ist; bei Records kann man damit die im Record enthaltenen benutzerdefinierten Attribute sequentiell bearbeiten.

Mit dem Modulo-Operator kann auch zirkuläres Suchen in der Regelsprache programmiert werden.

Beispiel

```
variable object Parent      := this.parent;  
variable integer ChildCount := Parent.childcount;  
while CurrentChild <> this do  
  print CurrentChild;  
  CurrentChild := Parent.child[CurrentChild % ChildCount + 1];  
endwhile
```

2.167 .indexscope[attribute]

Mit diesem Attribut kann der Gültigkeitsbereich des Index von benutzerdefinierten, assoziativen Attributen gelesen werden.

Für den Zugriff ist als Index der Attributname über den Datentyp *attribute* zu verwenden

Dieses Attribut ist für alle Objekte verfügbar, welche benutzerdefinierte Attribute enthalten.

Definition

Datentyp
anyvalue

Zugriff
get

C
Bezeichner: AT_indexscope
Datentyp: DT_anyvalue

COBOL
Bezeichner: AT-indexscope
Datentyp: DT-anyvalue

Klassifizierung
Allgemeines Attribut

Siehe auch

Attribute *.scope[attribute]*, *.typescope*, *.typescope[integer]*

Kapitel „Gültigkeitsbereich zur besseren Typ-Prüfung“ im Handbuch „Regelsprache“

2.168 .input[integer]

Mit Hilfe dieses Attributes kann bei Funktionen und Regeln erfragt werden, ob es sich bei dem Parameter um einen Eingabeparameter handelt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>boolean</i>	get
<i>C</i>	<i>COBOL</i>
Bezeichner: AT_input	Bezeichner: AT-input
Datentyp: DT_boolean	Datentyp: DT-boolean
<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	function, rule

Beispiel

```
function Test (integer P input, integer Q output,  
              string R input output)  
  
for I := 1 to Test.count do  
  print Test.input[I];  
endfor
```

Ausgabe

```
true  
false  
true
```

Siehe auch

Kapitel „Funktionen (functions)“ und „Benannte Regeln (Unterprogramme)“ im Handbuch „Regelsprache“

2.169 .instance[integer]

Das Attribut ermöglicht von einem Modell aus den Zugriff auf eine von diesem Modell abgeleitete Instanz.

Der Index ist dabei ein Integerwert. Die Gesamtzahl aller von einem Modell abgeleiteten Instanzen kann mit `<model_id>.count[.instance]` erfragt werden.

Definition

Datentyp
object

Zugriff
get

C
Bezeichner: AT_instance
Datentyp: DT_object

COBOL
Bezeichner: AT-instance
Datentyp: DT-object

Klassifizierung
Objektspezifisches Attribut

2.170 .interfaceid

Mit Hilfe dieses Attributes wird einem Control-Objekt eine eindeutige Id gegeben. Diese ID wird mit Hilfe des Programms **guidgen.exe** generiert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	get

C
Bezeichner: AT_interfaceid
Datentyp: DT_string

COBOL
Bezeichner: AT-interfaceid
Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	control

Siehe auch

Kapitel „Das control-Objekt“ im Handbuch „OLE-Schnittstelle“

2.171 .is_applet

Mit diesem Attribut kann zur Laufzeit am **setup**-Objekt abgefragt werden, ob der IDM-Java-Client als Applet oder als Applikation ausgeführt wird. Das Attribut kann z.B. dazu verwendet werden, einen Dialog abhängig davon, wo der Client ausgeführt wird, eingebettet in einer HTML-Seite oder in einem Fenster anzuzeigen.

Definition

Datentyp

boolean

Zugriff

get

C

Bezeichner: AT_is_applet

Datentyp: DT_boolean

COBOL

Bezeichner: AT-is-applet

Datentyp: DT-boolean

Klassifizierung

Objektspezifisches Attribut

Objekte

setup

2.172 .itemcount

Dieses Attribut spezifiziert die Anzahl der Einträge einer Listbox bzw. eines Poptextes (Combobox). Es kann in den Regeln gesetzt und gelesen werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_itemcount	Bezeichner: AT-itemcount
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, poptext

Die Anzahl der Texte, die in einer **Listbox** angezeigt werden sollen, können mit Hilfe von *.itemcount* spezifiziert werden. Aus Gründen der Effizienz erlaubt es der DM, den Betrag für das Attribut höher zu setzen, als die in der Listbox bereits vorhandene Anzahl der Einträge. Neue Texte können dann anschließend gesetzt werden. Soll eine Listbox mit verschiedenen Texten aus der Regelsprache oder mit DM_SetValue gefüllt werden, sollte *.itemcount* zuerst auf den zu erwartenden Wert gesetzt werden. Erst dann kann der Inhalt gesetzt werden.

In einem **Poptext** hingegen kann der Betrag für das Attribut *.itemcount* nur kleiner oder gleich der gegenwärtigen Anzahl der Texte im Poptext gesetzt werden. Durch das Setzen dieses Attributes kann nur eine Reduktion der Texte im Poptext erreicht werden.

2.173 .itemorder

Mit Hilfe dieses Attributes kann die Definitionsreihenfolge für Attribute und Records innerhalb von Records erfragt werden. Dabei wird für jedes Attribut ein „A“, für jeden Kindrecord ein „R“ in dem String an der entsprechenden Stelle abgelegt.

Definition

Datentyp
string

Zugriff
get

C
Bezeichner: AT_itemordewr
Datentyp: DT_string

COBOL
Bezeichner: AT-itemorder
Datentyp: DT-string

Klassifizierung
Objektspezifisches Attribut

Objekte
record

Beispiel

```
record R
{
  string Name;
  record Kind1
  {
    integer Nummer;
    boolean An;
  }
  boolean Aus;
}
print R.itemorder;
```

Ausgabe

```
"ARA"
```

Hinweis

Das Attribut *.itemorder* ist dafür vorgesehen, die statische Definitionsreihenfolge der Record-Kinder zu liefern. Es ist nicht garantiert, dass es jederzeit und für jegliche dynamische Änderung von Attributen und Sub-Records die korrekte Reihenfolge der Kinder eines **records** wiedergibt.

2.174 .ixmlomdocument2

Über dieses Attribut kann unter Microsoft Windows auf den IXMLDOMDocument2 COM Interface Pointer des XML-Dokuments zugegriffen werden. Wenn beim XML-Dokument ein neuer Wert gesetzt wird, wird der gespeicherte DOM-Baum gelöscht. Alle bestehenden XML-Cursor werden ungültig. Bei einem ungültigen XML-Cursor besitzt das Attribut *.mapped* den Wert *false*.

Definition

<i>Datentyp</i> <i>pointer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
<i>C</i> Bezeichner: AT_ixmlomdocument2 Datentyp: DT_pointer	<i>COBOL</i> Bezeichner: AT-ixmlomdocument2 Datentyp: DT-pointer	
<i>Vererbung</i> nein		
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> document	

In der Regelsprache darf das Attribut nur demselben Attribut eines anderen Dialog Manager Objektes zugewiesen werden. In den Programmierschnittstellen ist zu beachten, dass das COM Objekt nur so lange gültig ist, wie der Dialog Manager dieses verwendet. Eine Anwendung sollte deshalb nach einer Abfrage den Referenzzähler sofort erhöhen (COM Methode: IUnknown->AddRef). Wenn das Objekt nicht mehr gebraucht wird, muss der Zähler wieder herunter gezählt werden (COM Methode: IUnknown->Release). Es darf aber auf gar keinen Fall der Zähler öfter erniedrigt als erhöht werden, da sonst das COM Objekt freigegeben wird. Der Dialog Manager kann diese Situation nicht erkennen und wird abstürzen. Der Dialog Manager wird ebenso abstürzen, wenn der angegebene Zeiger nicht auf ein COM Interface zeigt.

2.175 .ixmlDOMNode

Über dieses Attribut kann unter Microsoft Windows auf den IXMLDOMNode COM Interface Pointer des XML-Cursors zugegriffen werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>pointer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_ixmlDOMNode	Bezeichner: AT-ixmlDOMNode
Datentyp: DT_pointer	Datentyp: DT-pointer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	doccursor

In der Regelsprache darf das Attribut nur demselben Attribut eines anderen Dialog Manager Objektes zugewiesen werden. In den Programmierschnittstellen ist zu beachten, dass das COM Objekt nur so lange gültig ist, wie der Dialog Manager dieses verwendet. Eine Anwendung sollte deshalb nach einer Abfrage den Referenzzähler sofort erhöhen (COM Methode: IUnknown->AddRef). Wenn das Objekt nicht mehr gebraucht wird, muss der Zähler wieder heruntergezählt werden (COM Methode: IUnknown->Release). Es darf aber auf gar keinen Fall der Zähler öfter erniedrigt als erhöht werden, da sonst das COM Objekt freigegeben wird. Der Dialog Manager kann diese Situation nicht erkennen und wird abstürzen.

Das Attribut ist für den XML-Cursor verfügbar. Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.176 .ixmldomodelist

Über dieses Attribut kann unter Microsoft Windows auf den IXMLDOMNodeList COM Interface Pointer des XML-Cursors zugegriffen werden. Über diesen Interface Pointer kann auf die direkten Kinder des XML-Cursors zugegriffen werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>pointer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_ixmldomodelist	Bezeichner: AT-ixmldomodelist
Datentyp: DT_pointer	Datentyp: DT-pointer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	dccursor

In der Regelsprache darf das Attribut nur demselben Attribut eines anderen Dialog Manager Objektes zugewiesen werden. In den Programmierschnittstellen ist zu beachten, dass das COM Objekt nur so lange gültig ist, wie der Dialog Manager dieses verwendet. Eine Anwendung sollte deshalb nach einer Abfrage den Referenzzähler sofort erhöhen (COM Methode: IUnknown->AddRef). Wenn das Objekt nicht mehr gebraucht wird, muss der Zähler wieder heruntergezählt werden (COM Methode: IUnknown->Release). Es darf aber auf gar keinen Fall der Zähler öfter erniedrigt als erhöht werden, da sonst das COM Objekt freigegeben wird. Der Dialog Manager kann diese Situation nicht erkennen und wird abstürzen.

Das Attribut ist für den XML-Cursor verfügbar. Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.177 .keyboard

Mit Hilfe dieses **setup**-Objekt-Attributes kann die Accelerator-Variante abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_keyboard	Bezeichner: AT-keyboard
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

2.178 .label

Mit Hilfe dieses Attributes können die Identifikatoren, die internen Namen von Objekten erfragt und manipuliert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_label	Bezeichner: AT-label
Datentyp: DT_string	Datentyp: DT-string

Klassifizierung
Standardattribut

2.179 .label[anyvalue]

Dieses Attribut können Identifikatoren von bestimmten IDM Elementen wie bspw. Methoden oder benutzerdefinierten Attributen von Objekten oder Argumente von Funktionen erfragt werden. Ebenso können durch dieses Attribut Typkonvertierungen vorgenommen werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string, attribute, method, object [text , instance], enum, class, event, type, boolean	get	ja

C

Bezeichner: AT_label

Datentyp: DT_string, DT_text, DT_instance, DT_attribute, DT_method, DT_enum, DT_event, DT_class, DT_type, DT_boolean

COBOL

Bezeichner: AT-label

Datentyp: DT-string, DT-text, DT-instance, DT-attribute, DT-method, DT-enum, DT-event, DT-class, DT-type, DT-boolean

Klassifizierung

Standardattribut

Ermitteln von Bezeichnern

Argumente von Funktionen

Bei Funktionen kann man über diesen Attribut die Bezeichner der Argumente an einer bestimmten Stelle erfragen.

Als Index wird dabei die Position des Arguments beim Funktionsaufruf angegeben.

Ein Aufruf erfolgt nach folgendem Schema:

```
string := function.label[integer]
```

Benutzerdefinierte Attribute und Methoden

Von einem Objekt können benutzerdefinierte Attribute und Methoden erfragt werden, indem der Name als Indexstring übergeben wird.

Ein Aufruf erfolgt dabei nach folgendem Schema:

```
attribute := object.label[string]  
method := object.label[string]
```

Dabei gilt folgende Unterscheidung:

- » Indexstring beginnt mit einem „.“: man erhält das benutzerdefinierte Attribut des Objektes
- » Indexstring beginnt mit einem „.“: man erhält die benutzerdefinierte Methode des Objektes

Typkonvertierungen

Über dieses Attribut können ebenso Typkonvertierungen der folgenden Datentypen vorgenommen werden:

- » attribute
- » method
- » enum
- » class
- » event
- » datatype
- » boolean

Eine Konvertierung kann dabei in in beide Richtungen erfolgen:

```
type := object.label["type-string"]
"type-string" := object.label[type]
```

Beispiel

```
dialog D

function boolean Func(string Path, object Id) {
    return false;
}

window Wi {
    integer MySum := 512;

    rule void MyRule (string S)
    {
        print "Method called with: "+S;
    }
}

on dialog start {
    variable method M;
    variable attribute A;

    !! query parameters
    print Func.label[1];           // result: "Path"
    print Func.label[2];           // result: "Id"
```

```

!! retrieve userdefined attributes and methods
M := Wi.label[":MyRule"];
Wi:call(M, "Hello");           // result: "Method called
with: Hello"

A := Wi.label[".MySum"];
Wi.type[A] := string;

!! type conversions
print this.label[button_ok];   // result: "button_ok"
print this.label["button_ok"]; // result: button_ok
(enum)

print Wi.label[".MySum"];       // result: .MySum
(attribute)
print Wi.label[.MySum];        // result: "MySum"

print Wi.label[:MyRule];       // result: "MyRule"

print this.label>window];      // result: "window"
print this.label["window"];    // result: window (class)

print this.label[select];      // result: "select"
print this.label["select"];    // result: select (event)
}

```

2.180 .language

Beim **setup**-Objekt ermöglicht dieses Attribut den Zugriff auf die aktuelle Sprachvariante eines Textes. Durch Setzen dieses Attributes kann die aktuell angezeigte Sprachvariante verändert werden.

Bei einer Funktion kann mit diesem Attribut die Programmiersprache erfragt werden, in der die Funktion implementiert ist.

Datentyp	Zugriff	changed-Ereignis
integer (setup)	get, set (setup)	nein
enum (function)	get (function)	

C	COBOL
Bezeichner: AT_language	Bezeichner: AT-language
Datentyp: DT_integer (setup)	Datentyp: DT-integer (setup)
Datentyp: DT_enum (function)	Datentyp: DT-enum (function)

Klassifizierung	Objekte
Objektspezifisches Attribut	function , setup

function

Wertebereich

lang_default

Bei der Funktionsdefinition ist keine Programmiersprache angegeben. Die Funktion ist in der Standard-Programmiersprache C implementiert.

lang_c

Die Funktion ist in der Programmiersprache C implementiert.

lang_cobol

Die Funktion ist in der Programmiersprache COBOL implementiert.

lang_cobol_cancel

Die Funktion ist in der Programmiersprache COBOL implementiert und wird mit einer CANCEL-Anweisung aufgerufen.

2.181 .language[integer]

Mit diesem Attribut kann für eine **text**-Ressource der Text in der angegebenen Variante / erfragt oder gesetzt werden.

Definition

<i>Datentyp</i> <i>string</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
<i>C</i> Bezeichner: AT_language Datentyp: DT_string		<i>COBOL</i> Bezeichner: AT-language Datentyp: DT-string
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> text	

2.182 .lastchild

Über dieses Attribut kann auf das letzte Kind eines Objektes zugegriffen werden.

Definition

<i>Datentyp</i> object	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
---------------------------	----------------------------	-------------------------------

C
Bezeichner: AT_lastchild
Datentyp: DT_object

COBOL
Bezeichner: AT-lastchild
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.183 .lastmenu

Dieses Attribut bezeichnet das letzte Kind in der Menühierarchie.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

C
Bezeichner: AT_lastmenu
Datentyp: DT_object

COBOL
Bezeichner: AT-lastmenu
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.184 .lastrecord

Dieses Attribut greift auf das letzte Kind in der Record-Hierarchie zu.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_lastrecord	Bezeichner: AT-lastrecord
Datentyp: DT_record	Datentyp: DT-record

Klassifizierung
Hierarchieattribut

2.185 .lastsubcontrol

Dieses Attribut greift auf das letzte **Subcontrol** eines **Control**- oder **Subcontrol**-Objekts zu.

Definition

<i>Datentyp</i> <i>object</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
----------------------------------	----------------------------	-------------------------------

C
Bezeichner: AT_lastsubcontrol
Datentyp: DT_object

COBOL
Bezeichner: AT-lastsubcontrol
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.186 .lasttoolbar

Dieses Attribut bezeichnet die letzte Toolbar in einem Fenster.

Definition

<i>Datentyp</i> <i>object</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
----------------------------------	----------------------------	-------------------------------

C
Bezeichner: AT_lasttoolbar
Datentyp: DT_object

COBOL
Bezeichner: AT-lasttoolbar
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.187 .layoutbox

Mit Hilfe dieses Attributes kann die hierarchisch übergeordnete Layoutbox erfragt werden.

Definition

<i>Datentyp</i> <i>object</i>	<i>Zugriff</i> get
----------------------------------	-----------------------

C
Bezeichner: AT_layoutbox
Datentyp: DT_object

COBOL
Bezeichner: AT-layoutbox
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.188 .level[integer]

Dieses Attribut des Treeview-Objektes definiert die hierarchische Ordnung der Einträge untereinander. Der daraus resultierende Baum ist durch eine Vater-Kind-Ordnung gekennzeichnet.

Der gültige Indexbereich ist $0 \dots n$, wobei n den Wert des Attributes *.itemcount* repräsentiert. Der Defaultwert für ungesetzte Attributwerte des Bereiches $1 \dots n$ steht dabei in *.level[0]*.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_level	Bezeichner: AT-level
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	treeview

Die Hierarchie ergibt sich aus den Unterschieden zwischen aufeinanderfolgenden *.level[I]*-Werten und **nicht** durch ihren Absolutwert, dieses ist aber durch disziplinierte Vergabe der *.level[I]*-Werte erreichbar. Damit lässt sich ein und dieselbe hierarchische Ordnung durch beliebig viele unterschiedliche *.level[]*-Definitionen erreichen.

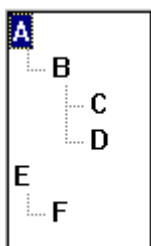
Bei der Bildung der Baumordnung wird nur der Bereich $1 \dots n$ berücksichtigt, da es nur für diesen Bereich auch einen tatsächlichen Inhalt (*.content[I]*) darzustellen gibt.

Dabei gelten die folgenden Regeln:

$\text{this.level}[I+1] > \text{this.level}[I]$	Der Eintrag $I+1$ ist ein Kind von I .
$\text{this.level}[I+1] = \text{this.level}[I]$	Der Eintrag $I+1$ ist ein Geschwister von I
$\text{this.level}[I+1] < \text{this.level}[I]$	Der Eintrag $I+1$ ist ein Kind vom Eintrag J , wobei der Vater J in $1 \dots I-1$ liegt und $\text{this.level}[J] < \text{this.level}[I+1]$ ist und für alle K in $J+1 \dots I+1$ gilt, dass $\text{this.level}[J] < \text{this.level}[K]$. Erfüllt kein J die obige Bedingungen, so ist $J=0$ der Vater.

Beispiel

Betrachten wir einmal folgende Baumansicht.



Diese Hierarchie kann gleichwertig z.B. durch die folgenden `.level[]`-Werte definiert werden:

Index I	.content[I]	.level[I] (absolute Werte)	.level[I] (andere Möglichkeit)
1	A	1	5
2	B	2	7
3	C	3	900
4	D	3	88
5	E	1	-42
6	F	2	0

Anmerkung

Dieses Attribut ist nur statisch verfügbar und wird nicht weitervererbt.

2.189 .license_key

In diesem Attribut des **control**-Objekts kann man den Lizenzschlüssel für ActiveX-Controls angeben, z.Zt. nur im Client-Modus relevant.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_license_key	Bezeichner: AT-license-key
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	control

2.190 .linemotion

Dieses Attribut definiert den Betrag, um den sich der Scrollbarwert (*.curvalue*) ändert, wenn der Benutzer zeilenweise scrollt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_linemotion	Bezeichner: AT-linemotion
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	scrollbar

Anmerkung

Das Attribut *.linemotion* wird von Motif ignoriert, wenn das Attribut *.arrows* auf *false* gesetzt ist.

2.191 .load

Dieses Attribut des Objekts **import** macht Angaben über den Ladezustand des zugehörigen Moduls.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_load	Bezeichner: AT-load
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	import

Siehe auch

Kapitel „Modularisierung“ im Handbuch „Programmiertechniken“

2.192 .local

Mit Hilfe dieses Attributes können Sie definieren, ob eine Anwendung lokal oder über ein Netzwerk laufen soll.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_local	Bezeichner: AT-local
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

2.193 .logfile

Mit Hilfe dieses Attributs kann während der Laufzeit über das **setup**-Objekt der absolute Pfad des Logfiles abgefragt werden.

Definition

Datentyp
string

Zugriff
get

C
Bezeichner: AT_logfile
Datentyp: DT_string

COBOL
Bezeichner: AT-logfile
Datentyp: DT-string

Klassifizierung
Objektspezifisches Attribut

Objekte
setup

2.194 .majortabheight

Definiert die Höhe aller Majortabs (Hauptindex) eines **Notebooks**.

Mit dem Standardwert 0 wird die Höhe automatisch berechnet, so dass die Majortabs aller sichtbarer Notepages (*.real_visible = true*) den im Tab enthaltenen Text vollständig anzeigen.

Wenn das Attribut *.sizeraster = true* gesetzt wird, wird dieser Wert in Rasterkoordinaten angegeben.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_majortabheight Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-majortabheight Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> notebook	

Hinweis

Da eine Neuberechnung der Größe sämtlicher Tabs nötig ist, wenn eine Notepage sichtbar oder unsichtbar gesetzt wird, und dies eine Größenänderung der Notepages und aller Kindobjekte zur Folge haben kann, ist die Verwendung des Wertes 0 nur für Notebooks zu empfehlen, die ein statisches Layout besitzen.

Anmerkung zum IDM für Windows

Unter MICROSOFT WINDOWS bedeutet der Wert 0, dass WINDOWS die Höhe selbst bestimmt. Wenn allerdings das Attribut *.majortabwidth* ungleich 0 oder das Attribut *.tabalignment = 1* gesetzt sind, wird die Höhe so berechnet, dass Text und Bild in den Tabs sichtbar sind.

Siehe auch

Attribute *.majortabwidth*, *.sizeraster*, *.tabalignment*

2.195 .majortabwidth

Definiert die Breite aller Majortabs (Hauptindex) eines **Notebooks**.

Mit dem Standardwert 0 wird die Breite automatisch berechnet, so dass die Majortabs aller sichtbarer Notepages (*.real_visible = true*) den im Tab enthaltenen Text vollständig anzeigen.

Wenn das Attribut *.sizeraster = true* gesetzt wird, wird dieser Wert in Rasterkoordinaten angegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_majortabwidth	Bezeichner: AT-majortabwidth
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	notebook

Hinweis

Da eine Neuberechnung der Größe sämtlicher Tabs nötig ist, wenn eine Notepage sichtbar oder unsichtbar gesetzt wird, und dies eine Größenänderung der Notepages und aller Kindobjekte zur Folge haben kann, ist die Verwendung des Wertes 0 nur für Notebooks zu empfehlen, die ein statisches Layout besitzen.

Anmerkung zum IDM für Windows

Unter MICROSOFT WINDOWS bedeutet der Wert 0, dass WINDOWS die Breite jedes Tab individuell bestimmt. Wenn allerdings das Attribut *.majortabheight* ungleich 0 oder das Attribut *.tabalignment = 1* gesetzt sind, wird die Breite so berechnet, dass Text und Bild des breitesten, sichtbar gesetzten Tab angezeigt werden kann. Alle Tabs erhalten diese einheitliche Breite.

Siehe auch

Attribute *.majortabheight*, *.sizeraster*, *.tabalignment*

2.196 .mapped

Ein Objekt, an dem dieses Attribut auf *false* steht, wird bei *.visible := true* zwar angelegt, aber noch nicht gezeichnet. Das für den Benutzer sichtbare Erscheinen des Objekts erfolgt nun erst beim Setzen dieses Attributs auf *true*.

Standardwert dieses Attributs ist *true*.

Achtung

Das Attribut sollte nur in Ausnahmefällen auf *false* gesetzt werden.

doccursor

Ist *true*, wenn der XML-Cursor auf einen Knoten im DOM-Baum zeigt. Es ist zu beachten, dass der Zugriff auf ein anderes Attribut den XML-Cursor in der Regel auf die Wurzel des DOM-Baumes positioniert.

Definition

<i>Datentyp</i> <i>boolean</i>	<i>Zugriff</i> get, set get (doccursor)	<i>changed-Ereignis</i> nein
<i>C</i> Bezeichner: AT_mapped Datentyp: DT_boolean		<i>COBOL</i> Bezeichner: AT-mapped Datentyp: DT-boolean
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> doccursor	

2.197 .mapping[integer]

Über das *mapping* Attribut kann auf die Mapping-Objekte eines Transformers zugegriffen werden. Das Attribut wird mit dem Objektindex indiziert (ähnlich zu child).

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object (mapping)</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_mapping	Bezeichner: AT-mapping
Datentyp: DT_mapping	Datentyp: DT-mapping

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	transformer

2.198 .masterapplication[enum]

Mit diesem Attribut wird die Zuordnung von Funktionen, die direkt unter einem Modul/Dialog liegen, zu einem **application**-Objekt definiert.

Als optionaler Index-Wert können Werte aus dem Enum-Bereich *lang_default...lang_java* und *func_normal...func_data* genutzt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [application]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_masterapplication	Bezeichner: AT-masterapplication
Datentyp: DT_application	Datentyp: DT-application

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	dialog, module

Indexbereich

lang_default

Bei der Funktionsdefinition ist keine Programmiersprache angegeben. Die Funktion ist in der Standard-Programmiersprache C implementiert.

lang_c

Die Funktion ist in der Programmiersprache C implementiert.

lang_cobol

Die Funktion ist in der Programmiersprache COBOL implementiert.

lang_cobol_cancel

Die Funktion ist in der Programmiersprache COBOL implementiert und wird mit einer CANCEL-Anweisung aufgerufen.

lang_fortran (veraltet)

Die Funktion ist in der Programmiersprache FORTRAN implementiert.

lang_java (veraltet)

Die Funktion ist in der Programmiersprache JAVA implementiert.

lang_pascal (veraltet)

Die Funktion ist in der Programmiersprache PASCAL implementiert.

func_callback (intern)

Die Funktion ist eine Objektcallback-Funktion, die einem Objekt im Attribut *.function* zugewiesen werden kann und für definierte Ereignisse aufgerufen wird.

Schlüsselwort bei der Funktionsdefinition: **callback**.

func_canvas (intern)

Die Funktion ist eine Canvas-Funktion, die einem **canvas**-Objekt im Attribut *.canvasfunc* zugewiesen werden kann.

Schlüsselwort bei der Funktionsdefinition: **canvasfunc**.

func_content (intern)

Die Funktion ist eine Nachlade-Funktion, die einem **tablefield** im Attribut *.contentfunc* zugewiesen werden kann.

Schlüsselwort bei der Funktionsdefinition: **contentfunc**.

func_data (intern)

Die Funktion ist eine Datenfunktion, die Objekten im *.datamodel*-Attribut zugewiesen werden kann und vom IDM bei der Synchronisierung zwischen Model- und View-Komponente aufgerufen wird.

Schlüsselwort bei der Funktionsdefinition: **datafunc**.

func_format (intern)

Die Funktion ist eine Format-Funktion, die den Objekten **edittext** und **tablefield** im Attribut *.formatfunc* zugewiesen werden kann.

Schlüsselwort bei der Funktionsdefinition: **formatfunc**.

func_normal (intern)

Die Funktion ist eine „normale“ Anwendungsfunktion, die aus der Regelsprache heraus aufgerufen werden kann.

func_spinbox (intern)

Die Funktion dient zur Steuerung der Wertanzeige einer **spinbox** mit *.style = void* durch die Anwendung und kann der **spinbox** im *.spinfunc*-Attribut zugewiesen werden.

Schlüsselwort bei der Funktionsdefinition: **spinfunc**.

Wird eine Modul-Funktion aufgerufen, so wird die Zuordnung zu einer Applikation über das *.masterapplication[]*-Attribut am Modul gesucht. Dies geschieht dabei entsprechend der angegebenen Sprachdefinition, dem Funktionstyp oder auch ganz ohne Typisierung (letzteres dann ohne Index-Angabe). Diese Applikations-Zuordnung wird dann genutzt um die Funktionsanbindung zu bestimmen und so die Funktion entsprechend aufzurufen. Wenn am Modul keine Zuordnung definiert wurde, wird am Dialog gesucht. Eine Setzung auf **null** verhindert die Suche am Dialog.

Das *.masterapplication*-Attribut wurde zur IDM-Version A.06.02.m eingeführt um in ähnlicher Weise zum *.application*-Attribut am Import-Objekt eine Zuordnung von Modul-Funktionen zu ermöglichen.

Beispiel

Modularisierter Dialog mit den Funktionen im Modul „Functions.mod“, bei dem COBOL-Funktionen der Server-Applikation „ApplServer“ zugeordnet werden und alle anderen Funktionen der „dynlib“-Applikation „ApplLocal“.

```
// Dialog.dlg
dialog Dlg
{
    .masterapplication ApplLocal;
}

application ApplLocal {
    .transport "dynlib";
    .exec "libusercheck.so";
    .active true;
}
```

```

}

use Functions;

on dialog start
{
    variable string Username := setup.env["USER"];
    if ValidateUser(Username) then
        print "Age = "+GetAge(Username);
    endif
    exit();
}

// Functions.mod
module Functions
{
    .masterapplication[lang_cobol] ApplServer;
}

use Server;

record RecUser
{
    string[50] FirstName;
    string[50] LastName;
    integer Age;
}

export function boolean ValidateUser(string Username);
function cobol integer GetUserProfile(string[50] Username,
                                     record RecUser output);

export rule integer GetAge(string Username)
{
    if GetUserProfile(Username, RecUser)>0 then
        return RecUser.Age;
    endif
    return 0;
}

// Server.mod
module Server

export application ApplServer {
    .connect "server:4712";
    .active true;
}

```

Verfügbarkeit

Ab IDM-Version A.06.02.m

Siehe auch

Attribut *.application*

Kapitel „Modularisierung“ im Handbuch „Programmiertechniken“

2.199 .maxchars

Das Attribut *.maxchars* gibt die maximale Anzahl von Zeichen an, die in einen editierbaren Text eingegeben werden können.

Ist *.maxchars = 0*, ist die Anzahl der Zeichen unbegrenzt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_maxchars	Bezeichner: AT-maxchars
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	edittext

Anmerkung

Wird bei einem einzeiligen editierbaren Text die Feldbreite durch *.width* kleiner gewählt als *.maxchars*, so werden die eingegebenen Zeichen horizontal verschoben.

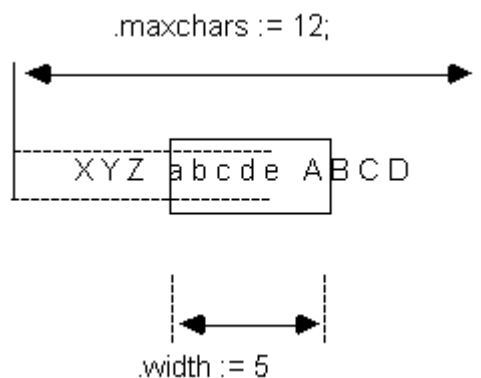


Abbildung 1: Sichtbarer Ausschnitt eines editierbaren Textes

Bei einem mehrzeiligen Text gilt:

Solange die in *.maxchars* definierte Anzahl von Zeichen noch nicht erreicht ist, können Zeilen angefügt werden. Sind mehr Zeilen erzeugt worden, als dargestellt werden können, so rückt die oberste Zeile außerhalb der oberen Begrenzung des Objektes editierbarer Text („scrolling“). Durch entsprechendes Positionieren mit dem Textcursor kann jede Position innerhalb der Eingabezeilen erreicht werden.

2.200 .maxchars[index]

Mit Hilfe dieses Attributes kann die maximale Anzahl von Zeichen definiert werden, die in bestimmte Felder [row, column] eingegeben werden darf.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_maxchars	Bezeichner: AT-maxchars
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.201 .maxheight

Dieses Attribut definiert die maximale Höhe des Objekts.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_maxheight
Datentyp: DT_integer

COBOL
Bezeichner: AT-maxheight
Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Attribute *.maxwidth*, *.minheight*

2.202 .maximized

Mit Hilfe dieses Attributs kann der Maximierungszustand des Objekts **Fenster** definiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_maximized	Bezeichner: AT-maximized
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Anmerkung

Unter Motif wird dieses Attribut ignoriert.

Siehe auch (nur IDM für Windows)

Kapitel „Fensterzustand bei ungewöhnlichen Attributwerten“ in der „Objektreferenz“

2.203 .maxsize[integer]

Mit diesem Attribut können an den einzelnen Splitbereichen einer Splitbox die maximal zulässigen Größen gesetzt werden, die beim interaktiven Verschieben der Splitbars von dem Endanwender nicht überschritten werden können. Es sind sowohl Pixel- als auch Rasterwerte zulässig.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
-----------------------------------	----------------------------	-------------------------------

<i>C</i> Bezeichner: AT_maxsize Datentyp: DT_integer	<i>COBOL</i> Bezeichner: AT-maxsize Datentyp: DT-integer
--	--

<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> splitbox
---	----------------------------

Der Index ist nullbasiert und die gültigen Werte dafür sind:

maxsize[l] – gültig, falls $0 \leq l$ und $l \leq \text{childcount}$

maxsize[0] ist das sogenannte Nullelement, das keinem Splitbereich zugeordnet ist, und nur der Bequemlichkeit dient. Der Wert des Nullelements wird auf die anderen Elemente des *maxsize[l]*-Vektors übertragen, für die es keine explizite Setzung gibt.

Siehe auch

Attribute *.minsize[integer]*, *.size[integer]*

2.204 .maxvalue

Dieses Attribut definiert den maximalen Wert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja
<i>string, object [text] (datetime)</i>		
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_maxvalue		Bezeichner: AT-maxvalue
Datentyp: DT_integer		Datentyp: DT-integer
Datentyp: DT_string, DT_text (datetime)		Datentyp: DT-string, DT-text (datetime)
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	datetime, progressbar, scrollbar	

scrollbar

Dieses Attribut enthält den maximal möglichen Scrollbarwert.

Der Wertebereich der abgefragten Sliderposition wird durch *.minvalue* und *.maxvalue* definiert.

progressbar

Maximalwert des Progress-Bereiches.

Gibt den 100% Wert vor. Dies ist der Wert, den *.curvalue* erreicht, wenn die Aktion, deren Fortlauf dargestellt wird, beendet ist.

datetime

Mit diesem Attribut wird der Maximalwert festgelegt.

Beim Schreibzugriff („set“) können Werte vom Datentyp *string* und **text**-Ressourcen angegeben werden. **text**-Ressourcen werden automatisch zu *string* gewandelt.

Lesezugriff („get“) liefert immer einen Wert vom Datentyp *string*.

Wertebereich

""	Kein Maximalwert gesetzt.
<value_string>	Setzt den enthaltenen Datums- oder Zeitwert als Maximalwert. Syntax und Auswertung des Strings sind beim Attribut .value beschrieben.

Das Attribut definiert den größten Wert, der vom Anwender eingegeben oder ausgewählt werden kann. Ein vorhandener Wert von *.value* wird in der Regel nicht korrigiert um *.maxvalue* einzuhalten. Es ist auch nicht definiert, ob beim Sichtbarmachen des **datetime** der angezeigte Wert korrigiert wird.

Siehe auch

Attribut *.minvalue*

2.205 .maxwidth

Dieses Attribut legt die maximale Breite des Objekts fest.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_maxwidth	Bezeichner: AT-maxwidth
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Attribute *.maxheight*, *.minwidth*

2.206 .member[integer]

Dieses Attribut liefert das i-te benutzerdefinierte Attribut.

Definition

<i>Datentyp</i> <i>attribute</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
-------------------------------------	----------------------------	-------------------------------

C
Bezeichner: AT_member
Datentyp: DT_attribute

COBOL
Bezeichner: AT-member
Datentyp: DT-attribute

Klassifizierung
Hierarchieattribut

Siehe auch

Kapitel „Benutzerdefinierte Attribute“ im Handbuch „Benutzerdefinierte Attribute und Methoden“

2.207 .membercount

Dieses Attribut definiert die Anzahl der benutzerdefinierten Attribute an einem Objekt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_membercount	Bezeichner: AT-membercount
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Hierarchieattribut

Siehe auch

Kapitel „Benutzerdefinierte Attribute“ im Handbuch „Benutzerdefinierte Attribute und Methoden“

2.208 .menu

Dieses Attribut ist der Identifikator eines Menüs, das an dem Objekt hängen soll. Dieses tritt als Popup-Menü in Erscheinung.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

C

Bezeichner: AT_menu

Datentyp: DT_object

COBOL

Bezeichner: AT-menu

Datentyp: DT-object

Klassifizierung

Allgemeines Attribut

Hinweis zum IDM für Windows

Popup-Menues werden hier auch über die Menütaste des Systems geöffnet (unter Windows ist das die Taste **Alt**). Will man direkt das Fenstermenü öffnen, so sollte die Taste **F10** genutzt werden.

Hinweis zum IDM für Motif

Popup-Menues können hier auch ab Motif 2.1 bei allen Objektklassen standardmäßig mit der Tastenkombination **Shift + F10** geöffnet werden.

Das Objekt **rectangle** kann kein Popup-Menü besitzen. Das Attribut *.menu* wird im IDM für Motif für dieses Objekt ignoriert.

Siehe auch

Attribut *.menu[integer]*

Methode openpopup

2.209 .menu[integer]

Dieses Attribut bezeichnet das i-te Menü-Kind eines Fensters (Pulldown-Menü).

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_menu	Bezeichner: AT-menu
Datentyp: DT_object	Datentyp: DT-object

Klassifizierung
Hierarchieattribut

Wenn das Menü über eine hierarchische Vorlage erzeugt wurde, kann es nur gelesen werden, d.h. es ist nur erlaubt, das Attribut zu definieren, jedoch nicht, es zu verändern.

Siehe auch

Attribut .menu

2.210 .menubgc

Über dieses Attribut wird die Hintergrundfarbe der Menüleiste definiert. Dieses Attribut wird nur unter und speziellen Motif-Versionen ausgewertet.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

C
Bezeichner: AT_menubgc
Datentyp: DT_color

COBOL
Bezeichner: AT-menubgc
Datentyp: DT-color

Klassifizierung
Layoutattribut

2.211 .menucount

Dieses Attribut erfragt die Anzahl der Menüelemente in einem Fenster.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

C
Bezeichner: AT_menucount
Datentyp: DT_integer

COBOL
Bezeichner: AT-menucount
Datentyp: DT-integer

Klassifizierung
Hierarchieattribut

2.212 .menufgc

Über dieses Attribut wird die Vordergrundfarbe der Menüleiste definiert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

C
Bezeichner: AT_menufgc
Datentyp: DT_color

COBOL
Bezeichner: AT-menufgc
Datentyp: DT-color

Klassifizierung
Layoutattribut

2.213 .message[integer]

An den Client zu verschickende Nachrichten beim **control**-Objekt (OLE).

Definition

<i>Datentyp</i>	<i>Zugriff</i>
object (message)	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_message	Bezeichner: AT-message
Datentyp: DT_object	Datentyp: DT-object

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	control

Siehe auch

Kapitel „Ereignisse“ im Handbuch „OLE-Schnittstelle“

2.214 .mincolwidth

Mit Hilfe dieses Attributes kann in einer **layoutbox** die minimale Spaltenbreite definiert werden. Es wird nur beachtet, wenn die **layoutbox** spaltenweise ausgerichtet ist. Das Attribut gibt dann die Mindestbreite für ein Objekt mit *.xauto* = 0 an, falls sich kein anderes Objekt oder nur noch weitere Objekte mit *.xauto* = 0 in der jeweiligen Spalte befinden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_mincolwidth	Bezeichner: AT-mincolwidth
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	layoutbox

Siehe auch

Attribut *.minrowheight*

2.215 .mincolwidth[integer]

Das Attribut legt die minimale Breite der Spalten in der Detailansicht des **listview**-Objekts fest. Anwender können die Spalte interaktiv nicht schmaler machen, als in diesem Attribut definiert.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_mincolwidth	Bezeichner: AT-mincolwidth
Datentyp: DT_integer	Datentyp: DT-integer

<i>Standardwert</i>	<i>Vererbung</i>
0	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listview

Der Wertebereich des Index ist $0 \dots \text{colcount}$, wobei der Wert mit Index 0 als Standardwert für nicht gesetzte Werte im Bereich $1 \dots \text{colcount}$ dient.

Wertebereich

0 (Standard)

Keine minimale Spaltenbreite.

Der Anwender kann die Spalte ausblenden, indem er den rechten Rand des Spaltenkopfs mit gedrückter Maustaste bis zum linken Spaltenrand zieht. Die Spalte kann wieder eingeblendet werden, indem die Maustaste im Spaltenkopf der nachfolgenden Spalte leicht rechts vom linken Spaltenrand gedrückt und die Maus mit gedrückter Taste nach rechts gezogen wird. Man achte auf die in diesem Fall unterschiedliche Form des Mauszeigers direkt über dem linken Spaltenrand und leicht rechts davon.

> 0

Minimale vom Anwender einstellbare Breite der Spalte in der Detailansicht.

Bei ungültigen Werten wird der Standardwert 0 verwendet. Der Attributwert wird allerdings nicht geändert.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

2.216 .minheight

Dieses Attribut legt die minimale Höhe eines Objekts fest.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_minheight	Bezeichner: AT-minheight
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Attribute *.maxheight*, *.minwidth*

2.217 .minortabheight

Definiert die Höhe aller Minortabs (Nebenindex).

Defaultwert 0. Damit wird die Höhe automatisch berechnet, sodass die Minortabs aller sichtbarer Notepages (*.real_visible = true*) passen.

Wenn das Attribut *.sizeraster = true* gesetzt wird, wird dieser Wert in Rasterkoordinaten angegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_minortabheight	Bezeichner: AT-minortabheight
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	notebook

Hinweis

Da eine Neuberechnung der Größe sämtlicher Tabs nötig ist, wenn eine Notepage sichtbar oder unsichtbar gesetzt wird, und dies eine Größenänderung der Notepages und aller Kindobjekte zur Folge haben kann, ist die Verwendung des Wertes 0 nur für Notebooks zu empfehlen, die ein statisches Layout besitzen.

Siehe auch

Attribute *.majortabheight*, *.minortabwidth*, *.sizeraster*

2.218 .minortabwidth

Definiert die Breite aller Minortabs (Nebenindex).

Defaultwert 0. Damit wird die Breite automatisch berechnet, sodass die Minortabs aller sichtbarer Notepages (*.real_visible = true*) passen.

Wenn das Attribut *.sizeraster = true* gesetzt wird, wird dieser Wert in Rasterkoordinaten angegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_minortabwidth	Bezeichner: AT-minortabwidth
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	notebook

Hinweis

Da eine Neuberechnung der Größe sämtlicher Tabs nötig ist, wenn eine Notepage sichtbar oder unsichtbar gesetzt wird, und dies eine Größenänderung der Notepages und aller Kindobjekte zur Folge haben kann, ist die Verwendung des Wertes 0 nur für Notebooks zu empfehlen, die ein statisches Layout besitzen.

Siehe auch

Attribute *.majortabwidth*, *.minortabheight*, *.sizeraster*

2.219 .minrowheight

Mit Hilfe dieses Attributes kann in einer **layoutbox** die minimale Zeilenhöhe definiert werden. Es wird nur beachtet, wenn die **layoutbox** zeilenweise ausgerichtet ist. Das Attribut gibt dann die Mindesthöhe für ein Objekt mit *.yauto = 0* an, falls sich kein anderes Objekt oder nur noch weitere Objekte mit *.yauto = 0* in der jeweiligen Zeile befinden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_minrowheight
Datentyp: DT_integer

COBOL
Bezeichner: AT-minrowheight
Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	layoutbox

Siehe auch

Attribut *.mincolwidth*

2.220 .minsize[integer]

Mit diesem Attribut können an den einzelnen Splitbereichen einer Splirbox die minimal zulässigen Größen gesetzt werden, die beim interaktiven Verschieben der Splitbars von dem Endanwender nicht unterschritten werden können. Es sind sowohl Pixel- als auch Rasterwerte zulässig.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_minsize	Bezeichner: AT-minsize
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	splitbox

Der Index ist nullbasiert und die gültigen Werte dafür sind:

minsize[I] – gültig, falls $0 \leq I$ und $I \leq \text{childcount}$

minsize[0] ist das sogenannte Nullelement, das keinem Splitbereich zugeordnet ist, und nur der Bequemlichkeit dient. Der Wert des Nullelements wird auf die anderen Elemente des *minsize[I]*-Vektors übertragen, für die es keine explizite Setzung gibt.

Siehe auch

Attribute *.maxsize[integer]*, *.size[integer]*

2.221 .minvalue

Dieses Attribut definiert den minimalen Wert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja
<i>string, object [text] (datetime)</i>		
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_minvalue		Bezeichner: AT-minvalue
Datentyp: DT_integer		Datentyp: DT-integer
Datentyp: DT_string, DT_text (datetime)		Datentyp: DT-string, DT-text (datetime)
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	datetime, progressbar, scrollbar	

scrollbar

Das Attribut enthält den minimal möglichen Scrollbarwert.

Der Wertebereich der abgefragten Sliderposition wird durch *.minvalue* und *.maxvalue* definiert.

progressbar

Minimalwert des Progress-Bereiches.

Gibt den 0% Wert vor. Dies ist der Wert, den *.curvalue* besitzt, wenn die Aktion, deren Fortlauf dargestellt wird, gestartet wird.

datetime

Mit diesem Attribut wird der Minimalwert festgelegt.

Beim Schreibzugriff („set“) können Werte vom Datentyp *string* und **text**-Ressourcen angegeben werden. **text**-Ressourcen werden automatisch zu *string* gewandelt.

Lesezugriff („get“) liefert immer einen Wert vom Datentyp *string*.

Wertebereich

""	Kein Minimalwert gesetzt.
<value_string>	Setzt den enthaltenen Datums- oder Zeitwert als Minimalwert. Syntax und Auswertung des Strings sind beim Attribut .value beschrieben.

Das Attribut definiert den kleinsten Wert, der vom Anwender eingegeben oder ausgewählt werden kann. Ein vorhandener Wert von *.value* wird in der Regel nicht korrigiert um *.minvalue* einzuhalten. Es ist auch nicht definiert, ob beim Sichtbarmachen des **datetime** der angezeigte Wert korrigiert wird.

Siehe auch

Attribut *.maxvalue*

2.222 .minwidth

Dieses Attribut gibt die minimal zulässige Breite des Objekts an.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_minwidth	Bezeichner: AT-minwidth
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Attribute *.maxwidth*, *.minheight*

2.223 .mode

Mit Hilfe dieses Attributes wird bei dem Objekt **control** gesteuert, ob das Control als OLE-Server dienen kann oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
enum	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_mode	Bezeichner: AT-mode
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	control

Wertebereich

mode_client

Das **control**-Objekt wird als OLE-Client betrachtet, d.h. über dieses Objekt wird ein OLE-Server angesprochen.

mode_server

Das **control**-Objekt dient als OLE-Server und kann von anderen Clients angesprochen werden.

mode_none

Einsatzart ist undefiniert, d.h. das **control** wird weder als Client noch als Server benutzt. Diesen Wert kann man z.B. setzen, wenn man eigentlich einen OLE-Server implementiert hat, aber aus irgendwelchen Gründen zur Zeit nicht in der Lage ist als Server zu agieren, z.B. weil das Programm als ganz normales Programm gestartet worden ist.

2.224 .model

Hier handelt es sich um den Identifikator eines Objektes, das als Vorlage definiert wurde. Das neue Objekt enthält alle Attribute des Vorlagenobjekts, sofern diese nicht durch eine lokale Definition beim Objekt überschrieben wurden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_model	Bezeichner: AT-model
Datentyp: DT_object	Datentyp: DT-object

Klassifizierung
Standardattribut

Siehe auch

Methode :instance_of()

Kapitel „Einsatz von Modellen“ im Handbuch „Programmiertechniken“

2.225 .module

Dieses Attribut definiert das Modul zu einem Objekt.

Definition

Datentyp
object

Zugriff
get

C
Bezeichner: AT_module
Datentyp: DT_object

COBOL
Bezeichner: AT-module
Datentyp: DT-object

Klassifizierung
Standardattribut

2.226 .mouse_buttons

Mit Hilfe dieses Setup-Objekt-Attributes kann die Anzahl der logischen Maustasten erfragt werden (0 = keine Maus).

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_mouse_buttons	Bezeichner: AT-mouse-buttons
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkung

Die Anzahl der physikalisch vorhandenen Maustasten kann sich von der Anzahl der logischen Maustasten unterscheiden, die von Maus und Treiber-Software emuliert werden. In solchen Fällen ist oft durch Drücken von zwei Tasten gleichzeitig eine weitere Taste emuliert.

2.227 .mouseover

Über das Attribut *.mouseover* kann ein **image**-Objekt so eingestellt werden, dass es auf Mouseover-Ereignisse reagiert.

Standardmäßig ist das Attribut auf *false* gesetzt und das vom **image**-Objekt angezeigte Bild ändert sich nicht, wenn sich der Mauszeiger über dem Objekt befindet.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_mouseover	Bezeichner: AT-mouseover
Datentyp: DT_boolean	Datentyp: DT-boolean

Standardwert
false

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	image

Wird *.mouseover* auf *true* gesetzt, werden die in *.picture[tile_mouse_over]* bzw. *.picture[tile_active_mouse_over]* angegebenen **tile**-Ressourcen angezeigt, wenn der Mauszeiger über das **image**-Objekt bewegt bzw. die linke Maustaste über ihm gedrückt wird.

Siehe auch

Attribut *.picture[enum]*

2.228 .moveable

Dieses Attribut legt fest, ob das Fenster interaktiv vom Benutzer verschoben werden darf oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_moveable	Bezeichner: AT-moveable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Besonderheit unter Motif

Das Attribut kann, abhängig vom verwendeten Display- bzw. Desktop Manager, nicht im sichtbaren Zustand umgesetzt werden; unter Umständen ist ein Setzen zur Laufzeit auch nicht möglich. In manchen Fällen kann es helfen, die Sichtbarkeit des Fensters umzuschalten.

Da unter MOTIF das Umsetzen dieses Attributs direkt vom verwendeten Display- bzw. Desktop Manager abhängig ist, wird empfohlen, das Attribut nur statisch bzw. direkt nach Erzeugen einer Instanz mittels `:create(..., true)` im unsichtbaren Zustand zu setzen.

Anmerkung Qt

Das Attribut `.moveable` hat unter Qt keine Auswirkung. Fenster lassen sich immer verschieben.

2.229 .msgboxtext[enum]

Mit Hilfe dieses Attributes können Sie die Beschriftung von MessageBox-Buttons definieren. Wird eine Button-Beschriftung vom Fenstersystem nicht unterstützt, wird das Attribut *.msgboxtext* ignoriert.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [text], string	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_msgboxtext	Bezeichner: AT-msgboxtext
Datentyp: DT_text, DT_string	Datentyp: DT-text, DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	dialog

Manche Fenstersysteme (MICROSOFT WINDOWS) haben die Buttons der Messageboxes fest beschriftet und erlauben keine Beschriftung durch den Anwender. Andere Fenstersysteme (MOTIF) definieren keine Beschriftungen und benötigen diese von der Anwendung (Mehrsprachigkeit).

Das Attribut ist indiziert, wobei der Index Werte vom Typ *enum* annehmen muss, die die Form „button_...“ haben. Die Werte sind Texte (**text**-Ressourcen oder Strings), die den String der 7 möglichen Buttons enthält (vgl. Objekte *messagebox* und *dialog* in der „Objektreferenz“). Die einzelnen Buttonstrings sind im Englischen üblicherweise: *OK*, *Cancel*, *Retry*, *Abort*, *Ignore*, *Yes*, *No*.

Indexbereich

button_abort

Definiert die Beschriftung für die Schaltfläche „Abbrechen“ („Abort“).

button_cancel

Definiert die Beschriftung für die Schaltfläche „Abbrechen“ („Cancel“).

button_ignore

Definiert die Beschriftung für die Schaltfläche „Ignorieren“.

button_no

Definiert die Beschriftung für die Schaltfläche „Nein“.

button_ok

Definiert die Beschriftung für die Schaltfläche „OK“.

button_retry

Definiert die Beschriftung für die Schaltfläche „Wiederholen“.

button_yes

Definiert die Beschriftung für die Schaltfläche „Ja“.

Beispiel mit deutschen Texten

```
dialog Test
{
    .msgboxtext[button_ok]      "OK";
    .msgboxtext[button_cancel]  "Abbrechen";
}
```

```
.msgboxtext[button_retry]    "Wiederholen";  
.msgboxtext[button_abort]    "Abbrechen";  
.msgboxtext[button_ignore]    "Ignorieren";  
.msgboxtext[button_yes]       "Ja";  
.msgboxtext[button_no]        "Nein";  
}
```

Siehe auch

Objekt messagebox

2.230 .multiline

Dieses Attribut definiert, ob ein **edittext** ein- oder mehrzeilig ist bzw. ob die Tabs eines Notebooks mehrzeilig dargestellt werden sollen oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_multiline	Bezeichner: AT-multiline
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	edittext, notebook

2.231 .multisel

Dieses Attribut steuert ob vom Benutzer in einem Objekt mehrere Einträge ausgewählt werden können oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_multisel	Bezeichner: AT-multisel
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	filereq, listbox, tablefield

filereq

Dieses Attribut steuert die Auswahlmöglichkeit des Benutzers eines **filereq**-Objektes. Bei dem Wert *true* kann er mehrere Dateien auswählen. Die selektierten Dateien sind dann im indizierten Attribut *.value[integer]* zu finden. Im Normalfall steht die ausgewählte Datei bzw. das ausgewählte Verzeichnis im skalaren Attribut *.value*.

Besonderheiten

Dieses Attribut wird **nur** unter MICROSOFT WINDOWS für die Modi *fr_load* und *fr_save*, also für die Auswahl von Dateien, unterstützt.

listbox

Dieses Attribut definiert, ob Einfach- oder Mehrfachselektion in einer **listbox** zugelassen ist.

Ist *.multisel* auf *false* gesetzt, kann nur jeweils ein Listboxeintrag über einen Mausklick selektiert werden. Wird *.multisel* auf *true* gesetzt, können mit Hilfe der Maus mehrere Einträge ausgewählt werden. Wird ein bereits selektierter Eintrag erneut mit der Maus angeklickt, ist die Selektion damit aufgehoben, der Eintrag ist deselektiert.

Anmerkung zum IDM für Windows

Bewegt man den Cursor in einer **listbox**, für die *.multisel = false* gesetzt ist, über die Cursortasten auf und ab, wird der Eintrag mit dem Fokus ebenfalls aktiviert.

tablefield

Mit Hilfe dieses Attributes wird beim **tablefield** gesteuert, ob mehrere Felder, Spalten oder Zeilen abhängig von den aktiven Selektionsarten selektiert werden können (siehe Objekt tablefield).

Mehrfachselektion in Motif

Mehrfachselektion ist in diesem Fall eine Auswahl aus verschiedenen Objekten oder einer Reihe von Objekten (z.B. Einträge einer *listbox*) mit nur einer Aktion.

1. Um **mehrere Listboxeinträge** zu selektieren, die **nicht direkt aufeinanderfolgen**, wird ein Eintrag mit Hilfe der linken Maustaste selektiert und die Maustaste anschließend losgelassen. Anschließend wird der Cursor auf den nächsten zu selektierenden Eintrag positioniert, die **Strg**-Taste und daraufhin die linke Maustaste gedrückt. Auf die beschriebene Art und Weise können weitere Einträge selektiert werden.
2. Um **direkt aufeinanderfolgende Listboxeinträge** zu selektieren, wird der erste Eintrag ebenfalls mit Hilfe der linken Maustaste selektiert. Der selektierte Eintrag erscheint daraufhin invers. Alle zuvor selektierten Einträge sind nun deselektiert. Hält man die Maustaste gedrückt und bewegt den Cursor über weitere Einträge, erscheinen alle Einträge zwischen dem zuerst selektierten Eintrag und dem aktuellen Cursorstand invers, d.h. auch sie sind selektiert. Wird die Maustaste losgelassen, bedeutet dies auch das Ende der Selektionsreihe. Damit sind alle Einträge zwischen dem zuerst und dem zuletzt selektierten Eintrag ausgewählt.

Derselbe Effekt lässt sich erreichen, indem der Cursor vom ersten selektierten Eintrag zum letzten Eintrag der Reihe bewegt und anschließend die **Shift**-Taste gedrückt wird. Bei gedrückter **Shift**-Taste drückt man nun zusätzlich die linke Maustaste.

Sie können Einträge oder eine Reihe von Einträgen zu einer schon vorher selektierten Reihe **hinzufügen** oder Einträge oder eine Reihe von Einträgen von einer selektierten Reihe mit Hilfe der **Strg**-Taste **abziehen**.

Hinzufügen

Nachdem Sie die „erste“ Reihe von Einträgen selektiert haben, bewegen Sie den Cursor zum ersten Eintrag in der „neuen“ Reihe von Einträgen und drücken die **Strg**-Taste und die linke Maustaste gleichzeitig. Der Eintrag, auf dem sich der Cursor befindet, erscheint invers. Vorherige Selektionen bleiben unverändert.

Auf dieselbe Art müssen Sie, wenn Sie eine weitere Reihe selektieren wollen, nachdem Sie die erste Reihe selektiert haben, den Cursor auf die entsprechenden Positionen bewegen, die **Strg**-Taste drücken und die weiteren Aktionen – wie oben unter 1. und 2. beschrieben – ausführen.

Abziehen

Abziehen wird durch die Umkehrung der oben beschriebenen Aktionen ausgeführt. Das bedeutet, dass Sie die gleiche Vorgehensweise auf bereits selektierte Einträge anwenden.

Setzen Sie den Cursor auf den Eintrag der als erstes deselektiert werden soll, drücken Sie die **Strg**-Taste und anschließend die linke Maustaste. Der Eintrag auf dem sich der Cursor befindet erscheint „de-invertiert“, d.h. er erscheint in normalem Format. Wenn Sie die Maustaste gedrückt halten und den Cursor weiterbewegen, werden alle Einträge, die der Cursor berührt, deselektiert. Die gleiche Wirkung wird erzielt, wenn Sie den Cursor zu der jeweiligen Position bewegen, die **Shift**-Taste und die linken Maustaste drücken.

Sie können das beschriebene Selektionsverfahren auch mit der **Tastatur** ausführen. Der „Tastatur-Cursor“ (**Tab**), der in einem Rahmen angezeigt wird, muss in die Listbox bewegt werden. In diesem Modus wird die linke Maustaste durch die Wahl der **Leer**-Taste ersetzt. Ansonsten wird das Verfahren wie oben beschrieben ausgeführt.

Beispiel

```
Listbox.multisel := true;  
Listbox.active[I] := false;
```

Siehe auch

Attribute *.active[integer]*, *.activeitem*, *.selstyle*

2.232 .mustexist

Dieses Attribut stellt sicher, dass nur existierende Dateien/Verzeichnisse ausgewählt werden können (Wert *true*). Andernfalls findet keine Überprüfung statt, ob die Datei oder die eingegebene Pfadangabe korrekt ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C
Bezeichner: AT_*mustexist*
Datentyp: DT_*boolean*

COBOL
Bezeichner: AT-*mustexist*
Datentyp: DT-*boolean*

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	filereq

Besonderheiten Microsoft Windows

Dieses Attribut wirkt nur im Modus *fr_load*.

2.233 .name

Bei den Layoutressourcen Farbe (**color**), Muster (**tile**) und Zeichensatz (**font**) kann der jeweilige Ressourcename abgefragt oder gesetzt werden. Bei **color**-Ressourcen kann *.name* einer der vordefinierten Farbnamen des Systems sein. Bei Schriftarten kann es die Systembezeichnung einer der im System verfügbaren Schriftarten sein. Bei **tile**-Ressourcen kann *.name* der Pfad einer Bilddatei sein.

Dieses Attribut definiert bei einem **control**-Objekt die Programm-ID des OLE-Servers.

Am **doccursor** liefert das Attribut den Tag-Namen des DOM-Knotens.

Beim **mapping**-Objekt definiert dieses Attribut ein Muster für Knoten eines XML-Baums oder einer IDM-Objekthierarchie.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set get (doccursor)	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_name	Bezeichner: AT-name
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	color, control, doccursor, font, mapping, tile

control

Dieses Attribut definiert bei einem **control**-Objekt, das als OLE-Client eingesetzt wird, wie der zugehörige Server heißt. Dabei kann in diesem Attribut entweder der Name oder die ProgID des Servers hinterlegt werden.

Wenn das **control**-Objekt als OLE-Server verwendet wird, wird der in diesem Attribut angegebene String in die Registry eingetragen als Servername.

Beispiel

```
model control CtTest
{
  mode    mode_client;
  name    "InternetExplorer.Application.1";
  visible true;
  active  true;
  connect false;
}
```

mapping (XML)

Damit kann an einem **mapping**-Objekt ein Muster (siehe hierzu Definition des **mapping**-Objektes) definiert werden, das während einer Transformation darüber entscheidet, an welchen Knoten in dem zu transformierenden XML-Baum oder IDM-Objekthierarchie das **mapping**-Objekt aktiv und seine `:action()`-Methode ausgeführt wird.

doccursor (XML-Cursor)

Es ist zu beachten, dass ein XML-Cursor, dessen Attribut `.mapped` den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.234 .navigable

Für die Objekte **tablefield** und **edittext** sind Attribute verfügbar, die dem Dialogdesigner die Möglichkeit zu geben, das Aussehen und Eingabeverhalten beeinflussen zu können. Dies betrifft die Darstellung von nicht selektierbaren Edittexten sowie die Definition, ob der Benutzer mit diesen Objekten interagieren können soll oder nicht:

» **Bedienbar- oder Selektierbarkeit**

Attribut: *sensitive*

Die Eigenschaft, ob ein Objekt vom Benutzer selektiert werden kann oder nicht. Nur wenn ein Objekt selektierbar ist, kann der Benutzer die für das Objekt üblichen Aktionen auf das Objekt anwenden, z.B. Daten eingeben und Scrollen.

» **Editierbarkeit**

Attribut: *editable*

Die Eigenschaft, dass der Benutzer den Inhalt des Objektes ändern kann; er ist also in der Lage, den Inhalt zu editieren und zu verändern.

» **Fokussierbarkeit**

Attribut: *navigable*

Die Eigenschaft, dass das Objekt in der Tastatursteuerung erfasst wird und damit den Eingabefokus erhalten kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_navigable	Bezeichner: AT-navigable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	edittext, tablefield

Diese drei Attribute beeinflussen nun das Verhalten des Objektes in folgender Weise:

Attribut	Wert	Auswirkung
<i>.sensitive</i>	<i>true</i>	Objekt kann vom Benutzer selektiert werden. Nur wenn das Attribut <i>.sensitive</i> den Wert <i>true</i> hat, können die Attribute <i>.navigable</i> und <i>.editable</i> wirksam werden, ansonsten werden sie ignoriert.
<i>.sensitive</i>	<i>false</i>	Objekt kann nicht vom Benutzer selektiert werden. Es sind also keinerlei Aktionen auf dem Objekt möglich. Der Inhalt des Objekts wird wie für die Fenstersysteme üblich gegraut dargestellt.

Attribut	Wert	Auswirkung
<i>.editable</i>	<i>true</i>	Benutzer kann den Inhalt des Objektes ändern, falls er das Objekt selektieren kann.
<i>.editable</i>	<i>false</i>	Benutzer kann den Inhalt des Objektes nicht verändern.
<i>.navigable</i>	<i>true</i>	Objekt ist in der normalen Tastatursteuerung enthalten, d.h., der Benutzer kann das Objekt durch Navigation im zugehörigen Fenster erreichen.
<i>.navigable</i>	<i>false</i>	Das Objekt kann nicht über die Tastatursteuerung erreicht werden. Das Objekt kann jedoch trotzdem über die Maus den Fokus erhalten.

Anmerkungen zum IDM für Motif

- » Das Umsetzen von *.content* oder *.navigable* an einem **Edittext** kann Einfluss darauf haben, ob in einem anderen Edittext der Cursor (Caret) durchgezogen gezeichnet wird. Zu beachten ist, dass bei einem Edittext mit *.navigable = false* unter Motif keine Eingaben möglich sind.
- » Der durchgezogene Cursor erlaubt keinen Rückschluss auf Navigationsreihenfolge oder Fokussierung eines **Edittexts**.

Siehe auch

Attribute *.editable*, *.sensitive*

2.235 .navigation

Dieses Attribut definiert den Navigationstyp der Tastaturnavigation am **checkbox**-Objekt. Diese unterscheidet Objekte, über die mittels **Tab**-Taste navigiert wird und Objektgruppen (z.B. mehrere Radio-buttons oder Checkboxes) innerhalb derer die Navigation über die Cursortasten erfolgt.

Definition

<i>Datentyp</i> enum	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
<i>C</i> Bezeichner: AT_navigation Datentyp: DT_enum		<i>COBOL</i> Bezeichner: AT-navigation Datentyp: DT-enum
<i>Standardwert</i> navi_default		
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> checkbox	

Wertebereich

navi_default (Standard)

Objekte mit demselben Vaterobjekt und auf derselben Hierarchieebene der Objekthierarchie bilden eine Gruppe. Innerhalb einer Gruppe kann mit den Cursortasten navigiert werden.

navi_grouped

Bildet mit anderen Objekten, welche ebenfalls **navi_grouped** gesetzt haben, eine eigene Gruppe, innerhalb welcher die Tastaturnavigation über die Cursortasten erfolgt.

navi_tab

Das Objekt ist alleinstehend und somit über die **Tab**-Taste erreichbar.

Dabei ist darauf zu achten, dass alle Objekte innerhalb einer Gruppe, d.h. alle Objekte, welche innerhalb eines Gruppierungsobjekts (z.B. **groupbox**, **window**) liegen, den gleichen Wert gesetzt haben.

Verfügbarkeit

Dieses Attribut ist ab den IDM-Versionen A.04.04.o bzw. A.05.01.e verfügbar.

2.236 .nextactive[integer]

Dieses Attribut erfragt den nächsten aktiven Listboxeintrag ab einem gegebenen Eintrag. *.nextactive* gibt 0 zurück, wenn nach dem gegebenen Eintrag kein aktiver Eintrag mehr vorhanden ist. Damit kann man sehr effizient durch alle Elemente der **listbox** laufen und nur die aktiven bearbeiten.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_nextactive	Bezeichner: AT-nextactive
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox

Beispiel

```
dialog D

listbox Lb
{
    .multisel    true;
    .content[1]  "Hallo";
    .content[2]  "Welt";
    .content[3]  "!!";
    .active[1]   true;
    .active[3]   true;
}

on dialog start
{
    variable integer I := Lb.nextactive[0];

    while I <> 0 do
        print I;
        I := Lb.nextactive[I];
    endwhile

    exit();
}
```

2.237 .nextactive[index]

Hiermit kann die Anwendung sich alle aktiven Felder des Tablefields holen bzw. auch herausfinden, ob kein, ein oder mehrere Felder aktiv sind.

Für die Definition des nächsten aktiven Feldes ist es notwendig einen Index zu spezifizieren, ab dem nach einem aktiven Eintrag gesucht werden soll. Wird kein weiterer Eintrag mehr gefunden, so wird als Index $[0,0]$ zurückgegeben.

Eine Suche mit den Indizes $[I,0]$ bzw. $[0,J]$ ist nicht möglich. Da diese Indizes keine echten Felder der Tabelle darstellen wird der Aufruf so behandelt, als wäre der Index $[0,0]$ angegeben worden.

Die Suchrichtung wird durch das *.direction*-Attribut bestimmt. Bei *.direction* = 1 wird zeilenweise und bei *.direction* = 2 spaltenweise nach der nächsten selektierten Zelle gesucht. Sind beispielsweise die Zellen $[4,2]$ und $[3,5]$ selektiert, liefert *.nextactive[1,1]* bei *.direction* = 1 als Ergebnis $[3,5]$ und bei *.direction* = 2 als Ergebnis $[4,2]$.

Definition

Datentyp

integer

Zugriff

get

C

Bezeichner: AT_nextactive

Datentyp: DT_integer

COBOL

Bezeichner: AT-nextactive

Datentyp: DT-integer

Klassifizierung

Objektspezifisches Attribut

Objekte

tablefield

Siehe auch

Attribute *.active[index]*, *.direction*, *.selstyle*

2.238 .nodetype

Dient zur Abfrage des Typs des DOM-Knotens.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>enum</i>	<i>get</i>
 <i>C</i>	 <i>COBOL</i>
Bezeichner: AT_nodetype	Bezeichner: AT-nodetype
Datentyp: DT_enum	Datentyp: DT-enum
 <i>Klassifizierung</i>	 <i>Objekte</i>
Objektspezifisches Attribut	doccursor

Wertebereich

<i>nodetype_attribute</i>
Der Knoten ist ein Attribut eines Elements.
<i>nodetype_cdata_section</i>
Der Knoten ist ein Abschnitt mit ungeparsten Zeichendaten (CDATA).
<i>nodetype_comment</i>
Der Knoten ist ein Kommentar.
<i>nodetype_document</i>
Der Knoten ist ein vollständiges Dokument.
<i>nodetype_document_fragment</i>
Der Knoten ist ein Ausschnitt aus einem Dokument.
<i>nodetype_document_type</i>
Der Knoten ist eine Dokumenttyp-Deklaration.
<i>nodetype_element</i>
Der Knoten ist ein Element.
<i>nodetype_entity</i>
Der Knoten ist eine Deklaration einer Entität.
<i>nodetype_entity_reference</i>
Der Knoten ist eine Referenz auf eine deklarierte Entität.
<i>nodetype_notation</i>
Der Knoten ist eine in der DTD deklarierte Notation.
<i>nodetype_processing_instruction</i>
Der Knoten ist eine Verarbeitungsanweisung.
<i>nodetype_text</i>
Der Knoten ist der Textinhalt eines Elements oder ein Attributwert.

Hinweis

Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.239 .notepage

Mit Hilfe dieses Attributes kann die Notepage erfragt werden, zu dem das Objekt gehört. Wenn das Objekt nicht in einer Notepage liegt, wird das *null*-Objekt als Ergebnis zurückgeliefert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>object</i>	get

C
Bezeichner: AT_notepage
Datentyp: DT_object

COBOL
Bezeichner: AT-notepage
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.240 .open[integer]

Mit Hilfe dieses Attributs wird beim **treeview** definiert, ob der Teilbaum des Knotens *I* geöffnet oder geschlossen dargestellt wird. Durch Umsetzen auf *true* wird der Teilbaum geöffnet, mit *false* wird er geschlossen. Dabei ist zu beachten, dass ein Setzen von *.open[I] := true* an einem Kindknoten **nicht** automatisch die Vaterknoten dieses Knotens öffnet.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_open	Bezeichner: AT-open
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	treeview

Der Wert mit Index 0 dient als Standardwert. Mit ihm kann bei der Definition des **treeviews** angegeben werden, ob dieser vollständig geöffnet oder geschlossen ist. Der Standardwert vererbt sich auf neu angelegte Knoten.

Befinden sich beim Schließen eines Teilbaumes aktive Einträge im Teilbaum, so werden diese inaktiv gesetzt und die Wurzel des Teilbaums wird aktiv gesetzt.

2.241 .opsys_string

Mit Hilfe dieses Setup-Objekt-Attributes kann die Betriebssystemkennung erfragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	<i>get</i>

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_opsys_string	Bezeichner: AT-opsys-string
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Attribut *.opsys_type*

2.242 .opsys_type

Mit Hilfe dieses Setup-Objekt-Attributes kann die Betriebssystemart erfragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>enum</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_opsys_type	Bezeichner: AT-opsys-type
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Wertebereich

os_nt
Das Betriebssystem ist eine 32-Bit-Variante von MICROSOFT WINDOWS, basierend auf der Technologie von WINDOWS NT.

os_unix
Das Betriebssystem ist eine Variante von UNIX oder LINUX.

os_w64
Das Betriebssystem ist eine 64-Bit-Variante von MICROSOFT WINDOWS.

Siehe auch

Attribut *.opsys_string*

2.243 .options[enum]

Dieses Attribut stellt besondere Einstellungen an den jeweiligen Objektklassen zur Verfügung.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_options	Bezeichner: AT-options
Datentyp: DT_boolean	Datentyp: DT-boolean

Klassifizierung

Objektspezifisches Attribut

.options kann mit IDM-Enums adressiert werden. Näheres entnehmen Sie bitte der folgenden Tabelle.

Objekt	option_index	Bedeutung
application	<i>opt_cert_required</i> (ab IDM-Version A.06.02.h)	Gibt an, ob bei einer SSL-Verbindung ein Zertifikat benötigt wird.
	<i>opt_no_ssl_v2</i> (ab IDM-Version A.06.02.h)	Gibt an, dass bei einer SSL-Verbindung die Version 2 des SSL-Protokolls nicht verwendet wird.
	<i>opt_verify_peer</i> (ab IDM-Version A.06.02.h)	Gibt an, ob bei einer SSL-Verbindung die andere Seite verifiziert werden soll.
canvas	<i>opt_accept_child</i> (nur IDM FÜR MOTIF)	Canvas akzeptiert Kind-Widgets (d.h. kein Fokus).
	<i>opt_addevents</i> (nur IDM FÜR QT)	Ruft die Canvas-Funktion bei allen <i>QEvents</i> auf. Als <i>reason</i> wird dabei <i>CCR_event</i> mitgegeben. So kann beispielsweise auf Mausbewegungen reagiert werden. false (Standard) Normaler Aufruf der Canvas-Funktion. true Die Canvas-Funktion wird zusätzlich zu den normalen Aufrufen auch für alle <i>QEvents</i> aufgerufen.

Objekt	option_index	Bedeutung
	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_focus_frame</i> (nur IDM FÜR MOTIF)	Fokusrahmen wird nicht gezeichnet.
	<i>opt_graphicsview</i> (nur IDM FÜR QT)	Steuert, ob die Canvas mit einem QFrame oder einer QGraphicsView umgesetzt wird. Je nach Optionswert ergeben sich Unterschiede der DM_CanvasUserArgs. false (Standard) Umsetzung mit einem QFrame. Im <i>Expose</i>-Ereignis kann direkt in das QWidget gezeichnet werden. true Umsetzung mit einem QGraphicsView. Es wird keine <i>Expose</i>-Nachricht an die Canvas-Funktion weitergeleitet. Dieser Modus, in dem die QGraphicsView der Container einer QGraphicsScene ist, dient zum Beispiel dazu, der QGraphicsScene Objekte hinzuzufügen, die von ihr dann selbständig gezeichnet werden.
	<i>opt_motif_shadow</i> (nur IDM FÜR MOTIF)	Canvas zeichnet Motif-Schattenrand.

Objekt	option_index	Bedeutung
checkbox	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_push_like</i> (nur IDM FÜR WINDOWS)	Die Checkbox soll ein Verhalten wie ein Pushbutton haben.
	<i>opt_use_widget</i> (nur IDM FÜR MOTIF)	Verwendet nicht Gadget, sondern immer Widget.
dialog	<i>opt_et_margin</i> (nur IDM FÜR WINDOWS)	Mit Hilfe dieser Option kann eingestellt werden, ob ein Rand innerhalb des Eingabetextes erscheinen soll (Standard) oder nicht.
edittext	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_html</i> (nur IDM FÜR QT)	Steuert, ob der edittext HTML verarbeitet und Inhalte entsprechend formatiert darstellt. false (Standard) Keine HTML-Verarbeitung und keine formatierte Darstellung. true HTML wird verarbeitet und der Inhalt formatiert dargestellt.

Objekt	option_index	Bedeutung
	<i>opt_rtf</i> (nur IDM FÜR WINDOWS)	Steuert, ob der edittext das „Rich Text Format“ (RTF) verarbeitet und Inhalte formatiert darstellt. false (Standard) Keine RTF-Verarbeitung und keine formatierte Darstellung. true RTF wird verarbeitet und der Inhalt formatiert dargestellt.
filereq	<i>fro_createprompt</i> (nur IDM FÜR WINDOWS, nur im Modus <i>fr_load</i> mit <i>.mustexist = true</i>)	Bei der Selektion einer vorhandenen Datei erscheint ein Dialog zur Absicherung des Anlegens einer Datei.
	<i>fro_overwriteprompt</i> (nur IDM FÜR WINDOWS und IDM FÜR QT, nur im Modus <i>fr_save</i>)	Bei der Selektion einer vorhandenen Datei erscheint ein Dialog mit der Frage, ob die Datei überschrieben werden soll. Hinweis zum IDM FÜR QT Bei der Kombination <i>.mustexist = true</i> mit <i>.options[fro_overwriteprompt] = true</i> wird kein Dialog zum Bestätigen des Überschreibens angezeigt.
	<i>fro_relativepath</i> (nur IDM FÜR MOTIF, ab Motif-Version 2.1)	Das Muster und der momentane Pfad werden getrennt dargestellt und die selektierte Datei wird nur als Dateiname ohne Verzeichnis angezeigt.
groupbox	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.

Objekt	option_index	Bedeutung
	<i>opt_scroll_on_focus</i> (nur IDM FÜR MOTIF, ab IDM-Version A.05.02.i)	Steuert das Verhalten bei Fokussierung und kann die Erreichbarkeit per Tastaturnavigation verbessern (siehe Kapitel „Motif-Option .options [opt_scroll_on_focus]“). true (Standard) Der IDM versucht, ein Kindobjekt, das den Fokus erhalten soll, in den sichtbaren Bereich zu scrollen. false Der IDM versucht nicht, zu fokussierende Kindobjekte in den sichtbaren Bereich zu scrollen.
image	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_sim_insensitive</i> (nur IDM FÜR WINDOWS und IDM FÜR QT, IDM FÜR QT ab IDM-Version A.06.02.g)	Steuert, ob automatisch ein Bild für die insensitive Darstellung erzeugt wird. false (Standard) Es wird kein Bild erzeugt. true Für die insensitive Darstellung wird ein verblasstes Bild erzeugt.
layoutbox	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.

Objekt	option_index	Bedeutung
	<i>opt_scroll_on_focus</i> (nur IDM FÜR MOTIF, ab IDM-Version A.05.02.i)	Steuert das Verhalten bei Fokussierung und kann die Erreichbarkeit per Tastaturnavigation verbessern (siehe Kapitel „Motif-Option .options [opt_scroll_on_focus]“). true (Standard) Der IDM versucht, ein Kindobjekt, das den Fokus erhalten soll, in den sichtbaren Bereich zu scrollen. false Der IDM versucht nicht, zu fokussierende Kindobjekte in den sichtbaren Bereich zu scrollen.
listbox	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_quick_navigate</i> (nur IDM FÜR MOTIF)	Ausschalten der „Quick-Navigate-Funktion“. Dies dient zur Umgehung eines Motif-Bugs (ab Motif 2.1), der bei eingeschalteter „Quick-Navigate-Funktion“ zu Pieptönen beim Betätigen der Tastatur führt. false „Quick-Navigate-Funktion“ ist ausgeschaltet. true (Standard) „Quick-Navigate-Funktion“ ist eingeschaltet.

Objekt	option_index	Bedeutung
	<i>opt_scroll_pixels</i> (nur IDM FÜR QT, ab IDM-Version A.06.01.c)	Aktiviert den „Pixel-Scrollmodus“, bei dem nicht eintragsweise gescrollt wird, sondern immer der nächste nicht sichtbare Bereich in den Anzeigebereich (viewport) geholt wird. Lange Einträge, die über den Anzeigebereich hinausgehen, können damit schrittweise gescrollt werden. Dabei entspricht ein Seitenschritt (page step) der Höhe des Anzeigebereichs und ein Einzelschritt (single step) der Höhe des ersten Eintrags (jeweils in Pixel). false (Standard) „Pixel-Scrollmodus“ ist deaktiviert. Es wird eintragsweise gescrollt. true „Pixel-Scrollmodus“ ist aktiviert. Beim Scrollen wird der nächste nicht sichtbare Bereich in den Anzeigebereich geholt.
menubox	<i>opt_enable_tearoff</i> (nur IDM FÜR MOTIF)	Die Menübox soll vom Benutzer aus der Menüleiste herausnehmbar und frei platzierbar sein.
menuitem	<i>opt_use_widget</i> (nur IDM FÜR MOTIF)	Verwendet nicht Gadget, sondern immer Widget.
notebook	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
notepage	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.

Objekt	option_index	Bedeutung
	<i>opt_scroll_on_focus</i> (nur IDM FÜR MOTIF, ab IDM-Version A.05.02.i)	Steuert das Verhalten bei Fokussierung und kann die Erreichbarkeit per Tastaturnavigation verbessern (siehe Kapitel „Motif-Option .options [opt_scroll_on_focus]“). true (Standard) Der IDM versucht, ein Kindobjekt, das den Fokus erhalten soll, in den sichtbaren Bereich zu scrollen. false Der IDM versucht nicht, zu fokussierende Kindobjekte in den sichtbaren Bereich zu scrollen.
poptext	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.

Objekt	option_index	Bedeutung
	<i>opt_hscroll</i> (nur IDM FÜR WINDOWS, ab IDM-Version A.06.01.h)	Aktiviert eine horizontale Scrollbar in der Liste des poptexts. false (Standard) Es wird keine horizontale Scrollbar eingeblendet. true Eine horizontale Scrollbar wird eingeblendet, wenn die Liste Einträge enthält, die nicht vollständig angezeigt werden können. Hinweis Die horizontale Scrollbar erscheint innerhalb des Darstellungsbereichs und überdeckt somit die untersten Zeilen. Es wird dafür dann auch eine vertikale Scrollbar eingeblendet. Da die Bedienung von Scrollbars in einer geöffneten Liste für den Anwender nicht einfach ist, sollte diese Option nur verwendet werden, wenn es keine andere Lösung gibt.
	<i>opt_mnemonic</i>	Definiert ob Mnemonics verarbeitet werden. false Mnemonics werden ignoriert. Ein „&“ wird als Zeichen dargestellt. true (Standard) Mnemonics werden verarbeitet. Ein „&“-Zeichen wird nicht angezeigt sondern als Kennzeichen für einen Mnemonic interpretiert.
	<i>opt_old_select</i>	Steuert das Auslösen von <i>select</i> - und <i>activate</i> -Ereignissen.

Objekt	option_index	Bedeutung
	<i>opt_quick_navigate</i> (nur IDM FÜR MOTIF)	Ausschalten der „Quick-Navigate-Funktion“. Dies dient zur Umgehung eines Motif-Bugs (ab Motif 2.1), der bei eingeschalteter „Quick-Navigate-Funktion“ zu Pieptönen beim Betätigen der Tastatur führt. false „Quick-Navigate-Funktion“ ist ausgeschaltet. true (Standard) „Quick-Navigate-Funktion“ ist eingeschaltet. Hinweis „Quick-Navigate“ funktioniert nur bei <i>.style = pop-text</i>.
	<i>opt_sort</i> (nur IDM FÜR WINDOWS)	Bestimmt ob die Liste sortiert angezeigt wird. Die Sortierung betrifft lediglich die Anzeige, die Reihenfolge im Attribut <i>.text[integer]</i> und die Indizes der Einträge bleiben unverändert. false (Standard) Keine automatische Sortierung der Einträge. true Die Listeneinträge werden automatisch sortiert. Unterstützt das Standardverhalten von MICROSOFT WINDOWS, das von einer sortierten Liste ausgeht.
progressbar	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.

Objekt	option_index	Bedeutung
pushbutton	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_use_widget</i> (nur IDM FÜR MOTIF)	Verwendet nicht Gadget, sondern immer Widget.
radiobutton	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_push_like</i> (nur IDM FÜR WINDOWS)	Der Radiobutton soll ein Verhalten wie ein Pushbutton haben.
	<i>opt_use_widget</i> (nur IDM FÜR MOTIF)	Verwendet nicht Gadget, sondern immer Widget.
rectangle	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.

Objekt	option_index	Bedeutung
scrollbar	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
setup	<i>opt_balloon_toolhelp</i> (nur IDM FÜR WINDOWS, ab IDM-Version A.05.01.c)	Legt das Aussehen der Toolhelp fest: true (Standard) Die Toolhelp wird als Sprechblase angezeigt. false Die Toolhelp wird als Rechteck angezeigt.
	<i>opt_yi_monitoring</i>	Wird diese Option auf <i>false</i> gesetzt, dann werden durch YiRegisterUserEventMonitor() installierte Monitorfunktionen, nicht aufgerufen. Dies ist zur Fehlersuche wichtig, wenn der Verdacht besteht, dass eine Monitorfunktion fehlerhaft implementiert ist. Sind die Monitorfunktionen mittels Startoption oder Umgebungsvariable ausgeschaltet, lassen sie sich durch Setzen von <i>true</i> nicht einschalten.

Objekt	option_index	Bedeutung
	<i>opt_fontrafter_compat</i>	Mit dieser Option kann festgelegt werden, dass die alte Berechnung der Rastergröße verwendet werden soll. Achtung: mit der Version A.06.03.a hat sich die Berechnung der Rastergröße geändert. Wenn kein Referenzstring angegeben ist, wird jetzt M als Referenzstring verwendet. Die Option <i>opt_fontrafter_compat</i> kann benutzt werden um vorübergehend die alte Berechnungsmethode zu verwenden. Bei Verwendung dieser Option basiert die Größenberechnung zum Teil auf dem Systemfont, der nicht High DPI fähig ist, daher sollten High DPI fähige Anwendungen, die mit IDM FÜR WINDOWS 11 erstellt wurden, diese Option nicht verwenden. Siehe auch „Optionen für die Rasterberechnung unter Windows“
spinbox	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
splitbox	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.

Objekt	option_index	Bedeutung
statictext	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_use_widget</i> (nur IDM FÜR MOTIF)	Verwendet nicht Gadget, sondern immer Widget.
tablefield	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_new_align</i> (IDM FÜR MOTIF und IDM FÜR WINDOWS)	Legt fest, welche Attribute für die Ausrichtung von Text maßgeblich sind. false (Standard) Für die Textausrichtung sind die Attribute <i>.colalignment[integer]</i> und <i>.rowalignment[integer]</i> maßgeblich. true Für die Textausrichtung sind die Attribute <i>.xalignment[index]</i> und <i>.yalignment[index]</i> maßgeblich.

Objekt	option_index	Bedeutung
	<i>opt_new_colwidth</i> (nur IDM FÜR MOTIF)	In älteren Versionen des IDM (vor A.03.10.f) wurde unter Motif bei Tabellen mit einem <i>reffont</i> die Spaltenbreite falsch berechnet. Dazu gab es in Version A.03.10.f eine zweite Korrektur, bei der die Option <i>opt_new_colwidth</i> eingeführt wurde. true (Standard) Korrigierte Spaltenbreite (Berechnung auf Basis von Rastergrößen, wie beim Layout anderer Objekte). false Berechnung der Spaltenbreite entsprechend der Option <i>opt_old_colwidth</i>.
	<i>opt_old_colwidth</i> (nur IDM FÜR MOTIF)	In älteren Versionen des IDM (vor A.03.04.a) waren unter Motif bei Tabellen mit einem <i>reffont</i> die Spalten zu breit. Dazu gab es in Version A.03.04.a eine erste Korrektur, bei der die Option <i>opt_old_colwidth</i> eingeführt wurde. false (Standard) Korrigierte, kleinere Spaltenbreite. true Spaltenbreite wie vor der Korrektur. Hinweis Ab Version A.03.10.f wurde die Spaltenbreite noch einmal korrigiert und dabei die Option <i>opt_new_colwidth</i> eingeführt. Wenn <i>opt_new_colwidth</i> den Wert <i>true</i> hat, dann hat diese Einstellung Vorrang vor <i>opt_old_colwidth</i>.

Objekt	option_index	Bedeutung
	<i>opt_old_select</i> (nur IDM FÜR MOTIF, ab IDM-Version A.05.02.h)	Mit dieser Option kann eine auswahl- oder eine aktionsorientierte Verschickung von <i>select</i>-Ereignissen eingestellt werden. Dadurch wird ein konsistentes Verhalten der Motif- und Windows-Version des IDM ermöglicht. <i>true</i> <i>select</i>-Ereignisse werden bei Änderungen des Aktivierungszustands verschickt (auswahlorientiert). Dies entspricht dem Verhalten bis einschließlich IDM-Version A.05.02.g. <i>false</i> (Standard) <i>select</i>-Ereignisse werden bei Mausklick bzw. beim Betätigen der Selektionstaste verschickt (aktionsorientiert). Dies entspricht dem Verhalten unter Microsoft Windows. Damit zeigt das <i>select</i>-Ereignis keine Änderung des Aktivierungszustands mehr an. Um auch bei <i>.options[opt_old_select] = false</i> auf Änderungen der Auswahl zu reagieren, können die Ereignisse <i>activate</i> und <i>deactivate</i> genutzt werden.
	<i>opt_xmborder_compat</i> (nur IDM FÜR MOTIF, ab IDM-Version A.06.01.a)	Kompatibilitätsoption um das tablefield wieder mit einem 1 Pixel breiten Rahmen zu zeichnen. Ab IDM-Version A.06.01.a ist die Standardbreite für den Rahmen 0. Dies könnte zwar auch durch das Setzen von <i>.borderwidth = 1</i> erreicht werden, jedoch erscheint dann eine Warnmeldung, da <i>.borderwidth</i> für das tablefield zukünftig nicht mehr unterstützt wird.

Objekt	option_index	Bedeutung
toolbar	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
treeview	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_rightclick_selects</i> (nur IDM FÜR WINDOWS)	Bestimmt ob Einträge mit der rechten Maustaste selektiert werden können: false (Standard) Keine Selektion von Einträgen mit der rechten Maustaste. true Einträge können mit der rechten Maustaste selektiert werden.

Objekt	option_index	Bedeutung
	<i>opt_scroll_on_focus</i> (nur IDM FÜR MOTIF, ab IDM-Version A.05.02.i)	Steuert das Verhalten bei Fokussierung und kann die Erreichbarkeit per Tastaturnavigation verbessern (siehe Kapitel „Motif-Option .options[opt_scroll_on_focus]“). true (Standard) Der IDM versucht, ein Kindobjekt, das den Fokus erhalten soll, in den sichtbaren Bereich zu scrollen. false Der IDM versucht nicht, zu fokussierende Kindobjekte in den sichtbaren Bereich zu scrollen. Hinweis Der treeview kann bei <i>.options[opt_scroll_on_focus] = false</i> per Tastaturnavigation unerreichbar sein, falls sich der aktive Eintrag außerhalb des sichtbaren Bereichs befindet. Er bleibt aber per Mausklick fokussierbar.
window	<i>opt_animated</i> (nur IDM FÜR QT)	Schaltet Animationen beim interaktiven Verschieben von toolbars ein und aus. false (Standard) Keine Animationen beim Verschieben von toolbars. true Animiertes, interaktives Verschieben von toolbars. Dies führt allerdings zu deutlich mehr <i>resize</i>- und <i>move</i>-Ereignissen.

Objekt	option_index	Bedeutung
	<i>opt_auto_close</i> (nur IDM FÜR WINDOWS und IDM FÜR QT, IDM FÜR QT ab IDM-Version A.06.01.e)	Diese Option verhindert das automatische Schließen des Fensters, wenn der Anwender den Befehl „<i>Schließen</i>“ aus dem Systemmenü oder die „<i>Schließen</i>“-Schaltfläche in der Titelleiste auswählt. true (Standard) Fenster wird bei Auswahl von „<i>Schließen</i>“ automatisch geschlossen. false Fenster wird bei Auswahl von „<i>Schließen</i>“ nicht geschlossen sondern muss explizit durch die Anwendung geschlossen werden. Ein Verhalten wie beim Wert <i>true</i> lässt sich durch die Regel <pre>on WINDOW close { this.visible := false; } simulieren.</pre>
	<i>opt_center_toolhelp</i> (nur IDM FÜR WINDOWS)	Steuert die Anzeige der Toolhelp: false (Standard) Die Toolhelp wird an der Position des Mauszeigers angezeigt. true Die Toolhelp wird zentriert unterhalb des Objekts angezeigt, sofern genügend Platz vorhanden ist.
	<i>opt_help</i> (nur IDM FÜR WINDOWS)	Wenn diese Option gesetzt ist, wird in der Titelleiste die Hilfe-Schaltfläche mit dem Fragezeichen dargestellt. Hinweis Die Hilfe-Schaltfläche wird nicht angezeigt, wenn alle drei anderen Symbole (maximize, minimize, close) dargestellt werden.

Objekt	option_index	Bedeutung
	<i>opt_nested_docks</i> (nur IDM FÜR QT)	Ermöglicht „Nested Docks“, d.h. die Anordnung von toolbars mit <i>.style = notepage</i> in mehrreihigen Docks. false (Standard) Nur einreihige Docks sind möglich. true Nebeneinander liegende, mehrreihige Docks sind möglich. Dies führt allerdings zu einer nicht mehr so klaren Bedienung bei Interaktionen und Verschiebungen. Das Attribut <i>.dock_line</i> wird beachtet. Das „Einnisten“ in dasselbe Dock erfolgt immer relativ zur vorherigen Toolbar im selben Dockbereich. Hinweise » Die Option wirkt sich nur auf toolbars mit <i>.style = notepage</i> aus. » Sind sowohl <i>.options[opt_nested_docks] = false</i> als auch <i>.options[opt_tabbed_docks] = false</i> (Standard), dann wird das Attribut <i>.dock_line</i> von toolbars mit <i>.style = notepage</i> ignoriert und alle diese Toolbars werden nebeneinander (horizontal) oder übereinander (vertikal) angeordnet.
	<i>opt_scroll_on_focus</i> (nur IDM FÜR MOTIF, ab IDM-Version A.05.02.i)	Steuert das Verhalten bei Fokussierung und kann die Erreichbarkeit per Tastaturnavigation verbessern (siehe Kapitel „Motif-Option <i>.options[opt_scroll_on_focus]</i>“). true (Standard) Der IDM versucht, ein Kindobjekt, das den Fokus erhalten soll, in den sichtbaren Bereich zu scrollen. false Der IDM versucht nicht, zu fokussierende Kindobjekte in den sichtbaren Bereich zu scrollen.

Objekt	option_index	Bedeutung
	<i>opt_tabbed_docks</i> (nur IDM FÜR QT)	Ermöglicht „Tabbed Docks“, d.h. die Anordnung von toolbars mit <i>.style = notepage</i> als Registerkarten in einem Dock. false (Standard) „Tabbed Docks“ sind nicht möglich. true „Tabbed Docks“, die sich den Platz teilen, sind möglich. toolbars mit derselben <i>.dock_line</i> werden als gestapelte Registerkarten mit Reitern dargestellt. „Tabbed Docks“ lassen sich ähnlich wie Notebooks und Notepages bedienen. Hinweise » Die Option wirkt sich nur auf toolbars mit <i>.style = notepage</i> aus. » Sind sowohl <i>.options[opt_nested_docks] = false</i> als auch <i>.options[opt_tabbed_docks] = false</i> (Standard), dann wird das Attribut <i>.dock_line</i> von toolbars mit <i>.style = notepage</i> ignoriert und alle diese Toolbars werden nebeneinander (horizontal) oder übereinander (vertikal) angeordnet.
	<i>opt_window_size</i> (nur IDM FÜR QT)	Bestimmt, wie Größenangaben für das Fenster interpretiert werden. false (Standard) Größenangaben (<i>.width</i>, <i>.height</i>, Minimal- und Maximalwerte) beziehen sich auf den Innenbereich („client area“) des Fensters. true Größenangaben beziehen sich auf das gesamte Fenster, inklusive Menüleiste, Toolbars, Tabbed Widgets und Statusleiste, aber ohne Dekoration (Titelzeile, Ränder).

Beispiel

Mit Motif 1.1 können Sie nicht mittels der Tastaturnavigation über ein Objekt navigieren, wenn dieses Objekt ein „Composite-Widget“ ist oder dieses Widget keine Kinder hat. Es kann dabei auch zu einem Programmabsturz kommen.

```
Canvas1.options[opt_accept_child] := true;
```

Siehe auch

Objekt canvas

2.243.1 Motif-Option .options[opt_scroll_on_focus]

Objekte mit virtueller Größe (**groupbox**, **layoutbox**, **notepage**, **window**) und **treeview** verfügen ab IDM-Version A.05.02.i über die Option `.options[opt_scroll_on_focus]`. Mit ihr kann eingestellt werden, ob ein Kindobjekt, das den Fokus erhält, in den sichtbaren Bereich gescrollt wird.

Zweck der Option

Die Option `.options[opt_scroll_on_focus]` dient in erster Linie dazu, die Erreichbarkeit von Objekten per Tastaturnavigation zu verbessern und zu ermöglichen, dass sie in bestimmten Konstellationen per Tastatur oder Mausklick den Fokus erhalten können. Damit kann ein konsistenteres Verhalten zwischen Motif- und Windows-Anwendungen erreicht werden.

Allerdings kann durch die Option – aufgrund von Unterschieden zwischen den Plattformen bei der Fokus-Behandlung – weder eine vollständige Konsistenz erzielt werden, noch die Erreichbarkeit von Objekten per Tastatur und ihre Fokussierbarkeit in allen Situationen gewährleistet werden.

Zusammenhang zwischen Objektposition, Sichtbarkeit, Tastaturnavigation und Fokussierbarkeit unter Motif

Unter Motif können nur Objekte, die vollständig sichtbar sind oder in den sichtbaren Bereich gescrollt werden können, den Fokus erhalten. Dies ist das typische Verhalten von Motif-Anwendungen, im Unterschied zu MICROSOFT WINDOWS. Dabei ist zu beachten, dass ein Kindobjekt in einem Gruppierungsobjekt eventuell vollständig sichtbar ist, aber trotzdem per Tastatur unerreichbar und nicht fokussierbar ist, weil das Gruppierungsobjekt nicht vollständig sichtbar ist.

Aus Gründen der Kompatibilität zwischen den Plattformen ermöglicht es der ISA Dialog Manager auch unter Motif, Kindobjekte durch negative Positionsangaben für die x- und y-Koordinaten der linken oberen Ecke in nicht sichtbaren Bereichen zu positionieren. Motif erlaubt dies eigentlich nicht, sodass der IDM dafür Prüfmechanismen umgehen muss. Als Konsequenz wird dadurch meistens ausgeschlossen, dass die Objekte per Tastaturnavigation erreichbar sind und den Fokus erhalten können.

Empfehlungen

- » Der beste Weg, um die Erreichbarkeit von Objekten per Tastaturnavigation und ihre Fokussierbarkeit per Tastatur und Maus zu gewährleisten, besteht darin, sie immer vollständig im sichtbaren Bereich zu positionieren und negative Werte für ihre x- und y-Koordinaten zu vermeiden.
- » Wenn dies nicht möglich ist, kann versucht werden, die Tastaturnavigation und Fokussierbarkeit mithilfe von `.options[opt_scroll_on_focus]` zu verbessern.

2.243.2 Optionen für Gruppierungsobjekte unter Windows

Achtung

Die Optionen `opt_wntsizebug_compat` und `opt_w2kprefsize_compat` sind veraltet. Seit IDM-Version A.06.01.a werden sie nicht mehr ausgewertet.

Es wird dringend empfohlen, Dialoge, die noch auf diese Optionen angewiesen sind, anzupassen. Nur wenn keine Anpassung möglich ist, sollten die Startoption **-IDMborder5_compat** oder die Umgebungsvariable `IDM_BORDER5_COMPAT` eingesetzt werden, um die Auswertung von `opt_wntsizebug_compat` und `opt_w2kprefsize_compat` wieder zu reaktivieren (verbunden mit dem Nachteil, dass die Unterstützung des Attributs `.borderstyle` dabei verloren geht).

Bei der Umstellung zu „Visual Styles“ ab Windows XP wurde gleichzeitig auch eine Reihe von Größenberechnungsfehlern behoben. Diese Fehler existierten, seitdem 3D-Objekte von Windows unterstützt werden.

Insbesondere in Dialogen, bei denen es auf eine pixelgenaue Darstellung ankommt, hatten diese Fehlerbehebungen sehr große Auswirkungen. Deshalb wurden die Versionen des IDM ab XP „fehlerkompatibel“ gemacht. Zusätzlich wurden zwei neue Optionen für Gruppierungsobjekte (**groupbox**, **layoutbox**, **notebook**, **notepage**, **spinbox**, **splitbox**, **statusbar**, **toolbar** und **window**) eingeführt:

`.options[opt_wntsizebug_compat]`

Standardwert: *false*

Der Wert *false* bedeutet, dass Position und Größe aller direkten Kindobjekte richtig gesetzt werden. Der Wert *true* bedeutet, dass Position und Größe aller direkten Kindobjekte fehlerkompatibel gesetzt werden. Es gab nämlich in der Windows-Version schon seit „Urzeiten“ einen Fehler bei der Berechnung von Position und Größe der Objekte **edittext**, **groupbox**, **listbox**, **poptext** und **treeview**. Durch ihn wurden diese Objekte an jeder Seite um ein Pixel kleiner als gewünscht dargestellt.

Die Option ist mit dem Wert *false* vorbelegt.

`.options[opt_w2kprefsize_compat]`

Standardwert: *false*

Der Wert *true* bedeutet, dass die bevorzugte Größe aller direkten Kindobjekte so berechnet wird, als ob die Anwendung unter Windows 2000 laufen würde. Der Wert *false* bedeutet, dass in den

Versionen des IDM ab Windows XP die bevorzugte Größe aller direkten Kindobjekte entsprechend den „Visual Styles“ berechnet wird.

Die Option ist mit dem Wert *false* vorbelegt, sodass Objekte, die keine Größe gesetzt haben, auch bei aktiven „Visual Styles“ in der optimalen Größe erscheinen.

Sollte es das Aussehen von Dialogen „zerreißen“, kann man an den Default-Objekten diese Option anschalten und anschließend den Dialog sukzessive anpassen.

2.243.3 Optionen für die Rasterberechnung unter Windows

Mit der IDM-Version A.06.03.a wurde die Berechnung der Rasterbreite grundlegend umgestellt. Wenn kein Referenzstring angegeben ist, wird die Rasterbreite jetzt aus einem internen Referenzstring („M“) berechnet, um ein übermäßiges Breitenwachstum durch überbreite Buchstaben innerhalb einer Schriftart zu vermeiden.

Aus Kompatibilitätsgründen kann jedoch durch die Option *opt_fontraster_compat*, die Startoption **-IDMfontraster_compat** oder die Umgebungsvariable `IDM_FONTRASTER_COMPAT` vorübergehend die veraltete Berechnung der Rasterbreite wieder reaktivieren werden (verbunden mit dem Nachteil, dass es wieder zu übermäßigem Breitenwachstum kommen kann).

Bei Verwendung der Option *opt_fontraster_compat* basiert die Größenberechnung zum Teil auf dem Systemfont, der nicht High DPI fähig ist, daher sollten High DPI fähige Anwendungen, die mit IDM FÜR WINDOWS 11 erstellt wurden, dies Option nicht verwenden.

2.244 .order

Mit Hilfe dieses Attributes kann die Reihenfolge von Menüs und Menübefehlen definiert werden. Die Menüs bzw. Menübefehle werden in Reihenfolge wachsender *.order*-Werte in Menüboxen von oben nach unten und in Fenstern von links nach rechts sortiert. Bei gleichen Werten entscheidet die Definitionsreihenfolge.

Das Attribut wird nur vom IDM FÜR WINDOWS unterstützt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_order	Bezeichner: AT-order
Datentyp: DT_integer	Datentyp: DT-integer

Standardwert
0

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	menubox, menuitem, menusep

Bei der Gruppierung von Menüeinträgen im Radiobutton-Stil gilt, dass wenn ein aktiver Eintrag durch Umsetzen des *.order*-Wertes die Gruppe wechselt, dieser Eintrag aktiv bleibt und der zuvor in dieser Gruppe aktive Eintrag deaktiviert wird. Dies ist auch beim statischen Anlegen von Menüs zu berücksichtigen.

Es ist empfehlenswert, sich bei der Vergabe der *.order*-Werte an die MICROSOFT-Konventionen für „gemischte“ Menüleisten zu halten. Gemischte Menüleisten setzen sich bei der **InPlace**-Darstellung von **ActiveX**-Controls aus den Menüs von Container und Control zusammen. Die Konventionen sehen für bestimmte Gruppen von Menüboxen feste Werte vor und zwar:

File	0
Edit	1
Container	2
Object	3
Window	4
Help	5

Dabei werden die Gruppen 0, 2 und 4 vom Container und die Gruppen 1, 3 und 5 vom **ActiveX**-Control in die gemeinsame Menüleiste übernommen.

Der IDM unterstützt zwar keine gemischten Menüleisten, doch wenn man die **InPlace**-Darstellung von **ActiveX**-Controls verwendet, sollte man die MICROSOFT-Konventionen berücksichtigen.

Hinweis

Es ist durchaus üblich, dass eine Menügruppe aus mehreren Menüs besteht, z.B. die Menügruppe „Edit“ aus „Bearbeiten“ und „Suchen“ (bei TEXTPAD).

2.245 .output[integer]

Mit Hilfe dieses Attributes kann bei Funktionen und Regeln erfragt werden, ob es sich bei dem Parameter um einen Ausgabeparameter handelt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>boolean</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_output	Bezeichner: AT-output
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	function, rule

Beispiel

```
function Test (integer P input, integer Q output,  
              string R input output)  
  
for I := 1 to Test.count do  
  print Test.output[I];  
endfor
```

Ausgabe

```
false  
true  
true
```

Siehe auch

Kapitel „Funktionen (functions)“ und „Benannte Regeln (Unterprogramme)“ im Handbuch „Regelsprache“

2.246 .overridecursor

Mit diesem Attribut des Setup-Objekts kann global ein temporärer Cursor für alle geladenen Dialoge gesetzt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [cursor]	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_overridecursor	Bezeichner: AT-overridecursor
Datentyp: DT_cursor	Datentyp: DT-cursor

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Dieser Cursor wird für alle Dialoge oder Fenster angezeigt, die das Attribut *ignorecursor* nicht auf *true* gesetzt haben. Der temporäre Cursor lässt sich abschalten, indem *overridecursor* auf *null* gesetzt wird.

Mit dem Attribut kann sehr einfach ein Warte-Cursor eingebunden werden, um bei länger andauernden Aktionen ein visuelles Feedback bei den noch offenen Fenstern zu geben. Sinnvoll ist der Einsatz in Verbindung mit einer Dialogbox oder MessageBox.

Hinweis

Das Objekt **messagebox** ignoriert den bei Setup gesetzten *overridecursor* und zeigt immer den für die MessageBox definierten Cursor oder den Default-Cursor an.

Beispiel

Beim Betätigen eines Buttons soll der Warte-Cursor für alle offenen Fenster der laufenden Dialoge gesetzt werden, außer für das Fenster, in dem sich der Button befindet. Dieses wird zu einer Dialogbox gemacht, um Eingaben in andere Fenster zu sperren.

```
on PbCompute select
{
  this.window.dialogbox      := true;
  this.window.ignorecursor   := true;
  setup.overridecursor       := CursorBusy;
  Compute();
  setup.overridecursor       := null;
  this.window.ignorecursor   := false;
  this.window.dialogbox      := false;
}
```

Siehe auch

Attribut *.ignorecursor*

Objekt messagebox

2.247 .pagemotion

Dieses Attribut definiert den Betrag, um den sich der Scrollbarwert (*.curvalue*) ändert, wenn der Benutzer seitenweise scrollt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_pagemotion	Bezeichner: AT-pagemotion
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	scrollbar

2.248 .parent

Mit Hilfe dieses Attributes kann der Vater eines Objektes erfragt oder verändert werden.

Definition

Datentyp
object

Zugriff
get, set

C
Bezeichner: AT_parent
Datentyp: DT_object

COBOL
Bezeichner: AT-parent
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

Siehe auch

Methode :parent_of()

2.249 .password

In diesem Attribut kann das Passwort definiert werden, das zum Starten der Anwendungsseite beim RSH- oder SSH-Protokoll verwendet wird.

Das Attribut wird ausgewertet, wenn beim Kommando im *.exec*-Attribut kein Passwort angegeben ist. Das Attribut stellt eine Alternative dar, wenn es Probleme bereitet, das Kommando im *.exec*-Attribut korrekt aufzubauen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_password	Bezeichner: AT-password
Datentyp: DT_string	Datentyp: DT-string

Vererbung
ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Verfügbarkeit

Ab IDM-Version A.06.02.g

Siehe auch

Attribut *.username*

2.250 .path

Liefert eine String-Repräsentation für die Position des ***XML-Cursors*** im DOM-Baum. Mit diesem String kann die ***:select()***-Methode aufgerufen werden, um einen ***XML-Cursor*** auf den Knoten im DOM-Baum zu positionieren.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	<i>get</i>

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_path	Bezeichner: AT-path
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	doccursor

Wird der Wert des *.path*-Attributs woanders gemerkt (zum Beispiel im *.userdata*-Attribut), dann ist zu beachten, dass dieser gemerkte Wert nicht angepasst wird, wenn die Struktur des DOM-Baums verändert wird. Ein anschließender Aufruf der Methode ***:select()*** mit diesem gemerkten Wert, wird demzufolge den ***XML-Cursor*** auf einen falschen DOM-Knoten zeigen lassen.

2.251 .pattern

Dieses Attribut dient dazu, die Auswahlmöglichkeiten in einem **filereq**-Objekt durch ein Muster einzustellen. In der Auswahlliste des Filerequestors werden nur Dateien bzw. Verzeichnisse angezeigt, deren Name diesem Muster entspricht. Bei einem leeren String als Muster werden alle Dateien (Motif *, Windows *.*) angezeigt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_pattern	Bezeichner: AT-pattern
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	filereq

Besonderheiten Motif

Hier handelt es sich bei dem Muster um einen regulären Ausdruck, der vom Benutzer verändert werden kann. Der Funktionsumfang ist von Motif vorgegeben, daher sind im Folgenden nur die wichtigsten Ausdrücke genannt:

Ausdruck	Bedeutung
*	keins oder mehrere beliebige Zeichen
?	ein beliebiges Zeichen
[a-z]	ein Zeichen aus dem Bereich a-z, also ein kleiner Buchstabe
[0123]	ein Zeichen aus der Aufzählung 0, 1, 2, 3

Besonderheiten Microsoft Windows

Das Muster ist nur im Modus *fr_load* und *fr_save* wirksam. Windows erwartet eine Liste von Typbezeichnern mit Musterlisten. Der Benutzer bekommt nur die Typbezeichner angeboten, ein eigenes Muster kann er nicht angeben. Typbezeichner und Muster werden durch ein Tabulatorzeichen "\t" getrennt, in den Musterlisten dient das Semikolon (Strichpunkt) als Trennzeichen.

Beispiel

```
.pattern "IDM-Dateien\t*.idm;*.dlg;*.mod\tDokumente
(*.doc)\t*.doc;\tAlle Dateien (*.*)\t*.*";
```

2.252 .picheight

Dieses Attribut definiert bei **listbox** und **poptext** die Höhe der Bilder an den einzelnen Listeneinträgen. Beim **listview** definiert es die Höhe der großen Symbole.

Beim **notebook** definiert das Attribut die Höhe der Bilder in den Reitern der einzelnen **notepages**.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_picheight	Bezeichner: AT-picheight
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, listview, notebook, poptext

listbox, poptext

Das Attribut definiert die Höhe der Fläche links neben den Einträgen, in der die Bilder (**tile**-Ressourcen) angezeigt werden, die den einzelnen Einträgen in den Attributen *.picture[integer]* und *.picture_hilite[integer]* zugeordnet wurden.

Höhere oder niedrigere Bilder werden in dieser Fläche vertikal zentriert. Bei **tile**-Ressourcen mit *.scale = true* wird deren Höhe auf *.picheight* skaliert.

listview

Mit diesem Attribut wird die Höhe des großen Symbols festgelegt.

Wertebereich

- 0** Die Breite *.picwidth* wird verwendet. Ist diese ebenfalls 0, dann wird die Größe des Symbols vom System bestimmt.
- > 0** Höhe des großen Symbols.

Das große Symbol wird in der Symbolansicht (*.style = "icon"*, *.style = "picture"*) und in der Kachelansicht (*.style = "tile"*) angezeigt.

Bei ungültigen Werten wird der Standardwert 0 verwendet. Der Attributwert wird allerdings nicht geändert.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

notebook

Das Attribut definiert die Höhe der Bilder, die in den Reitern angezeigt werden. Die anzuzeigenden **tile**-Ressourcen werden den **notepages** in ihrem **.picture**-Attribut zugeordnet.

Der Wertebereich ist **.picheight** ≥ 0 . Bei **.picheight** = 0 wird die Höhe vom System bestimmt und beträgt 16 Pixel.

Die Höhe der Bilder wird unabhängig vom **.scale**-Attribut der **tile**-Ressource auf **.picheight** skaliert.

Das Attribut ist nur unter MICROSOFT WINDOWS verfügbar.

2.253 .picture

Das *.picture*-Attribut definiert das Bild (**tile**-Ressource), das in einem **image**-Objekt mit *.style = 0* (*pushbutton*, Standardwert) angezeigt wird.

Bei der **notepage** definiert das Attribut das Bild, das neben der Beschriftung im zur **notepage** gehörenden Reiter des **Notebooks** angezeigt wird.

Beim **control**-Objekt definiert dieses Attribut das Bild, das im inaktiven Zustand des Objekts angezeigt wird.

Definition

Datentyp	Zugriff	changed-Ereignis
object [tile]	get, set	ja

C	COBOL
Bezeichner: AT_picture	Bezeichner: AT-picture
Datentyp: DT_tile	Datentyp: DT-tile

Klassifizierung	Objekte
Objektspezifisches Attribut	control, image, notepage

image (Bild)

Durch Setzen des Attributs *.mouseover* auf *true* kann das **image**-Objekt andere Bilder anzeigen, wenn der Mauszeiger über das Objekt bewegt bzw. die linke Maustaste über ihm gedrückt wird. Diese Bilder werden im *.picture[enum]*-Attribut mit den Indizes *tile_mouse_over* bzw. *tile_active_mouse_over* definiert.

Bei *.style <> 0* werden die in unterschiedlichen Situationen angezeigten Bilder ebenfalls im indizierten *.picture[enum]*-Attribut definiert.

Anstelle einer **tile** Ressource kann dem *.picture*-Attribut auch ein String mit dem Dateipfad einer Grafikdatei zugewiesen werden. Diese Möglichkeit kann in zukünftigen IDM-Versionen allerdings entfallen. Daher wird empfohlen, eine **tile**-Ressource mit einer externen Grafikdatei zu verwenden.

Hinweis zum IDM für Qt

Die Angabe eines Dateipfads (Datentyp *string*) wird **nicht** unterstützt.

Siehe auch

Attribute *.mouseover*, *.picture[enum]*, [style](#)

Hinweis zur notepage

Die Größe des Bildes wird mit den Attributen *.picheight* und *.picwidth* des **notebooks** für alle enthaltenen Notepages festgelegt.

Siehe auch

Ressource tile

2.254 .picture[enum]

Mit diesem Attribut können unterschiedliche Bilder definiert werden, die im Objekt je nach dessen Stil, Aktivierungszustand und Situation angezeigt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [tile]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_picture	Bezeichner: AT-picture
Datentyp: DT_tile	Datentyp: DT-tile

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	image, menuitem

Indexbereich

tile_default

tile-Ressource, die im „normalen“ Zustand des Objekts angezeigt wird.

tile_active

tile-Ressource, die angezeigt wird wenn das Objekt aktiviert ist.

tile_insensitive (nur **image**)

tile-Ressource, die angezeigt wird, wenn das Objekt nicht bedienbar (insensitiv) ist.

tile_mouse_over (nur **image**)

tile-Ressource, die angezeigt wird, wenn sich der Mauszeiger über dem Objekt befindet und die linke Maustaste nicht gedrückt bzw. das Objekt nicht aktiviert ist.

Das Attribut *.mouseover* muss *true* sein, damit dieses Bild angezeigt wird.

tile_active_mouse_over (nur **image**)

tile-Ressource, die angezeigt wird, wenn sich der Mauszeiger über dem Objekt befindet und die linke Maustaste gedrückt bzw. das Objekt aktiviert ist.

Das Attribut *.mouseover* muss *true* sein, damit dieses Bild angezeigt wird.

image

Beim **image**-Objekt werden im Attribut *.picture[enum]* die Bilder bzw. **tile**-Ressourcen angegeben, die in unterschiedlichen Situationen angezeigt werden. Welches Bild angezeigt wird, hängt von mehreren Faktoren ab, die in der folgenden Tabelle beschrieben sind.

Abgesehen von den Ausnahmen, die in der Tabelle genannt sind, wird kein anderes Bild angezeigt, wenn das Bild für einen Zustand fehlt. Deshalb müssen immer alle benötigten Bilder bzw. **tile**-Ressourcen definiert sein.

Index	Wird angezeigt	Notwendig
<i>tile_default</i>	Im normalen Zustand, wenn keine der folgenden Situationen vorliegt.	Immer.
<i>tile_active</i>	Bei <i>.style = checkbox</i>, wenn » das Image-Objekt aktiv ist. » das Image-Objekt nicht aktiv ist und die linke Maustaste über dem Objekt gedrückt wird. Bei <i>.style = pushbutton</i>, wenn die linke Maustaste über dem Objekt gedrückt wird. Ausnahme: In diesem Fall wird <i>tile_default</i> angezeigt, falls <i>tile_active</i> nicht gesetzt ist. Bei <i>.style = menubox</i> temporär und unabhängig von <i>.mouseover</i>, wenn das Kontextmenü geöffnet und das Image-Objekt nicht das eigentliche Menüobjekt ist. <i>tile_active</i> wird nicht angezeigt, wenn <i>.mouseover = true</i> ist und sich der Mauszeiger über dem Image-Objekt befindet.	Bei den Stilen <i>checkbox</i> und <i>menubox</i> .
<i>tile_insensitive</i>	Wenn das Image- Objekt insensitiv ist (<i>.real_sensitive = false</i>). Ausnahme: Ist <i>tile_insensitive</i> nicht gesetzt, wird <i>tile_default</i> bzw. <i>tile_active</i> (bei <i>.style = checkbox</i> und aktivierter Checkbox) angezeigt.	Bei allen Stilen optional.

Index	Wird angezeigt	Notwendig
<i>tile_mouse_over</i>	Wenn <i>.mouseover = true</i> ist und sich der Mauszeiger über dem Image-Objekt befindet und » die linke Maustaste nicht gedrückt ist und. » bei <i>.style = checkbox</i> die Checkbox nicht aktiviert ist. Bei <i>.style = menubox</i> unabhängig von <i>.mouseover</i>, wenn das Kontextmenü geschlossen und das Image-Objekt das eigentliche Menüobjekt ist. In diesem Zustand wird der Rahmen hervorgehoben dargestellt.	Bei <i>.style = menubox</i>. Sonst nur, wenn <i>.mouseover = true</i> ist.
<i>tile_active_mouse_over</i>	Wenn <i>.mouseover = true</i> ist und sich der Mauszeiger über dem Image-Objekt befindet und » die linke Maustaste gedrückt ist. » bei <i>.style = checkbox</i> die Checkbox aktiviert ist. Bei <i>.style = menubox</i> unabhängig von <i>.mouseover</i>, wenn das Kontextmenü geöffnet ist und das Image-Objekt das eigentliche Menüobjekt ist.	Bei <i>.style = menubox</i>. Bei <i>.style = checkbox</i>, wenn <i>.mouseover = true</i> ist.

Siehe auch

Attribute *.mouseover*, *.picture*

menuitem

Beim **menuitem** kann das Attribut *.picture[enum]* verschiedene Bilder enthalten, die je nach Stil (Attribut *.style*) und Aktivierungszustand als Symbol bei einem Menüeintrag angezeigt werden. Das Attribut ist mit einem Aufzählungstyp indiziert, als Indexwerte sind *tile_default* und *tile_active* zugelassen.

Hat das **menuitem** den Stil *radiobutton* oder *checkbox*, wird das Bild in *tile_active* angezeigt, wenn der Radiobutton oder die Checkbox aktiviert (eingeschaltet) ist. Sind sie nicht aktiviert, wird das Bild in *tile_default* dargestellt. Der IDM schaltet automatisch zwischen beiden Bildern um.

Bei einem „normalen“ Menüeintrag – also mit *.style = pushbutton* – wird immer das Bild in *tile_default* angezeigt.

Siehe auch

Attribut *.style*

2.255 .picture[integer]

Das Attribut definiert die Bilder (**tile**-Ressourcen), die links neben den einzelnen Einträgen angezeigt werden.

Beim **listview** definiert das Attribut die großen Symbole für die Listeneinträge.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [tile]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_picture	Bezeichner: AT-picture
Datentyp: DT_tile	Datentyp: DT-tile

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, listview, poptext, treeview

listbox, poptext

Der Wertebereich für den Index ist *1itemcount*. Der jeweilige Eintrag *.content[l]* bei der **listbox** bzw. *.text[l]* beim **poptext** muss bereits existieren, bevor auf das Attribut *.picture[l]* mit dem korrespondierenden Index zugegriffen werden kann.

Bei selektierten Einträgen wird – sofern vorhanden – das in *.picture_hilite[integer]* definierte Bild dargestellt (vom IDM FÜR MOTIF **nicht** unterstützt).

Die Größe der Fläche, die links von den Einträgen für die Bilder zur Verfügung steht, wird mit den Attributen *.picwidth* und *.picheight* festgelegt.

An der **listbox** wird das Attribut **nicht** vererbt.

Bei einem **poptext** mit *.style = edittext* oder *.style = listbox* wird im Eingabefeld **kein** Bild angezeigt.

Hinweis zum IDM für Qt

Wenn bei **listbox** und **poptext** nur bei *.picture_hilite[l]* ein **tile** gesetzt ist, aber nicht bei *.picture[l]*, dann zeigt Qt auch im unselektierten Zustand das in *.picture_hilite[l]* definierte **tile** an (oder eine leichte Abwandlung davon). Qt versucht für die Listeneinträge immer Bilder für alle Zustände bereitzustellen. Fehlt das Bild für einen Zustand, generiert Qt es aus den bereitgestellten Bildern.

listview

Das Attribut definiert das große Symbol für jeden Listeneintrag.

Der Wertebereich des Index ist *0rowcount*, wobei der Wert mit Index *0* als Standardwert für nicht gesetzte Werte im Bereich *1rowcount* dient.

Wertebereich

- null** Das angezeigte Symbol ergibt sich wie folgt:
- » Wenn `.picture[0] <> null` ist, wird das dort definierte Symbol verwendet.
 - » Wenn `.picture[0] = null` ist, dann wird das kleine Symbol vergrößert dargestellt (je nach Vorhandensein `.smallpicture[1]` oder `.smallpicture[0]`).
 - » Wenn weder ein großes noch ein kleines Symbol vorhanden sind, dann wird der Eintrag ohne Symbol dargestellt.
- tile** Ressource, die das große Symbol enthält.

Das große Symbol wird in der Symbolansicht (`.style = "icon"`, `.style = "picture"`) und in der Kachelansicht (`.style = "tile"`) angezeigt.

Großes und kleines Symbol gehören zusammen und sollten dieselbe Information darstellen. Beide Symbole können nur gemeinsam referenziert werden. Deshalb sind unterschiedliche Kombinationen zu vermeiden, da jede Kombination Speicherplatz verbraucht.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

treeview

Der Wertebereich für den Index ist $0 \dots .itemcount$. Die **tile** Ressource in `.picture[0]` definiert einerseits die Größe der Bilder, andererseits wird sie bei allen Einträgen angezeigt, für die `.picture[1]` mit $1 \leq I \leq .itemcount$ nicht gesetzt (*null*) ist. Alle anderen Bilder werden unabhängig vom `.scale`-Attribut der **tile**-Ressource auf die Größe von `.picture[0]` skaliert.

Wenn `.picture[0]` nicht gesetzt (*null*) ist, wird die Größe der Bilder vom System bestimmt und beträgt 16x16 Pixel. Bei Einträgen ohne Bild wird dann ein Ersatzsymbol (weißes Kreuz auf rotem Hintergrund) angezeigt.

Bevor auf das Attribut `.picture[1]` mit $1 \leq I \leq .itemcount$ zugegriffen werden kann, muss der Eintrag `.content[1]` mit dem jeweiligen Index bereits existieren.

Links von den Einträgen wird nur dann Platz für Bilder vorgesehen, wenn `.picture[1]` für mindestens einen Index gesetzt ist.

Das Attribut wird **nicht** vererbt.

Das Attribut wird vom IDM FÜR MOTIF **nicht** unterstützt.

Anmerkung zum IDM für Windows

Bei Verwendung einer „Windows Icon Resource“ in `.picture[0]` haben die Bilder immer die Standardgröße von 16x16 Pixel.

2.256 .picture_hilite[integer]

Das Attribut definiert die Bilder (**tile**-Ressourcen), die neben selektierten Einträgen angezeigt werden.

Der Wertebereich für den Index ist $1 \dots \text{itemcount}$. Der jeweilige Eintrag `.content[i]` bei der **listbox** bzw. `.text[i]` beim **poptext** muss bereits existieren, bevor auf das Attribut `.picture_hilite[i]` mit dem korrespondierenden Index zugegriffen werden kann.

Bei nicht selektierten Einträgen wird – sofern vorhanden – das in `.picture[integer]` definierte Bild dargestellt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [tile]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_picture_hilite	Bezeichner: AT-picture-hilite
Datentyp: DT_tile	Datentyp: DT-tile

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, poptext

Die Größe der Fläche, die links von den Einträgen für die Bilder zur Verfügung steht, wird mit den Attributen `.picwidth` und `.picheight` festgelegt.

An der **listbox** wird das Attribut **nicht** vererbt.

Bei einem **poptext** mit `.style = edittext` oder `.style = listbox` wird im Eingabefeld **kein** Bild angezeigt.

Das Attribut wird vom IDM FÜR MOTIF **nicht** unterstützt.

Hinweis zum IDM für Qt

Wenn bei **listbox** und **poptext** nur bei `.picture_hilite[i]` ein **tile** gesetzt ist, aber nicht bei `.picture[i]`, dann zeigt Qt auch im unselektierten Zustand das in `.picture_hilite[i]` definierte **tile** an (oder eine leichte Abwandlung davon). Qt versucht für die Listeneinträge immer Bilder für alle Zustände bereitzustellen. Fehlt das Bild für einen Zustand, generiert Qt es aus den bereitgestellten Bildern.

2.257 .picwidth

Dieses Attribut definiert bei **listbox** und **poptext** die Breite der Bilder an den einzelnen Listeneinträgen. Beim **listview** definiert es die Breite der großen Symbole.

Beim **notebook** definiert das Attribut die Breite der Bilder in den Reitern der einzelnen **notepages**.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_picwidth	Bezeichner: AT-picwidth
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, listview, notebook, poptext

listbox, poptext

Das Attribut definiert die Breite der Fläche links neben den Einträgen, in der die Bilder (**tile**-Ressourcen) angezeigt werden, die den einzelnen Einträgen in den Attributen *.picture[integer]* und *.picture_hilite[integer]* zugeordnet wurden.

Breitere oder schmalere Bilder werden in dieser Fläche horizontal zentriert. Bei **tile**-Ressourcen mit *.scale = true* wird deren Breite auf *.picwidth* skaliert.

listview

Mit diesem Attribut wird die Breite des großen Symbols festgelegt.

Wertebereich

- 0** Die Breite des Symbols wird vom System bestimmt.
- > 0** Breite des großen Symbols.

Das große Symbol wird in der Symbolansicht (*.style = "icon"*, *.style = "picture"*) und in der Kachelansicht (*.style = "tile"*) angezeigt.

Bei ungültigen Werten wird der Standardwert **0** verwendet. Der Attributwert wird allerdings nicht geändert.

Wenn *.picheight = 0* ist, bestimmt *.picwidth* auch die **Höhe** des großen Symbols.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

notebook

Das Attribut definiert die Breite der Bilder, die in den Reitern angezeigt werden. Die anzuzeigenden **tile**-Ressourcen werden den **Notepages** in ihrem *.picture*-Attribut zugeordnet.

Der Wertebereich ist *.picwidth* ≥ 0 . Bei *.picwidth* = 0 wird die Breite vom System bestimmt und beträgt 16 Pixel.

Die Breite der Bilder wird unabhängig vom *.scale*-Attribut der **tile**-Ressource auf *.picwidth* skaliert.

Das Attribut ist nur unter MICROSOFT WINDOWS verfügbar.

2.258 .pointer_height

Mit Hilfe dieses Setup-Objekt-Attributes kann die maximale Mauscursorhöhe (in Pixel) abgefragt werden.

Es wird bei Multiscreendialogen (nur Motif) der Wert für den Default-Screen zurückgegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

C
Bezeichner: AT_pointer_height
Datentyp: DT_integer

COBOL
Bezeichner: AT-pointer-height
Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

2.259 .pointer_height[integer]

Mit Hilfe dieses Setup-Objekt-Attributes kann die maximale Mauscursorhöhe (in Pixel) abgefragt werden.

Das indizierte Attribut ist nur bei Multiscreendialogen (nur Motif) verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_pointer_height	Bezeichner: AT-pointer-height
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung. Es ist zu beachten, dass mit Screen-Index nicht die Screen-Nummer gemeint ist, die z.B. mit dem Programm **xdpyinfo** erhalten wird.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

2.260 .pointer_width

Mit Hilfe dieses Setup-Objekt-Attributes kann die maximale Mauscursorbreite (in Pixel) abgefragt werden.

Es wird bei Multiscreendialogen (nur Motif) der Wert für den Default-Screen zurückgegeben.

In multiscreen systems (IDM for Motif only) the attribute returns the value for the default screen.

Definition

Datentyp
integer

Zugriff
get

C
Bezeichner: AT_pointer_width
Datentyp: DT_integer

COBOL
Bezeichner: AT-pointer-width
Datentyp: DT-integer

Klassifizierung
Objektspezifisches Attribut

Objekte
setup

2.261 .pointer_width[integer]

Mit Hilfe dieses Setup-Objekt-Attributes kann die maximale Mauscursorbreite (in Pixel) abgefragt werden.

Das indizierte Attribut ist nur bei Multiscreendialogen (nur Motif) verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_pointer_width	Bezeichner: AT-pointer-width
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung. Es ist zu beachten, dass mit Screen-Index nicht die Screen-Nummer gemeint ist, die z.B. mit dem Programm **xdpyinfo** erhalten wird.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

2.262 .posraster

Dieses Attribut bestimmt die Interpretation der Attributwerte für die Position eines Objektes (*.xleft*, *.xright*, *.ytop*, *.ybottom*). Es kann genutzt werden, um die Position eines Objektes in Abhängigkeit vom benutzten Raster oder genau in Pixeln anzugeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C

Bezeichner: AT_posraster

Datentyp: DT_boolean

COBOL

Bezeichner: AT-posraster

Datentyp: DT-boolean

Klassifizierung

Geometrieattribut

Wertebereich

true

Die Werte der Positionsattribute werden als Rastereinheiten interpretiert. Das zugrundeliegende Raster ist das Raster des Vaterobjektes.

false

Die Werte der Positionsattribute werden als Pixeleinheiten interpretiert. Das Raster des Vaterobjektes wird ignoriert.

Siehe auch

Attribut *.sizeraster*

Kapitel „Geometrieattribute“

2.263 .preedit

Dieses Attribut des **window**-Objekts steuert Anzeige und Auswahl des Eingabemodus für die **edit-texte** innerhalb des Fensters.

Anzeige und Auswahl des Eingabemodus unterstützen die Vorbearbeitung von symbolischen (ideografischen) Zeichen, die aus mehreren Zeichen eines phonetischen Alphabets zusammengesetzt sind, wie es bei vielen asiatischen Sprachen der Fall ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_preedit	Bezeichner: AT-preedit
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Wertebereich

preedit_root(Standard)

Eingabemodus-Anzeige nur wenn man sich innerhalb des Edittextes befindet

preedit_overspot

Permanente Eingabemodus-Anzeige am unteren Rand des Fensters

Hinweis

Das Fenster wird dadurch größer.

preedit_offthespot (nicht unterstützt)

Wird vom IDM nicht unterstützt.

preedit_onthespot (nicht unterstützt)

Wird vom IDM nicht unterstützt.

Verfügbarkeit

Nur IDM FÜR MOTIF mit Unicode-Unterstützung.

2.264 .preeditssel

Das Attribut *.preeditssel* steuert die initiale Markierung des Textes, wenn der Edittext eines **tablefields** explizit oder implizit aktiviert wird. Eine implizite Aktivierung des Edittextes erfolgt durch Eingabe eines Zeichens, eine explizite Aktivierung kann zum Beispiel per Doppelklick erfolgen.

Mithilfe von *.preeditssel* kann für beide Aktivierungsformen dieselbe initiale Textselektion eingestellt und dasselbe Eingabeverhalten erreicht werden. Als Eingabeverhalten sind das Ersetzen des vorhandenen Textes durch die Eingabe, sowie das Einfügen der eingegebenen Zeichen vor oder nach dem Inhalt des Edittextes möglich.

Definition

<i>Datentyp</i> enum	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
C Bezeichner: AT_preeditssel Datentyp: DT_enum		COBOL Bezeichner: AT-preeditssel Datentyp: DT-enum
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield	

Wertebereich

presel_default(Standard)

Bei expliziter Aktivierung ersetzt ein eingegebenes Zeichen den Inhalt.

Bei impliziter Aktivierung wird ein eingegebenes Zeichen an den Inhalt angehängt.

presel_all

Sowohl bei expliziter als auch bei impliziter Aktivierung wird der Inhalt vollständig – von *.startsel = 0* bis *.endsel = length(content)* – markiert.

Ein eingegebenes Zeichen ersetzt den Inhalt.

presel_begin

Keine Markierung, der Cursor wird sowohl bei expliziter als auch bei impliziter Aktivierung an den Anfang gesetzt.

Ein eingegebenes Zeichen wird vor dem Inhalt eingefügt.

presel_end

Keine Markierung, der Cursor wird sowohl bei expliziter als auch bei impliziter Aktivierung ans Ende gesetzt.

Ein eingegebenes Zeichen wird an den Inhalt angehängt.

Verfügbarkeit

Das Attribut wird ab IDM-Version A.05.02.h auf den Plattformen UNIX mit MOTIF und MICROSOFT WINDOWS unterstützt.

2.265 .privatekeyfile

In diesem Attribut wird die Datei mit dem privaten Schlüssel definiert, der für das SSH-Protokoll verwendet wird.

Standardmäßig wird die Datei **private.pem** aus dem Installationsverzeichnis der Anwendung verwendet.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_privatekeyfile	Bezeichner: AT-privatekeyfile
Datentyp: DT_string	Datentyp: DT-string

Vererbung
ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Verfügbarkeit

Ab IDM-Version A.06.02.h

Siehe auch

Attribut .publickeyfile

2.266 .propscale

Das Attribut steuert, ob das horizontale und vertikale Raster proportional auf den maximalen Wert gesetzt werden soll, welcher durch die Werteberechnung von `xraster` und `yraster` des Font-Rasters ermittelt wird.

Die genaue Berechnung des Raster ist im Kapitel „Berechnung der Rastergröße aus einem Referenzfont“ *Calculating the Grid Size from a Reference Font* der font-Ressource aufgeführt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_propscale	Bezeichner: AT-propscale
Datentyp: DT_boolean	Datentyp: DT-boolean

Standardwert
false

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	font

Wertebereich

true
Breite und Höhe des Font-Rasters werden auf den gleichen Wert ermittelt. Der Wert ergibt sich dabei aus dem Maximum der bisherigen Werteberechnung von `xraster` und `yraster`.

false
Breite und Höhe des Font-Rasters werden nicht angepasst und behalten die ausgehend vom Font ermittelten Werte.

Verfügbarkeit

Ab IDM-Version A.06.03.a

Siehe auch

Kapitel „HighDPI Unterstützung“ im Handbuch „Programmiertechniken“

2.267 .publicid

Öffentliche Kennung des DOM-Knotens. Das Attribut ist nur verfügbar, wenn der nodetype entweder *nodetype_entity* oder *nodetype_notation* ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	<i>get</i>

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_publicid	Bezeichner: AT-publicid
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	doccursor

Das Attribut ist für den XML-Cursor verfügbar. Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

Siehe auch

Attribut *.nodetype*

2.268 .publickeyfile

In diesem Attribut wird die Datei mit dem öffentlichen Schlüssel definiert, der für das SSH-Protokoll verwendet wird.

Standardmäßig wird die Datei **public.pem** aus dem Installationsverzeichnis der Anwendung verwendet.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_publickeyfile	Bezeichner: AT-publickeyfile
Datentyp: DT_string	Datentyp: DT-string

Vererbung
ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Verfügbarkeit

Ab IDM-Version A.06.02.h

Siehe auch

Attribut .privatekeyfile

2.269 .real_height

Mit Hilfe dieses Attributes wird erfragt, wie hoch das Objekt tatsächlich ist (Angabe in Pixel). Dazu muss das Objekt jedoch sichtbar sein (*.real_visible = true*).

Diese Höhe kann für Objekte ohne Höhendefinition erfragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_real_height	Bezeichner: AT-real-height
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Anmerkung

Wurde bei der Definition von Objekten keine Breite und/oder Höhe angegeben, so stehen die Attribute *.real_width* bzw. *.real_height* zur Verfügung, um die Breite und Höhe solcher Objekte zu erfragen.

Mit Hilfe dieser Attribute ist es z.B. möglich, die Breite eines Objektes zu erfragen, dessen Geometrie mit Hilfe einer Fixierung am linken und rechten Rand eines Vaterobjekts (*.xauto = 0*) definiert wurde.

Diese Angaben erfolgen immer in Pixel!

Siehe auch

Attribute *.height*, *.real_width*

2.270 .real_modified

Dieses Attribut zeigt an, ob der Inhalt (*.content*) eines **edittextes** oder **poptextes** geändert wurde, seitdem das Objekt den Fokus erhalten hat.

Beim **datetime**-Objekt gibt dieses Attribut an, ob das Attribut *.value* seit dem Erhalt des Fokus geändert wurde.

Das Attribut kann nur für sichtbare Objekte (*.real_visible = true*) abgefragt werden.

Beim **edittext** und beim **datetime**-Objekt wird das Attribut auf *false* gesetzt, wenn dem Attribut *.content* bzw. *.value* ein Wert zugewiesen wird.

Der **poptext** unterstützt das Attribut nur in den Stilen *.style = edittext* und *.style = listbox*.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>boolean</i>	<i>get</i>

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_real_modified	Bezeichner: AT-real-modified
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	datetime, edittext, poptext

Beispiel

Prüfen, ob der Inhalt bei einem *charinput* tatsächlich geändert wurde:

```
edittext Et
{
  on charinput
  {
    if this.real_modified then
      print "Content of " + this + " has changed";
    endif
  }
}
```

Hinweis zum IDM für Motif

Das Attribut *.real_modified* wird nicht unterstützt und hat immer den Wert *true*.

2.271 .real_screen

Dieses Attribut liefert die wirkliche und gültige Screen-Nummer zurück die immer ≥ 0 ist, auch wenn an der **display**-Resource eine ungültige definiert wurde.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_real_screen	Bezeichner: AT-real-screen
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	display

Hinweis

Nur der IDM für Motif besitzt Multiscreen-Support.

Siehe auch

Attribut *.screen*

Kapitel „Multiscreen Support unter Motif“ im Handbuch „Programmiertechniken“.

2.272 .real_sensitive

Über dieses Attribut kann erfragt werden, ob das Objekt wirklich vom Benutzer selektiert werden kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>boolean</i>	get

C
Bezeichner: AT_real_sensitive
Datentyp: DT_boolean

COBOL
Bezeichner: AT-real-sensitive
Datentyp: DT-boolean

Klassifizierung
Standardattribut

Siehe auch

Attribut *.sensitive*

2.273 .real_shadowobject

Mit Hilfe u.a. dieses Attributes können Sie auf benutzerdefinierte Attribute zugreifen. Im Speziellen wird damit das aktuell referenzierte Objekt abgefragt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>object</i>	get

C
Bezeichner: AT_real_shadowobject
Datentyp: DT_object

COBOL
Bezeichner: AT-real-shadowobject
Datentyp: DT-object

Klassifizierung
Allgemeines Attribut

Siehe auch

Attribute *.count*, *.shadowattr*, *.shadowindex*, *.shadowobject*, *.type*

2.274 .real_size[integer]

Mit diesem Attribut kann die tatsächliche Größe einzelner Splitbereiche abgefragt werden. Es wird immer ein Pixelwert zurückgeliefert. Der Index ist einsbasiert und die gültigen Werte dafür sind:

real_size[l] – gültig, falls $1 \leq l$ und $l \leq \text{childcount}$

Das Attribut kann wie alle anderen *real_...*-Attribute erst abgefragt werden, wenn das Objekt erzeugt und auf dem Bildschirm dargestellt worden ist. Wird *.real_size[l]* von einem unsichtbaren Kind abgefragt (dessen Sektion deswegen noch gar nicht erst erzeugt worden ist), wird 0 zurückgeliefert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_real_size	Bezeichner: AT-real-size
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	splitbox

Siehe auch

Attribut *.size[integer]*

2.275 .real_version[enum]

Das Attribut *.real_version* dient zur Abfrage, ob die gewünschte Version des XML-Toolkits geladen werden kann. Die Abfrage des Attributs erzwingt ein Laden des XML-Toolkits. Kann das XML-Toolkit nicht geladen werden, wird der Wert *-1* zurückgeliefert, ansonsten wird der Wert von *.version[enum]* zurückgeliefert.

Als Index dient die *toolkit*-Enumeration. Die Angabe eines Index ist optional. Wird kein Index angegeben, erfolgt der Zugriff implizit auf das aktuelle Toolkit.

Als Index dient die *toolkit*-Enumeration. Momentan wird aber nur der Wert *toolkit_windows* unterstützt, da nur unter Microsoft Windows die Laufzeitversion des MSXML-Controls vorgegeben werden kann. Hierzu muss die Majorversion mit *100* multipliziert und gegebenenfalls die Minorversion addiert werden. Wird *0* angegeben, wird das MSXML-Control versionsunabhängig geladen, was momentan die Version 3.0 ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	<i>get</i>

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_real_version	Bezeichner: AT-real-version
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	document

Beispiel

Es soll MSXML 5.0 verwendet werden, damit XSD als Dokumenttyp verwendet werden kann:

```
this.version[toolkit_windows] := 500;
```

Anschließend wird überprüft, ob MSXML 5.0 geladen werden kann:

```
if (this.real_version[toolkit_windows] = 500) then
  // OK
endif
```

Siehe auch

Attribut *.version[enum]*

2.276 .real_visible

Mit Hilfe dieses Attributes kann erfragt werden, ob ein Objekt auf dem Bildschirm wirklich sichtbar erzeugt wurde oder nicht.

Ob dieses Objekt für den Benutzer durch andere Objekte verdeckt wird ist dabei nicht berücksichtigt.

Definition

Datentyp
boolean

Zugriff
get

C
Bezeichner: AT_real_visible
Datentyp: DT_boolean

COBOL
Bezeichner: AT-real-visible
Datentyp: DT-boolean

Klassifizierung
Standardattribut

Siehe auch

Attribut *.visible*

2.277 .real_width

Mit Hilfe dieses Attributes wird erfragt, wie breit das Objekt tatsächlich ist (Angabe in Pixel). Dazu muss das Objekt jedoch sichtbar sein (*.real_visible = true*).

Diese Breite kann für Objekte ohne Breitendefinition erfragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_real_width	Bezeichner: AT-real-width
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Anmerkung

Wurde bei der Definition von Objekten keine Breite und/oder Höhe angegeben, so stehen die Attribute *.real_width* bzw. *.real_height* zur Verfügung, um die Breite und Höhe solcher Objekte zu erfragen.

Mit Hilfe dieser Attribute ist es z.B. möglich, die Breite eines Objektes zu erfragen, dessen Geometrie mit Hilfe einer Fixierung am linken und rechten Rand eines Vaterobjekts (*.xauto = 0*) definiert wurde.

Diese Angaben erfolgen immer in Pixel!

Siehe auch

Attribute *.real_height*, *.width*

2.278 .real_x

Das Attribut *.real_x* wurde verfügbar gemacht, um bei Objekten, die über virtuelle Koordinaten definiert werden, den realen Abstand von links (in Pixel) bestimmen zu können.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_real_x	Bezeichner: AT-real-x
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Hinweis

Dieses Attribut ist nur für den Dialog Manager mit MOTIF verfügbar.

Siehe auch

Attribut *.real_y*

2.279 .real_xraster

Breite des intern verwendeten Rasters, das aufgrund des angegebenen Referenzfonts berechnet wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

C
Bezeichner: AT_real_xraster
Datentyp: DT_integer

COBOL
Bezeichner: AT-real-xraster
Datentyp: DT-integer

Klassifizierung
Objektspezifisches Attribut

Siehe auch

Attribut *.real_yraster*

2.280 .real_y

Das Attribut *.real_y* wurde verfügbar gemacht, um bei Objekten, die über virtuelle Koordinaten definiert werden, den realen Abstand von oben (in Pixel) bestimmen zu können.

Definition

Datentyp
integer

Zugriff
get

C
Bezeichner: AT_real_y
Datentyp: DT_integer

COBOL
Bezeichner: AT-real-y
Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Hinweis

Dieses Attribut ist nur für den Dialog Manager mit MOTIF verfügbar.

Siehe auch

Attribut *.real_x*

2.281 .real_yraster

Höhe des intern verwendeten Rasters, das aufgrund des angegebenen Referenzfonts berechnet wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

C
Bezeichner: AT_real_yraster
Datentyp: DT_integer

COBOL
Bezeichner: AT-real-yraster
Datentyp: DT-integer

Klassifizierung
Objektspezifisches Attribut

Siehe auch

Attribut *.real_xraster*

2.282 .record[integer]

Dieses Attribut liefert oder setzt den i-ten Record. Es beinhaltet – ähnlich wie *.child* – den Vektor der Records.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

C
Bezeichner: AT_record
Datentyp: DT_record

COBOL
Bezeichner: AT-record
Datentyp: DT-record

Klassifizierung
Hierarchieattribut

Siehe auch

Objekt record

2.283 .recordcount

Dieses Attribut erfragt die Anzahl der Records in einem Objekt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

C
Bezeichner: AT_recordcount
Datentyp: DT_integer

COBOL
Bezeichner: AT-recordcount
Datentyp: DT-integer

Klassifizierung
Hierarchieattribut

2.284 reexport

Mit *reexport* werden Objekte und benannte Regeln eines Moduls nach außen bekannt gemacht, sodass in einem importierenden Modul bzw. Dialog darauf zugegriffen werden kann. Es erfüllt damit die gleiche Funktion wie *export* mit dem Unterschied, dass sich bei *reexport* die Export-Eigenschaft der Kindobjekte eines Modells vererbt.

Dazu muss *reexport* beim Modell am Kind- und Vaterobjekt angegeben werden. Bei einem aus dem Modell abgeleiteten Objekt muss *reexport* dann nur noch am Vater angegeben werden. Am geerbten Kind des abgeleiteten Objekts ist – im Gegensatz zu *export* – kein *reexport* mehr notwendig.

Definition

```
reexport <Objektklasse> <Bezeichner>
{
  <weitere Objektdefinitionen wie Attribute, Kindobjekte...>
}
```

Wird *reexport* bei einem abgeleiteten Objekt am Vater angegeben, dann werden an diesem Objekt automatisch alle Kindobjekte exportiert, die bereits am Modell exportiert sind. Dabei spielt es keine Rolle, ob ihr Export am Modell durch die Schlüsselwörter *export* oder *reexport* zustande kommt oder bereits geerbt ist. Wenn bei einem abgeleiteten Objekt am Vater *export* angegeben wird, dann wird dadurch die Vererbung der Export-Eigenschaft für die Kindobjekte abgebrochen. Das ist auch dann der Fall, wenn ihnen am Modell *reexport* zugewiesen wurde bzw. wenn sie diese Eigenschaft geerbt haben. Ein Kindobjekt, dessen Vater nicht in irgendeiner Form exportiert ist, wird nicht (re)exportiert.

Wie *export* steht *reexport* am Anfang der Objektdefinition. Es kann überall verwendet werden, wo auch *export* verwendet werden kann. Das bedeutet, dass *reexport* nur in Modulen und nicht in Dialogen eingesetzt werden kann. An Kindobjekten, die ihre Export-Eigenschaft geerbt haben, werden die Attribute *reexport* und *export* ignoriert. Dadurch kann in vorhandenen Modulen *export* überall durch *reexport* ersetzt werden, um das veränderte Vererbungsverhalten zu nutzen.

Verfügbarkeit

Ab IDM-Version A.05.02.j

Beispiele

```
reexport model window MwiBase
{
  reexport child edittext Et {}
}
reexport model MwiBase MwiDerived
{
}
```

Auf das geerbte Kind *MwiDerived.Et* kann von außen zugegriffen werden, da es die Export-Eigenschaft von *MwiBase.Et* erbt.

Wird `export` anstelle von `reexport` verwendet, muss *MWiDerived.Et* explizit exportiert werden, damit man von außen darauf zugreifen kann.

```
export model window MWiBase
{
  export child edittext Et {}
}
export model MWiBase MWiDerived
{
  export .Et {} // geerbtes Kind
}
```

Dies gilt auch, wenn am abgeleiteten Objekt `export` verwendet und die Vererbung dadurch abgebrochen wird. Im folgenden Beispiel unterbindet `export` an *MWiOne* den Export von *MWiOne.Et*, obwohl am Modell *MWiRe* sowohl am Vater als auch am Kind `reexport` angegeben ist. Umgekehrt bewirkt bei einem abgeleiteten Objekt ein `reexport` des Vaters, dass am Modell exportierte Kinder beim abgeleiteten Objekt ebenfalls exportiert werden. Im Beispiel erbt *MWiTwo.Et* durch den `reexport` von *MWiTwo* seine Export-Eigenschaft, obwohl an *MWiEx* und *MWiEx.Et* nicht `reexport` sondern `export` angegeben ist.

```
reexport model window MWiRe
{
  reexport child edittext Et {}
}
export model window MWiEx
{
  export child edittext Et {}
}
export model MWiRe MWiOne
{
  // Kein Zugriff von außen auf das geerbte Kind MWiOne.Et,
  // da export an MWiOne die Vererbung abbricht.
  // Um MWiOne.Et von außen zugänglich zu machen,
  // ist ein explizites export .ET {} notwendig.
}
reexport model MWiEx MWiTwo
{
  // Zugriff von außen auf das geerbte Kind MWiTwo.Et ist möglich,
  // da MWiEx.Et export und MWiTwo reexport aufweist.
  // Dadurch erbt MWiTwo.Et die Export-Eigenschaft von MWiEx.Et.
}
```

Das nächste Beispiel verdeutlicht, dass eine Vererbung des Exports nur bei geerbten, hierarchischen Kindern stattfinden kann. Wie bei `export` gibt es auch bei `reexport` keine Vererbung beim obersten instanziierten Objekt in einer Vater-Kind-Hierarchie:

```
export default pushbutton PUSHBUTTON { }
reexport default image IMAGE { }
```

```

model pushbutton MPbOne {} // erbt nicht den export von PUSHBUTTON
model image MImOne {}      // erbt auch nicht den reexport von IMAGE
export model pushbutton MPbTwo {} // Expliziter export bzw. reexport sind
reexport model image MImTwo {}    // notwendig, wenn von außen auf MPbTwo
                                   // und MImTwo zugegriffen werden soll.

// Im Folgenden sind die Pushbuttons und Images keine
// geerbten hierarchischen Kinder der Groupbox- bzw. Layoutbox-Modelle.
// Deshalb erben sie nicht den export bzw. reexport von ihren Defaults,
// und müssen explizit exportiert werden, um sie außerhalb des Moduls
// bekannt zu machen.
export model groupbox MGb {
  pushbutton PbOne {}      // kein Zugriff von außen möglich
  export pushbutton PbTwo {} // export für Zugriff von außen notwendig
  image ImOne {}           // kein Zugriff von außen möglich
  reexport image ImTwo {}   // reexport für Zugriff von außen notwendig
}
reexport model layoutbox MLy {
  pushbutton PbOne {}      // kein Zugriff von außen möglich
  reexport pushbutton PbTwo {} // reexport für Zugriff von außen notwendig
  image ImOne {}           // kein Zugriff von außen möglich
  reexport image ImTwo {}   // reexport für Zugriff von außen notwendig
}

```

Anmerkung

- » Wir empfehlen durchgängig entweder *export* oder *reexport* zu nutzen, um schwer nachvollziehbare Zugriffsbeschränkungen zu vermeiden.
- » An Kindobjekten, die ihre Export-Eigenschaft geerbt haben, wird *reexport* ignoriert.

Siehe auch

Attribut *export*

Kapitel „Modularisierung“ im Handbuch „Programmiertechniken“

2.285 .reffont

Dieses Attribut gibt den Namen des Zeichensatzes an, auf den sich das X- und das Y-Raster beziehen sollen.

Auch das Objekt **dialog** hat das Attribut *.reffont*. Damit kann man die Positionierung von Fenstern steuern.

Definition

<i>Datentyp</i> <i>object [font]</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
---	----------------------------	-------------------------------

C
Bezeichner: AT_reffont
Datentyp: DT_font

COBOL
Bezeichner: AT-reffont
Datentyp: DT-font

Klassifizierung
Geometrieattribut

Anmerkungen

- » Die Attribute *.reffont*, *.xraster* und *.yraster* haben vor allem Bedeutung, wenn die Anwendung so realisiert werden soll, dass sie auf möglichst vielen unterschiedlichen Hardware-Umgebungen ablauffähig ist. Mit Hilfe dieser Attribute wird eine zugrundeliegende Objektgrößeneinheit definiert, die unabhängig von den bisher verwendeten Pixeleinheiten ist. Zur neuen Einheit wird die Größe eines Buchstabens des ausgewählten Referenzfonts oder die Angaben, die der Benutzer direkt ohne Angabe eines Zeichensatzes gemacht hat. Alle Dimensionen oder Positionen beziehen sich dann auf diese Einheit.

Bezeichner	Datentyp	Bedeutung
<i>.reffont</i>	<i>object (font)</i>	Name des Zeichensatzes, auf den sich diese Größe beziehen soll.
<i>.xraster</i>	<i>integer</i>	Zugrundeliegende Einheit in X-Richtung.
<i>.yraster</i>	<i>integer</i>	Zugrundeliegende Einheit in Y-Richtung.

- » Wird ein Referenzfont angegeben, so rechnet der IDM die Werte für *.xraster* und *.yraster* automatisch aus, d.h. die Angabe von *.reffont* überschreibt die Angaben von *.xraster* und *.yraster*.

Beispiel

```
.xraster    8;  
.yraster    16;  
...  
.posraster  true;
```

```
.xleft    10;  
.ytop     4;
```

Dieses so definierte Objekt hat in Pixelkoordinaten die Position 80, 64, da die Positionsangabe zur Basis 8 bzw. 16 erfolgte.

Siehe auch

Attribute *.xraster*, *.yraster*

2.286 .refstring

Das Attribut bezeichnet den an einer **font**-Ressource angegebenen Referenzstring. Dieser Referenzstring wird für die Berechnung der Rasterbreite verwendet.

Definition

<i>Datentyp</i> <i>string</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
----------------------------------	----------------------------	---------------------------------

<i>C</i> Bezeichner: AT_refstring Datentyp: DT_string	<i>COBOL</i> Bezeichner: AT-refstring Datentyp: DT-string
---	---

2-

<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> font
---	------------------------

.-

287 .rgb[enum]

Mit diesem Attribut werden die Intensitäten (0 ... 255) der Rot-, Grün- und Blau-Anteile einer RGB-Farbe bei einer **color**-Ressource definiert.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
-----------------------------------	----------------------------	---------------------------------

<i>C</i> Bezeichner: AT_rgb Datentyp: DT_integer	<i>COBOL</i> Bezeichner: AT-rgb Datentyp: DT-integer
--	--

Wertebereich
0 ... 255

<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> color
---	-------------------------

Indexbereich

color_red
Bestimmt die Intensität des Rot-Anteils.

color_green
Bestimmt die Intensität des Grün-Anteils.

color_blue
Bestimmt die Intensität des Blau-Anteils.

Hinweis

Die Attribute *.rgb[enum]*, *.hls[enum]* und *.name* schließen sich gegenseitig aus, sodass „get“ zu einer Fehlermeldung „*can't get value*“ führen kann.

2.288 .root

Dient zur Abfrage der Position, an der in einem XML-Baum oder in einer IDM-Objekthierarchie die Transformation begonnen wurde. Den Wert des Attributes abzufragen, ist deswegen auch nur während einer Transformation (also während des Aufrufs der **:apply()**-Methode) sinnvoll. Ansonsten sollte der Wert *null* sein.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string, object</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_root	Bezeichner: AT-root
Datentyp: DT_string, DT_object	Datentyp: DT-string, DT-object

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	transformer

In der Standard-Implementierung der **:apply()**-Methode des **transformers** wird dieses Attribut wie folgt gesetzt.

- » Wurde im Parameter *Src* ein **document**-Objekt übergeben, so wird *.root* ein leerer String "" zugewiesen, der die Wurzel des Dokumentes repräsentiert.
- » Wurde im Parameter *Src* ein **doccursor**-Objekt übergeben, so wird *.root* ein String zugewiesen, das die Position im XML-Baum bestimmt, auf die der **doccursor** verweist. Also: *.root := Src.path*.
- » Wurde im Parameter *Src* ein anderes IDM-Objekt übergeben, so wird *.root* diesem Objekt selbst zugewiesen.

Beim Verlassen der **:apply()**-Methode wird *.root* auf *null* gesetzt.

2.289 .rowalignment[integer]

Mit Hilfe dieses Attributes kann die Ausrichtung des Inhaltes bezüglich seines Feldes (oben, zentriert, unten) beschrieben werden. *.rowalignment[0]* ist dabei der Defaultwert, der für alle anderen genommen wird, wenn nichts weiter angegeben ist.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
-----------------------------------	----------------------------	-------------------------------

C
Bezeichner: AT_rowalignment
Datentyp: DT_integer

COBOL
Bezeichner: AT-rowalignment
Datentyp: DT-integer

<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield
---	------------------------------

Wertebereich

-1	unten bündig
0	vertikal zentriert
1	oben bündig

2.290 .rowcount

Das Attribut definiert die Anzahl der Zeilen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_rowcount	Bezeichner: AT-rowcount
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listview, tablefield

tablefield

Mit Hilfe dieses Attributes kann die Anzahl der insgesamt im Tablefield vorhandenen Zeilen definiert werden. Der Wertebereich reicht von 0 bis 65535.

Diese Anzahl muss nicht unbedingt wirklich im Tablefield vorhanden sein, sondern soll dem Tablefield vielmehr dazu dienen, die vertikale Scrollbar richtig zu berechnen.

listview

Dieses Attribut definiert die Anzahl der Listeneinträge.

Der Wertebereich von *.rowcount* ist $0 \dots n$ ($n > 0$).

Der Wert von *.rowcount* kann implizit erhöht werden, indem *.content* um jeweils eine neue Zeile erweitert wird.

Bei ungültigen Werten wird der Standardwert 0 verwendet. Der Attributwert wird allerdings nicht geändert.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

Siehe auch

Attribut *.colcount*

2.291 .rowfirst

Mit Hilfe dieses Attributes kann die Nummer der obersten, unterhalb der Überschrift sichtbaren Zeile definiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_rowfirst	Bezeichner: AT-rowfirst
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.292 .rowheader

Mit Hilfe dieses Attributes können Sie die Anzahl der Textzeilen (einzeln adressierbare Felder, Elemente) definieren, die als Überschrift dienen und damit nicht vertikal gescrollt werden sollen. Wertebereich 0-255.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_rowheader Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-rowheader Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield	

2.293 .rowheadfgc

Mit Hilfe dieses Attributes kann die Farbe definiert werden, mit der die Überschriften der Zeilen geschrieben werden sollen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

C
Bezeichner: AT_rowheadfgc
Datentyp: DT_color

COBOL
Bezeichner: AT-rowheadfgc
Datentyp: DT-color

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.294 .rowheadfont

Mit Hilfe dieses Attributes kann der Zeichensatz definiert werden, in dem die Überschriften (Header) der Zeilen geschrieben werden sollen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [font]	get, set	ja

C
Bezeichner: AT_rowheadfont
Datentyp: DT_font

COBOL
Bezeichner: AT-rowheadfont
Datentyp: DT-font

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.295 .rowheadshadow

Mit Hilfe dieses Attributes kann definiert werden, in welcher Form die Überschriften gestaltet werden. Dabei bedeutet *true*, dass die Überschriften mit Schatten (= Buttons) dargestellt werden sollen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_rowheadshadow	Bezeichner: AT-rowheadshadow
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Anmerkung für DM mit Motif

Das Attribut *.rowheadshadow* wird nicht unterstützt.

2.296 .rowheadvisible

Mit Hilfe dieses Attributes kann angegeben werden, ob die Überschriften (Header) angezeigt werden sollen oder nicht. Damit können Überschriften dynamisch eingeschaltet und wieder ausgeschaltet werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C
Bezeichner: AT_rowheadvisible
Datentyp: DT_boolean

COBOL
Bezeichner: AT-rowheadvisible
Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.297 .rowheight[integer]

Mit Hilfe dieses Attributes können Sie die Höhe einer Zeile (in Pixel, falls *.sizeraster* nicht gesetzt ist, in Rastereinheiten, falls *.sizeraster* gesetzt ist) definieren.

.rowheight[0] ist dabei der Defaultwert für alle nachfolgenden Zeilen, falls dort kein Wert angegeben ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_rowheight	Bezeichner: AT-rowheight
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.298 .rowlinewidth[integer]

Mit Hilfe dieses Attributes kann die Breite der vertikalen Linien, die im Tablefield gezeichnet werden, definiert werden. *.rowlinewidth[0]* ist dabei der Defaultwert, der für alle anderen genommen wird, wenn nichts weiter angegeben ist.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_rowlinewidth Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-rowlinewidth Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> tablefield	

2.299 .rowsizeable[integer]

Dieses einfach indizierte Attribut steuert die interaktive Vergrößerbarkeit von Zeilen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_rowsizeable	Bezeichner: AT-rowsizeable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Wird die Maus über den Rand eines manipulierbaren Feldes bewegt, so ändert sich der Mauszeiger und zeigt ein Größenänderungssymbol. Das Aussehen des Cursors ist vom Anwendungsprogrammierer nicht spezifizierbar.

Der Benutzer kann mit der linken Maustaste die Größenänderung einleiten. Es erscheint eine graue schraffierte Linie, die die Position der Maus anzeigt (Die Breite der schraffierten Linie ist abhängig von dem Attribut *.rowlinewidth[integer]*). Wird die Maus (mit gedrückter linker Maustaste) bewegt, wandert die schraffierte Linie mit. Die Größenänderung wird abgeschlossen, indem die Maustaste losgelassen wird.

Dieses Attribut wird wie andere einfach indizierte Attribute beim Tablefield gehandhabt, d.h. in *.rowsizeable[0]* steht der Default, der genommen werden soll, wenn für die Spalte kein Wert definiert ist. Der Default von *.rowsizeable[0]* ist *false*.

2.300 .rowvisible[integer]

Dieses einfach indizierte Attribut steuert die Sichtbarkeit von Zeilen, so dass sie je nach Anforderung ausgeblendet werden können.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_rowvisible	Bezeichner: AT-rowvisible
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

In manchen Anwendungen werden Informationen im **Tablefield** gehalten, die ausschließlich für die Anwendung und nicht für den Benutzer bestimmt sind. Diese Spalten bzw. Zeilen sollen dabei immer nur in speziellen Situationen angezeigt werden.

Diese Attribute werden wie andere einfach indizierte Attribute beim Tablefield gehandhabt, d.h. in `.rowvisible[0]` steht der Default, der genommen werden soll, wenn für die Zeile kein Wert definiert ist. Der Default ist `.rowvisible[0]` auf `true`.

2.301 .scale

Bedeutung an der *tile*-Ressource

Achtung: Dieses Attribut ist für die *tile*-Ressource **veraltet** und nur noch aus Kompatibilitätsgründen vorhanden. Es sagt aus, ob es an dem Bild/ Musters eine Skalierung gibt oder nicht. Welche Skalierung genau aktiv ist kann dem Attribut *.scalestyle* entnommen werden. Das Attribut sollte nicht mehr verwendet werden. Die Skalierung einer *tile*-Ressource sollte nur noch über das Attribut *.scalestyle* gesteuert werden.

Bedeutung am *setup*-Objekt

Am *setup*-Objekt liefert dieses Attribut die eingestellte Systemvergrößerung (die vom primären Monitor). Ein Setzen von *.scale* am *setup*-Objekt ist **nicht** möglich.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean (tile)</i>	get, set (<i>tile</i>)	nein
<i>integer (setup)</i>	get (<i>setup</i>)	
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_scale		Bezeichner: AT-scale
Datentyp: DT_boolean, DT_integer		Datentyp: DT-boolean, DT-integer
<i>Standardwert</i>	<i>Vererbung</i>	
<i>false (tile)</i>	nein	
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	tile, setup	

Wertebereich bei Tile

false

Das Muster behält seine definierte Größe bei. Es wird keine Skalierung angewandt.

true

Sagt kompatibel nur aus, dass es eine Skalierung gibt. Welche Skalierung angewandt wird kann dem Attribut *.scalestyle* entnommen werden. Bei einer Setzung wird das Attribut *.scalestyle* auf den Wert *scalestyle_any* gesetzt.

Siehe auch Attribut *.scalestyle*

Wertebereich bei Setup

0

Windows: DPI-Awareness ist ausgestellt.

Motif: Die Skalierung ist ausgeschaltet. Es werden keine XFT-Fonts genutzt und Bilder/Muster werden nicht skaliert.

Qt: Es wird die vom System verwendete Vergrößerung genutzt.

> 0

Aktuell aktive Vergrößerung der Anwendung in %.

Hinweis: eine Skalierung sollte in sinnvollen Schritten von 50%, 100%, 200%, 300% usw. erfolgen. Andernfalls kann es zu unschönen Skalierungs- und ggf. auch Performanceeffekten kommen.

Verfügbarkeit

Ab IDM-Version A.06.03.a.

Dies betrifft die Verfügbarkeit am **setup**-Objekt sowie die Einstufung als veraltet für die **tile**-Resource.

Siehe auch

Kapitel „HighDPI UnterstützungSupport“ im Handbuch „Programmiertechniken“

2.302 .scalestyle

Dieses Attribut steuert die Darstellung und Skalierung eines Bildes oder Musters.

Eine Skalierung hat Auswirkungen wenn die **tile**-Ressource in den Attributen *.picture*, *.picture[enum]*, *.picture[integer]* und/ oder *.picture_hilite[integer]* verwendet wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_scalestyle	Bezeichner: AT-scalestyle
Datentyp: DT_enum	Datentyp: DT-enum

<i>Standardwert</i>	<i>Vererbung</i>
<i>scalestyle_auto</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tile

Wertebereich

scalestyle_none

Das Muster bzw. Bild wird nicht skaliert. *.tiledpi* hat keine Auswirkungen.

scalestyle_any

Höhe und Breite des Musters bzw. Bildes werden voll auf die verfügbare Fläche vergrößert.

scalestyle_prop

Höhe und Breite des Musters bzw. Bildes werden auf die verfügbare Fläche vergrößert, wobei Höhen- und Breitenproportionen des Musters bzw. Bildes auf jeden Fall erhalten bleiben. D.h. es können oben und unten bzw. links und rechts Freiflächen entstehen.

scalestyle_num

Das Muster bzw. Bild wird entsprechend eines angegeben Faktors skaliert.

scalestyle_auto

Das Muster bzw. Bild wird entsprechend der eingestellten Bildschirmskalierung skaliert. Eine zur Vorgängerversion kompatible Skalierung findet für das Fenster-, Treeview- und Notepage-Icon statt.

scalestyle_dpi

Das Muster bzw. Bild wird immer entsprechend der eingestellten Bildschirmskalierung skaliert.

Siehe auch den Abschnitt „SkalierungScaling“ im Handbuch zur tile Ressource .

AnmerkungMOTIF & QT

Die Objekte menuitem und notepage unterstützen das Attribut nicht oder nur eingeschränkt.

Anmerkung Gruppierungsobjekte:

Gruppierungsobjekte, die über das Attribut *.tile* ein Bild oder Muster gesetzt haben, unterstützen das Attribut *.scalestyle* nicht.

Verfügbarkeit

Ab IDM-Version A.06.03.a

Siehe auch

Kapitel „HighDPI UnterstützungSupport“ im Handbuch „Programmiertechniken“

2.303 .scope

Mit Hilfe dieses Attributes können Sie erfragen, ob es sich bei dem Objekt um einen Default, ein Modell oder eine Instanz handelt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

C
Bezeichner: AT_scope
Datentyp: DT_scope

COBOL
Bezeichner: AT-scope
Datentyp: DT-scope

Klassifizierung
Allgemeines Attribut

Wertebereich

- | | |
|---|---------|
| 1 | |
| | Default |
| 2 | |
| | Modell |
| 3 | |
| | Instanz |

2.304 .scope[attribute]

Mit diesem Attribut kann der Gültigkeitsbereich von benutzerdefinierten Attributen erfragt werden.

Für den Zugriff auf den Gültigkeitsbereich von benutzerdefinierten Attributen ist als Index der Attributname über den Datentyp *attribute* zu verwenden.

Dieses Attribut ist für alle Objekte verfügbar, welche benutzerdefinierten Attribute enthalten dürfen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>anyvalue</i>	get

C
Bezeichner: AT_scope
Datentyp: DT_anyvalue

COBOL
Bezeichner: AT-scope
Datentyp: DT-anyvalue

Klassifizierung
Allgemeines Attribut

Siehe auch

Attribute *.indexscope[attribute]*, *.typescope*, *.typescope[integer]*

Kapitel „Gültigkeitsbereich zur besseren Typ-Prüfung“ im Handbuch „Regelsprache“

2.305 .screen

An einer **display**-Ressource dient dieses Attribut zum dynamischen Erfragen und Setzen der Screen-Nummern. Ein Umsetzen hat zur Folge, dass alle Fenster, die die betroffene **display**-Ressource verwenden auf den angegebenen Screen wandern.

Ein Zugriff auf das skalare Attribut des **setup**-Objekts liefert die Screen-Nummer des Default-Screens.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get (setup) get, set (display)	<i>changed-Ereignis</i> nein
<i>C</i> Bezeichner: AT_screen Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-screen Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> display, setup	

display

Der Wertebereich liegt zwischen -32767 und +32767, wobei eigentlich nur Zahlen ≥ 0 gültige Screen-Nummern sind (siehe dazu auch die Anmerkungen). Ungültige Screen-Nummern werden wie der Default-Screen behandelt.

Der Standardwert des Attributs ist -1.

Anmerkungen

Nur der IDM FÜR MOTIF besitzt Multiscreen-Support.

Gültige Screen-Nummern können entweder durch das Programm **xdpyinfo** ermittelt werden oder sind über das **.screen[integer]**-Attribut am **setup**-Objekt erfragbar. Die gelieferten Screen-Nummern entsprechen denen von **xdpyinfo** gelieferten Nummern und sind normalerweise von 0 ... (**.screen-count**-1).

Ein Beispieldialog ist bei der display-Ressource zu finden.

Siehe auch

Attribut **.real_screen**.

Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“.

2.306 .screen[integer]

Mit Hilfe dieses indizierten setup-Objekt-Attributes kann die Screen-Nummern aller vorhandenen Screens erfragt werden, wobei sich der Index zwischen 1 ... *setup.screencount* bewegen muss.

Dieses Attribut ist nur bei Multiscreendialogen (nur MOTIF) verfügbar.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_screen	Bezeichner: AT-screen
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkungen

Nur der IDM FÜR MOTIF besitzt Multiscreen-Support.

Die gelieferten Screen-Nummern entsprechen denen von **xdpyinfo** gelieferten Nummern und sind normalerweise von 0 ... (*screencount*-1) . Die über *.screen[integer]* erfragbaren Nummern sind nicht zwangsläufig aufsteigend sortiert.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

2.307 .screen_height

Mit Hilfe dieses setup-Objekt-Attributes kann die Bildschirmhöhe (in Pixel) abgefragt werden.

Es wird bei Multiscreendialogen (nur MOTIF) der Wert für den Default-Screen zurückgegeben.

Bei Multi-Monitordialogen wird (nur WINDOWS) wird der Wert für den Default-Monitor zurückgegeben.

Definition

Datentyp

integer

Zugriff

get

C

Bezeichner: AT_screen_height

Datentyp: DT_integer

COBOL

Bezeichner: AT-screen-height

Datentyp: DT-integer

Klassifizierung

Objektspezifisches Attribut

Objekte

setup

2.308 .screen_height[integer]

Mit Hilfe dieses setup-Objekt-Attributes kann die Bildschirmhöhe (in Pixel) abgefragt werden.

Das indizierte Attribut ist nur bei Multiscreendialogen (nur MOTIF) oder in einer Multi-Monitor Umgebung (nur WINDOWS) verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Der Wert ist in einer Multi-Monitor Umgebung unter WINDOWS dynamisch, da man jederzeit Monitore hinzufügen oder wegnehmen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_screen_height	Bezeichner: AT-screen-height
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung. Es ist zu beachten, dass mit Screen-Index nicht die Screen-Nummer gemeint ist, die z.B. mit dem Programm **xdpyinfo** erhalten wird.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

2.309 .screen_width

Mit Hilfe dieses setup-Objekt-Attributes kann die Bildschirmbreite (in Pixel) abgefragt werden.

Es wird bei Multiscreendialogen (nur MOTIF) der Wert für den Default-Screen zurückgegeben.

Bei Multi-Monitordialogen wird (nur WINDOWS) wird der Wert für den Default-Monitor zurückgegeben.

Definition

Datentyp

integer

Zugriff

get

C

Bezeichner: AT_screen_width

Datentyp: DT_integer

COBOL

Bezeichner: AT-screen-width

Datentyp: DT-integer

Klassifizierung

Objektspezifisches Attribut

Objekte

setup

2.310 .screen_width[integer]

Mit Hilfe dieses setup-Objekt-Attributes kann die Bildschirmbreite (in Pixel) abgefragt werden.

Das indizierte Attribut ist nur bei Multiscreendialogen (nur MOTIF) oder in einer Multi-Monitor Umgebung (nur WINDOWS) verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Der Wert ist in einer Multi-Monitor Umgebung unter WINDOWS dynamisch, da man jederzeit Monitore hinzufügen oder wegnehmen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_screen_width	Bezeichner: AT-screen-width
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung. Es ist zu beachten, dass mit Screen-Index nicht die Screen-Nummer gemeint ist, die z.B. mit dem Programm **xdpyinfo** erhalten wird.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

2.311 .screen_x

Mit Hilfe dieses setup-Objekt-Attributes kann die X-Koordinate des Ursprungs des primären Monitors bzw. Arbeitsbereichs (in Pixel) abgefragt werden.

Bei Multi-Monitordialogen wird (nur WINDOWS) wird der Wert für den Default-Monitor zurückgegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_screen_x	Bezeichner: AT-screen-x
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.312 .screen_x[integer]

Mit Hilfe dieses setup-Objekt-Attributes kann die X-Koordinate des Monitors (in Pixel) abgefragt werden.

Das indizierte Attribut ist nur bei einer Multi-Monitor Umgebung (nur WINDOWS) verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Der Wert ist in einer Multi-Monitor Umgebung unter WINDOWS dynamisch, da man jederzeit Monitore hinzufügen oder wegnehmen kann.

Definition

Datentyp

integer

Zugriff

get

C

Bezeichner: AT_screen_x

Datentyp: DT_integer

COBOL

Bezeichner: AT-screen-x

Datentyp: DT-integer

Klassifizierung

Objektspezifisches Attribut

Objekte

setup

Siehe auch

Kapitel „Multi-Monitor Support unter Windows“ im Handbuch „Programmiertechniken“.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.313 .screen_y

Mit Hilfe dieses setup-Objekt-Attributes kann die Y-Koordinate des Ursprungs des primären Monitors bzw. Arbeitsbereichs (in Pixel) abgefragt werden.

Bei Multi-Monitordialogen wird (nur WINDOWS) wird der Wert für den Default-Monitor zurückgegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_screen_y	Bezeichner: AT-screen-y
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.314 .screen_y[integer]

Mit Hilfe dieses setup-Objekt-Attributes kann die Y-Koordinate des Monitors (in Pixel) abgefragt werden.

Das indizierte Attribut ist nur in einer Multi-Monitor Umgebung (nur WINDOWS) verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Der Wert ist in einer Multi-Monitor Umgebung unter WINDOWS dynamisch, da man jederzeit Monitore hinzufügen oder wegnehmen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_screen_y	Bezeichner: AT-screen-y
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.315 .screencount

Dieses Attribut ist am **setup**-Objekt verfügbar. Es liefert unter MOTIF die Anzahl verfügbarer Screens (Bildschirme), die für das benutzte Display konfiguriert sind. Unter WINDOWS liefert es die Anzahl der Monitore und ist dynamisch, da man jederzeit Monitore hinzufügen oder wegnehmen kann.

Der Wert von *.screencount* ist immer größer als 0. Auf Systemen ohne Multi-Screen- oder Multi-Monitor Support ist er 1.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get
<i>C</i>	<i>COBOL</i>
Bezeichner: AT_screencount	Bezeichner: AT-screencount
Datentyp: DT_integer	Datentyp: DT-integer
<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

2.316 .searchpath

Mit diesem Attribut kann der aktuelle Suchpfad, der für Dialog-, Modul-, Interface- und Binärdateien bei Imports mit use verwendet wird, erfragt und umgesetzt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_searchpath	Bezeichner: AT-searchpath
Datentyp: DT_string	Datentyp: DT-string

Standardwert
"~;"

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Verfügbarkeit

Ab IDM-Version A.06.02.g

Siehe auch

Kapitel „Suchpfad für Interface-, Modul-, Dialog- und Binärdateien“ im Handbuch „Programmiertechniken“

2.317 .selected[integer]

Dieses Attribut legt für jeden Listeneintrag des **listview**-Objekts fest, ob er markiert ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_selected	Bezeichner: AT-selected
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Standardwert</i>	<i>Vererbung</i>
<i>false</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listview

Der Wertebereich des Index ist $0 \dots \text{.rowcount}$, wobei der Wert mit Index 0 als Standardwert für nicht gesetzte Werte im Bereich $1 \dots \text{.rowcount}$ dient.

Wertebereich

false

Der Listeneintrag ist **nicht** markiert.

true

Der Listeneintrag ist markiert.

Das Attribut wird durch Interaktion des Anwenders (*activate*- und *deactivate*-Ereignisse) geändert.

Hinweis

Die Darstellung markierter Einträge hängt vom System ab. Dies gilt insbesondere für die Sichtbarkeit der Markierung, wenn das **listview** nicht den Fokus besitzt.

2.318 .selection[enum]

Mit Hilfe dieses Attributes kann definiert werden, welche Art von Selektion im **tablefield** zulässig ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_selection	Bezeichner: AT-selection
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Indexbereich

- sel_column* (Spaltenselektion)
Ganze Spalten können ausgewählt werden.
- sel_header* (Headerselektion)
Spaltenüberschriften und Zeilenköpfe können ausgewählt werden.
- sel_row* (Zeilenselektion)
Ganze Zeilen können ausgewählt werden .
- sel_single* (Einzelfeldselektion; Standard)
Einzelne Felder im Inneren können ausgewählt werden.

Diese Selektionsarten können durch Setzen von *.selection[<selection_type>]* auf *true* einzeln aktiviert werden.

Siehe auch

Attribut *.sel/style*

Kapitel „Selektion im Tablefield“ in der „Objektreferenz“

2.319 .self

Mit *.self* wird ein Objekt selbst angesprochen. In den meisten Fällen kann dies einfacher, nur über den Identifikator des Objekts, erreicht werden. In bestimmten Situationen kann es allerdings notwendig sein, ein Attribut anzugeben. Dann kann das Attribut *.self* verwendet werden.

Beispielsweise wird auf den Wert einer globalen Variablen zugegriffen, wenn deren Identifikator in der Regelsprache verwendet wird. Ein Zugriff auf die Variable als Objekt, z.B. um ihren Datentyp abzufragen oder zu ändern, kann hier durch *.self* erreicht werden.

Definition

Datentyp
object

Zugriff
get

C
Bezeichner: AT_self
Datentyp: DT_object

COBOL
Bezeichner: AT-self
Datentyp: DT-object

Klassifizierung
Allgemeines Attribut

Beispiel

```
dialog Dialog
variable integer Answer := 42;
on dialog start
{
  print Answer;           // gibt "42" aus, also den Wert
  print Answer.self;      // gibt "variable Dialog.Answer" aus,
                          // also die Variable selbst
}
```

2.320 .selstyle

Dieses Attribut steuert, ob der Benutzer mehrere Einträge in einem Objekt auswählen kann und auf welche Art er die Auswahl vornehmen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
enum	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_selstyle	Bezeichner: AT_selstyle
Datentyp: DT_enum	Datentyp: DT_enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, tablefield

Wertebereich

single

Es kann immer nur ein Eintrag ausgewählt werden und aktiv sein.

multiple

Es können mehrere Einträge ausgewählt werden und aktiv sein, wobei Einträge einzeln per Mausklick aus- und abgewählt werden können.

Kompatibel zu *.multisel = true*.

extended

Es können mehrere Einträge ausgewählt werden und aktiv sein, wobei eine bestehende Auswahl durch Mausklicks in Kombination mit den Tasten **Strg** und **Shift** erweitert und verringert werden kann. Unter Verwendung der **Shift**-Taste können mehrere Einträge gleichzeitig aus- und abgewählt werden.

Im Einzelnen sieht das Verhalten wie folgt aus:

Benutzeraktion	Selektion
„Klick“	bisherige Selektion löschen, aktuelles Element aktivieren
Shift + „Klick“	bisherige Selektion erhalten, Bereich vom letzten „Klick“ ohne Shift bis zum aktuellen Element aktivieren
Strg + „Klick“	bisherige Selektion erhalten, Aktivierung des aktuellen Elements umschalten
Strg + Shift + „Klick“	bisherige Selektion erhalten, Bereich vom letzten „Klick“ ohne Shift bis zum aktuellen Element erhält den Aktivierungszustand des ersten

Das Zusammenspiel von *.selstyle* mit dem Attribut *.selection[enum]*, mit dem im **tablefield** eine Selektionsart ausgewählt wird, wird in der nachfolgenden Tabelle dargestellt:

Selektionsart	Aktuelles Element
<i>sel_single</i>	ein Feld
<i>sel_header</i>	ein Feld eines Headers
<i>sel_row</i>	eine Zeile
<i>sel_column</i>	eine Spalte

Für *.selstyle single* und *extended* gilt: ein einspaltiges **tablefield** verhält sich wie eine **listbox**.

Anmerkung

Durch das Attribut *.selstyle* für **tablefield** und **listbox** ist das Attribut *.multisel* überflüssig, *.multisel* wird jedoch weiterhin unterstützt.

Siehe auch

Attribut *.selection[enum]*

Kapitel „Selektion im Tablefield“ in der „Objektreferenz“

2.321 .sensitive

Mit Hilfe dieses Attributes wird definiert, ob das Objekt vom Benutzer selektiert werden kann oder nicht.

.*sensitive* beim **tablefield** beschreibt, ob das Objekt selektierbar ist. Damit kann das gesamte Tablefield auf einmal insensitiv gemacht werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C

Bezeichner: AT_sensitive

Datentyp: DT_boolean

COBOL

Bezeichner: AT-sensitive

Datentyp: DT-boolean

Klassifizierung

Standardattribut

Besonderheiten der Attribute .sensitive und .visible

Im Gegensatz zu den anderen Objektattributen erhalten die Attribute .*visible* und .*sensitive* erst im Zusammenhang mit der Objekthierarchie ihre endgültige Ausprägung, denn die entsprechenden Attribute des zugehörigen Vaterobjektes haben auf diese entscheidenden Einfluss. Konkret heißt das, dass ein Objekt nur dann sichtbar sein kann, wenn es selbst sichtbar **und** alle seine Väter sichtbar sind. Dasselbe gilt für die Selektierbarkeit.

Für die Objekte **tablefield** und **edittext** sind Attribute verfügbar, die dem Dialogdesigner die Möglichkeit zu geben, das Aussehen und Eingabeverhalten beeinflussen zu können. Dies betrifft die Darstellung von nicht selektierbaren Edittexten sowie die Definition, ob der Benutzer mit diesen Objekten interagieren können soll oder nicht:

» Bedienbar- oder Selektierbarkeit

Attribut .*sensitive*

Die Eigenschaft, ob ein Objekt vom Benutzer selektiert werden kann oder nicht. Nur wenn ein Objekt selektierbar ist, kann der Benutzer die für das Objekt üblichen Aktionen auf das Objekt anwenden, z.B. Daten eingeben und Scrollen.

» Editierbarkeit

Attribut .*editable*

Die Eigenschaft, dass der Benutzer den Inhalt des Objektes ändern kann; er ist also in der Lage, den Inhalt zu editieren und zu verändern.

» Fokussierbarkeit

Attribut .*navigable*

Die Eigenschaft, dass das Objekt in der Tastatursteuerung erfasst wird und damit den Eingabefokus erhalten kann.

Diese drei Attribute beeinflussen nun das Verhalten des Objektes in folgender Weise:

Attribut	Wert	Auswirkung
<code>.sensitive</code>	<code>true</code>	Objekt kann vom Benutzer selektiert werden. Nur wenn das Attribut <code>.sensitive</code> den Wert <code>true</code> hat, können die Attribute <code>.navigable</code> und <code>.editable</code> wirksam werden, ansonsten werden sie ignoriert.
<code>.sensitive</code>	<code>false</code>	Objekt kann nicht vom Benutzer selektiert werden. Es sind also keinerlei Aktionen auf dem Objekt möglich. Der Inhalt des Objekts wird wie für die Fenstersysteme üblich gegraut dargestellt.
<code>.editable</code>	<code>true</code>	Benutzer kann den Inhalt des Objektes ändern, falls er das Objekt selektieren kann.
<code>.editable</code>	<code>false</code>	Benutzer kann den Inhalt des Objektes nicht verändern.
<code>.navigable</code>	<code>true</code>	Objekt ist in der normalen Tastatursteuerung enthalten, d.h., der Benutzer kann das Objekt durch Navigation im zugehörigen Fenster erreichen.
<code>.navigable</code>	<code>false</code>	Das Objekt kann nicht über die Tastatursteuerung erreicht werden. Das Objekt kann jedoch trotzdem über die Maus den Fokus erhalten.

Anmerkung zum IDM für Motif

Motif 1.1

Wenn bei der **listbox** `.sensitive = false` ist, bleiben die Scrollbars sensitiv, d.h. sie sind bedienbar.

Motif 1.2

Wenn bei der **listbox** `.sensitive = false` ist, sind auch die Scrollbars insensitiv, d.h. sie sind nicht bedienbar.

Siehe auch

Attribute `.editable`, `.navigable`, `.visible`

2.322 .sensitive[integer]

Mit Hilfe dieses Attributes kann angegeben werden, ob einzelne Einträge eines unterstützten Listenobjekts selektierbar sind oder nicht.

In der Motif-Version wird *.sensitive[integer]* ignoriert.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_sensitive	Bezeichner: AT-sensitive
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox

2.323 .sensitive[index]

Mit Hilfe dieses Attributes kann angegeben werden, ob einzelne Felder des Tablefields [row, column] selektierbar sind oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_sensitive	Bezeichner: AT-sensitive
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.324 .shadowattr

Mit Hilfe u.a. dieses Attributes können Sie auf benutzerdefinierte Attribute zugreifen und diese ändern. Im Speziellen wird damit das Attribut des referenzierten Attributes von shadow-Attributen angegeben.

Definition

<i>Datentyp</i> <i>attribute</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
-------------------------------------	----------------------------	-------------------------------

C
Bezeichner: AT_shadowattr
Datentyp: DT_attribute

COBOL
Bezeichner: AT-shadowattr
Datentyp: DT-attribute

Klassifizierung
Allgemeines Attribut

Siehe auch

Attribute *.count*, *.real_shadowobject*, *.shadowindex*, *.shadowobject*, *.type*

2.325 .shadowindex

Mit Hilfe u.a. dieses Attributes können Sie auf benutzerdefinierte Attribute zugreifen und diese ändern. Im Speziellen wird damit der Index des referenzierten Attributes von shadow-Attributen angegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_shadowindex	Bezeichner: AT-shadowindex
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Allgemeines Attribut

Siehe auch

Attribute *.count*, *.real_shadowobject*, *.shadowattr*, *.shadowobject*, *.type*

2.326 .shadowinstance

Dieses Attribut definiert, dass beim Instanzieren eines Modells der im benutzerdefinierten Attribut angegebene Verweis auf die neu generierte Instanz umgehängt werden soll.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C
Bezeichner: AT_shadowinstance
Datentyp: DT_boolean

COBOL
Bezeichner: AT-shadowinstance
Datentyp: DT-boolean

Klassifizierung
Allgemeines Attribut

Beispiel

```
model record MRec
{
  integer I 123;
}
model window MWi
{
  string S shadows instance Et.content;
  integer I shadows instance MRec.I;
  child edittext Et
  {
    .content "Hello";
  }
}
MRec Rec
{
  .I := 456;
}

MWi Wi
{
  Et.content := "World";
}
on dialog start
{
  !! Wi.S shadows Wi.Et.content
  !! and not MWi.Et.content!
  print Wi.content;    // result: "World"
  !! shadow and shadowed object are not in the same model
  print Wi.I;          // result: 123
}
```

}

2.327 .shadowobject

Mit Hilfe u.a. dieses Attributes können Sie auf benutzerdefinierte Attribute zugreifen und diese ändern. Im Speziellen wird damit das referenzierte Objekt von shadow-Attributen angegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

C
Bezeichner: AT_shadowobject
Datentyp: DT_object

COBOL
Bezeichner: AT-shadowobject
Datentyp: DT-object

Klassifizierung
Allgemeines Attribut

Siehe auch

Attribute *.count*, *.real_shadowobject*, *.shadowattr*, *.shadowindex*, *.type*

2.328 .shortdaynames

Dieses Attribut bestimmt, ob im aufklappbaren Kalender kurze Wochentagsnamen angezeigt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_shortdaynames	Bezeichner: AT-shortdaynames
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Standardwert</i>	<i>Vererbung</i>
<i>false</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	datetime

Wertebereich

false

Im Kalender werden normale Wochentagsnamen angezeigt.

true

Im Kalender werden kurze Wochentagsnamen angezeigt.

Hinweis

Im Deutschen besteht kein Unterschied zwischen kurzen und normalen Wochentagsnamen.

2.329 .showitem

Mit diesem Attribut wird definiert, wie viele Einträge in der Liste eines **poptextes** angezeigt werden sollen. Enthält die Liste mehr Einträge als hier angegeben, wird eine vertikale Scrollbar in der geöffneten Liste angezeigt.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_showitem Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-showitem Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> poptext	

Hinweise zum IDM FÜR QT

Unter QT werden standardmäßig in einem aufgeklappten **poptext** maximal 10 Einträge angezeigt. Enthält die **poptext**-Liste mehr als 10 Einträge, dann wird zusätzlich eine Scrollbar in der Liste angezeigt.

Das Attribut *.showitem* wird bei den UI-Stilen „mac“ und „gtk+“ ignoriert. Im UI-Stil „cleanlooks“ wird *.showitem* bei *.style = poptext* ignoriert, hier wird immer die komplette Liste angezeigt.

Bei *.style = listbox* wird *.showitem* gänzlich ignoriert.

2.330 .size

Dieses Attribut der **font**-Ressource definiert die Schriftgröße (in Punkten).

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
<i>C</i> Bezeichner: AT_size Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-size Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> font	

2.331 .size[integer]

Die Ausdehnung des Splitbereichs einer Splitbox mit dem Index I (einsbasiert) wird mit diesem Attribut festgelegt (Pixel- oder Rasterwerte). Mit Index = 0 kann ein Defaultwert definiert werden, welcher an alle Elemente von .size[I] übertragen wird, für die kein expliziter Wert vorliegt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_size	Bezeichner: AT-size
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	splitbox

Siehe auch

Attribute .maxsize[integer], .minsize[integer]

2.332 .sizeable

Dieses Attribut steuert, ob die Größe des Objekts vom Benutzer interaktiv verändert werden kann.

Beim **Fenster** bestimmt dieses Attribut auch, ob es mit einem Mechanismus (Knopf, Box) ausgestattet sein soll, mit dem es vom Benutzer das Fenster vergrößert oder verkleinert werden kann.

Bei der **toolbar** definiert *.sizeable* (ohne Index) den Standardwert für die nicht mit *.sizeable[class]* gesetzten Werte.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_sizeable	Bezeichner: AT-sizeable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	toolbar, window

Besonderheit unter Motif

Das Attribut kann, abhängig vom verwendeten Display- bzw. Desktop Manager, nicht im sichtbaren Zustand umgesetzt werden; unter Umständen ist ein Setzen zur Laufzeit auch nicht möglich. In manchen Fällen kann es helfen, die Sichtbarkeit des Fensters umzuschalten.

Da unter MOTIF das Umsetzen dieses Attributs direkt vom verwendeten Display- bzw. Desktop Manager abhängig ist, wird empfohlen, das Attribut nur statisch bzw. direkt nach Erzeugen einer Instanz mittels *:create(..., true)* im unsichtbaren Zustand zu setzen.

2.333 .sizeable[class]

Das Attribut gibt an, ob ein Toolwindow bzw. eine eingedockte Toolbar interaktiv in der Größe geändert werden kann oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_sizeable	Bezeichner: AT-sizeable
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	toolbar

Wertebereich

true

Die Größe der **toolbar** kann mit der Maus (angedockt und abgedockt) oder über das Systemmenü (abgedockt) geändert werden.

false

Die Größe der **toolbar** kann nicht interaktiv geändert werden.

Indexbereich

toolbar

Möglichkeit zur interaktiven Größenänderung wenn die Toolbar angedockt ist

window

Möglichkeit zur interaktiven Größenänderung wenn die Toolbar abgedockt ist (Toolwindow)

Ohne einen Index gibt das Attribut den Standardwert für beide Andockzustände zurück oder legt ihn fest.

2.334 .sizeraster

Dieses Attribut bestimmt die Interpretation der Attributwerte für die Größe eines Objektes (*.width*, *.height*). Es kann genutzt werden, um die Größe eines Objektes in Abhängigkeit vom benutzten Raster oder genau in Pixeln anzugeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C

Bezeichner: AT_sizeraster

Datentyp: DT_boolean

COBOL

Bezeichner: AT-sizeraster

Datentyp: DT-boolean

Klassifizierung

Geometrieattribut

Wertebereich

true

Die Werte der Größenattribute werden als Rastereinheiten interpretiert. Das zugrundeliegende Raster ist das Raster des Vaterobjektes.

false

Die Werte der Größenattribute werden als Pixeleinheiten interpretiert. Das Raster des Vaterobjektes wird ignoriert.

Hinweis zum IDM für Motif (ab IDM A.05.02.e)

Die Größenänderung von Fenstern mit *.sizeraster = true* erfolgt in Schrittweiten des Rasters, wenn dies vom Fenstermanager unterstützt wird.

Siehe auch

Attribut *.posraster*

Kapitel „Geometrieattribute“

2.335 .smallpicheight

Mit diesem Attribut wird die Höhe eines kleinen Symbols festgelegt.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_smallpicheight	Bezeichner: AT-smallpicheight
Datentyp: DT_integer	Datentyp: DT-integer

<i>Standardwert</i>	<i>Vererbung</i>
0	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listview

0

Die Breite *.smallpicwidth* wird verwendet. Ist diese ebenfalls 0, dann wird die Größe des Symbols vom System bestimmt.

> 0

Höhe des kleinen Symbols.

Das kleine Symbol wird in der kleinen Symbolansicht (*.style = "smallicon"*, *.style = "smallpicture"*), in der Listenansicht (*.style = "list"*) und in der Detailansicht (*.style = "detail"*, *.style = "report"*) angezeigt.

Bei ungültigen Werten wird der Standardwert 0 verwendet. Der Attributwert wird allerdings nicht geändert.

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

Siehe auch

Attribute *.smallpicture[integer]*, *.smallpicwidth*

2.336 .smallpicture[integer]

Das Attribut definiert das kleine Symbol für jeden Listeneintrag.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [tile]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_smallpicture	Bezeichner: AT-smallpicture
Datentyp: DT_tile	Datentyp: DT-tile

<i>Standardwert</i>	<i>Vererbung</i>
null	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listview

Der Wertebereich des Index ist *0rowcount*, wobei der Wert mit Index *0* als Standardwert für nicht gesetzte Werte im Bereich *1rowcount* dient.

null

Das angezeigte Symbol ergibt sich wie folgt:

- » Wenn *.smallpicture[0]* $\neq$ *null* ist, wird das dort definierte Symbol verwendet.
- » Wenn *.smallpicture[0]* = *null* ist, dann wird das große Symbol verkleinert dargestellt (je nach Vorhandensein *.picture[1]* oder *.picture[0]*).
- » Wenn weder ein kleines noch ein großes Symbol vorhanden sind, dann wird der Eintrag ohne Symbol dargestellt.

tile

Ressource, die das kleine Symbol enthält.

Das kleine Symbol wird in der kleinen Symbolansicht (*.style = "smallicon"*, *.style = "smallpicture"*), in der Listenansicht (*.style = "list"*) und in der Detailansicht (*.style = "detail"*, *.style = "report"*) angezeigt.

Kleines und großes Symbol gehören zusammen und sollten dieselbe Information darstellen. Beide Symbole können nur gemeinsam referenziert werden. Deshalb sind unterschiedliche Kombinationen zu vermeiden, da jede Kombination Speicherplatz verbraucht.

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

Siehe auch

Attribute *.smallpicheight*, *.smallpicwidth*

2.337 .smallpicwidth

Mit diesem Attribut wird die Breite eines kleinen Symbols festgelegt.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_smallpicwidth	Bezeichner: AT-smallpicwidth
Datentyp: DT_integer	Datentyp: DT-integer

<i>Standardwert</i>	<i>Vererbung</i>
<i>0</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listview

0

Die Breite des Symbols wird vom System bestimmt.

> 0

Breite des kleinen Symbols.

Das kleine Symbol wird in der kleinen Symbolansicht (*.style = "smallicon"*, *.style = "smallpicture"*), in der Listenansicht (*.style = "list"*) und in der Detailansicht (*.style = "detail"*, *.style = "report"*) angezeigt.

Bei ungültigen Werten wird der Standardwert *0* verwendet. Der Attributwert wird allerdings nicht geändert.

Wenn *.smallpicheight = 0* ist, bestimmt *.smallpicwidth* auch die **Höhe** des kleinen Symbols.

Das Ändern des Attributs im sichtbaren Zustand kann zu einem Flackern des Objekts führen.

Siehe auch

Attribute *.smallpicheight*, *.smallpicture[integer]*

2.338 .source

Dieses Attribut bestimmt das Verhalten des Objekts als Quelle einer Drag & Drop oder Clipboard-Operation. Als Werte werden Ressourcen vom Typ **source** verwendet.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [source]	get, set	ja

C
Bezeichner: AT_source
Datentyp: DT_source

COBOL
Bezeichner: AT-source
Datentyp: DT-source

Klassifizierung
Standardattribut

2.339 .spacing

Definiert den Zwischenraum zwischen Bild und Text in einem **image**-Objekt. Der Zwischenraum wird **nicht** berücksichtigt, wenn das Objekt nur ein Bild oder nur einen Text enthält.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_spacing	Bezeichner: AT-spacing
Datentyp: DT_integer	Datentyp: DT-integer

<i>Wertebereich</i>	<i>Standardwert</i>
-32767 ... 32767	0 (MICROSOFT WINDOWS) 2 (MOTIF)

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	image

Die Wirkung von *.spacing* in Abhängigkeit von den Attributen *.alignment* und *.tilestyle* ist beim Attribut *.tilestyle* beschrieben.

2.340 .specified

Gibt an, ob ein Attribut eines DOM-Knotens explizit angegeben wurde oder von einem Standardwert ererbt wurde. Das Attribut ist immer *true*, außer wenn das Attribut *.nodetype* den Wert *nodetype_attribute* besitzt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>boolean</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_specified	Bezeichner: AT-specified
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	doccursor

Das Attribut ist für den XML-Cursor verfügbar. Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

Siehe auch

Attribut *.nodetype*

2.341 .startsel

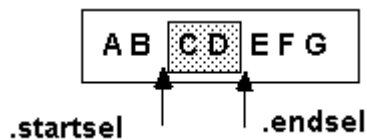
Bei **edittext** und **poptext** definiert das Attribut **.startsel** den Anfang der Markierung (selektierter Bereich) im Eingabefeld.

Bei den Datei- und Verzeichnisdialogen (**filereq**) lässt sich mit diesem Attribut ein initialer Wert für die Datei- und Verzeichnisdialoge angeben. Wenn **.startsel = true** ist, wird der Wert des Attributs **.value** als Vorgabe übernommen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja
<i>boolean (filereq)</i>		nein (filereq)
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_startsel		Bezeichner: AT-startsel
Datentyp: DT_integer		Datentyp: DT-integer
Datentyp: DT_boolean (filereq)		Datentyp: DT-boolean (filereq)
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	edittext, filereq, poptext	

2.341.1 edittext, poptext



Beim Verändern von **.startsel** bzw. **.endsel** muss **.focus = true** sein, damit die neue Selektion dargestellt werden kann. Dabei darf **.startsel** auch größer als **.endsel** sein.

Besondere Werte

Folgende Werte, die insbesondere bei gesetzten Formaten sinnvoll sind, haben eine Sonderstellung:

- 1
Markierung einschließlich aller Nachkommastellen (auch per Format eingefügter Nullen, die nicht im Inhaltsstring enthalten sind)
- 2
Markierung aller Vorkommastellen bis zum rechten Ende (Komma)
- 3
Markierung aller Vorkommastellen einschließlich per Format gesetzter führender Nullen

Eine Markierung des Vorzeichens ist in keinem Fall möglich.

Anmerkung zum IDM für Windows

Beim Abfragen von *.startsel* und *.endsel* des **edittextes** kann *.endsel* ab IDM-Version A.05.02.I auch kleiner sein als *.startsel*. Der Cursor befindet sich in diesem Fall links von der Markierung. Wenn allerdings der Anwender Text mit der Maus markiert, ist *.startsel* immer kleiner als *.endsel* und es kann nicht erkannt werden, wo sich der Cursor befindet. In vorherigen IDM-Versionen war *.startsel* immer kleiner oder gleich *.endsel*, außer es wurde von der Regelsprache aus anders gesetzt.

Siehe auch

Attribut *.endsel*

2.341.2 filereq

Es kann nur ein Wert vorgegeben werden, das heißt *.value* muss skalar sein. Das Attribut *.value* muss einen vollständigen Pfad enthalten, bei dem das Verzeichnis dem Attribut *.directoy* entspricht.

Der Standardwert von *.startsel* ist *false*.

Hinweise

- » Der vorgegebene Wert wird im Eingabefeld des Datei- bzw. Verzeichnisdialogs angezeigt, aber nicht in deren Auswahlliste selektiert. Dies ist eine Eigenschaft der WINDOWS- bzw. MOTIF-Systemdialoge.
- » Unter MICROSOFT WINDOWS ist nur eine Vorgabe für das Laden und Speichern von Dateien möglich, nicht aber für die Verzeichnisauswahl.

Siehe auch

Attribute *.directory*, *.value*

2.342 .starttime

Mit Hilfe dieses Attributes wird definiert, wann der angegebene Timer zum ersten Mal aktiv werden soll. Dies kann dabei sowohl eine relative als auch eine absolute Zeit sein. *.starttime* kann bei der absoluten Zeitdefinition oder beim Zeitabstand angegeben werden.

Definition

<i>Datentyp</i> string	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_starttime Datentyp: DT_string		<i>COBOL</i> Bezeichner: AT-starttime Datentyp: DT-string
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> timer	

2.343 .state

Das *.state*-Attribut definiert den Aktivierungszustand einer Tristate-**checkbox**.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_state	Bezeichner: AT-state
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	checkbox

Wertebereich

state_unchecked

Normaler Zustand – das Objekt ist deaktiviert; entspricht *.active := false*.

state_checked

Das Objekt ist aktiviert.

state_indeterminate

Das Objekt befindet sich in einem undefinierten Zustand; *.active* ist in diesem Fall undefiniert (*false*).

Nur möglich wenn *.style = 3* ist.

Ein Setzen des Attributs *.state* ist auch möglich, wenn *.style = 2* ist – wobei dann *state_indeterminate* als *state_unchecked* interpretiert wird. Dadurch wird dann *.active* ebenfalls umgeschaltet.

Wenn durch die Zuweisung eines Wertes ein Tristatebutton in eine **checkbox** umgewandelt wird, ist das Attribut *.active* immer mit *false* belegt

2.344 .static

Dieses Attribut liefert die Information, ob ein **Modul** entladen werden darf oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_static	Bezeichner: AT-static
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	import

2.345 .statusbar

Mit Hilfe dieses Attributes kann an einem Fenster die zugehörige Statusbar erfragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>object</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_statusbar	Bezeichner: AT-statusbar
Datentyp: DT_object	Datentyp: DT-object

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

2.346 .statushelp

In diesem Attribut wird der Hilfetext abgelegt, der dargestellt wird, wenn sich die Maus über dem Objekt befindet und falls das Vaterfenster eine sichtbare Statusbar besitzt, die eine gültige *.helppos*-Definition enthält. Ist das Objekt ein Menüitem, wird dessen Hilfetext dargestellt werden, wenn es den Focus besitzt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string, object [text]</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_statushelp	Bezeichner: AT-statushelp
Datentyp: DT_string, DT_text	Datentyp: DT-string, DT-text

Klassifizierung
Standardattribut

Ein Leerstring zeigt keinen Hilfetext an.

Ist im Attribut ein Nullstring (null) angegeben, dann wird die Statushilfe vom Vater benutzt. Diese Abfrage-Hierarchie wird nur bis zum Vaterfenster verfolgt; Dialoge und Module besitzen keine Statushilfe.

In der Statushilfe wird immer nur die Hilfe zu den Objekten aus dem Hauptfenster dargestellt. Es ist also nicht möglich, mehrere Hauptfenster aber nur eine Statusbar für die gesamte Statushilfe zu besitzen.

2.347 .style

Das Attribut *.style* bestimmt die Darstellung und das Verhalten des Objekts.

Definition

Datentyp	Zugriff	changed-Ereignis
<i>integer</i> (menusep)	get, set	ja
<i>integer, class</i> (checkbox , image , menuitem)		nein (font)
<i>integer, string</i> (listview)		
<i>class</i> (datetime , poptext , toolbar)		
<i>enum</i> (filereq , font)		
<i>datatype</i> (spinbox)		

C	COBOL
Bezeichner: AT_style	Bezeichner: AT-style
Datentyp: DT_integer (menusep)	Datentyp: DT-integer (menusep)
Datentyp: DT_integer, DT_class (image , menuitem)	Datentyp: DT-integer, DT-class (image , menuitem)
Datentyp: DT_integer, DT_class/DT_enum (checkbox)	Datentyp: DT-integer, DT-class/DT-enum (checkbox)
Datentyp: DT_integer, DT_string (listview)	Datentyp: DT-integer, DT-string (listview)
Datentyp: DT_class (datetime , poptext , toolbar)	Datentyp: DT-class (datetime , poptext , toolbar)
Datentyp: DT_enum (filereq , font)	Datentyp: DT-enum (filereq , font)
Datentyp: DT_datatype (spinbox)	Datentyp: DT-datatype (spinbox)

Klassifizierung	Objekte
Objektspezifisches Attribut	checkbox, datetime, filereq, font, image, listview, menuitem, menusep, poptext, spinbox, toolbar

2.347.1 checkbox, image, listview, menuitem und menusep

checkbox

Das Attribut *.style* bestimmt, ob die **checkbox** zwei oder drei Zustände annehmen kann.

Beim Schreibzugriff („set“) können Werte vom Datentyp *integer* und *class/enum* angegeben werden. Lesezugriff („get“) liefert immer einen Wert vom Datentyp *integer*.

Wertebereich

2 | checkbox

Die Checkbox kann die beiden Zustände „an“ und „aus“ annehmen.

Ihr Zustand kann über das Attribut *.active* abgefragt und gesetzt werden.

Interaktive Zustandsänderungen durch den Anwender lösen *activate*- bzw. *deactivate*-Ereignisse aus.

3 | *tristate*

Die Checkbox kann die drei Zustände „an“, „aus“ und „unbestimmt“ annehmen.

Ihr Zustand kann über das Attribut *.state* abgefragt und gesetzt werden.

Interaktive Zustandsänderungen durch den Anwender lösen **keine** *activate*- bzw. *deactivate*-Ereignisse aus.

Beim Setzen des Wertes 3 wird die Checkbox im Zustand „unbestimmt“ dargestellt.

image (Bild)

Mit dem Attribut *.style* kann beim **image**-Objekt eingestellt werden, dass es zwei Zustände darstellen kann oder – unter MICROSOFT WINDOWS – bei Mausklick ein Kontextmenü öffnet. Damit lässt sich das Objekt als individuelle Checkbox oder als bebildertes Menüelement verwenden.

Beim Schreibzugriff („set“) können Werte vom Datentyp *integer* und *class* angegeben werden. Lesezugriff („get“) liefert immer einen Wert vom Datentyp *integer*.

Wertebereich

0 | *pushbutton*

Das **image**-Objekt ist wie ein Pushbutton selektierbar.

2 | *checkbox*

Das **image**-Objekt kann zwischen den Zuständen „aktiv“ und „inaktiv“ umgeschaltet werden. Jedem Zustand kann mit dem Attribut *.picture[enum]* ein eigenes Bild (**tile**-Ressource) zugeordnet werden.

3 | *menubox*

Das **image**-Objekt hat das Verhalten einer **menubox**. Bei einem Mausklick wird das Kontextmenü geöffnet. In diesem Stil wird **kein** *select*-Ereignis erzeugt.

Bei geöffnetem Kontextmenü werden die Bilder *.picture[tile_active]* bzw. *.picture[tile_active_mouse_over]* angezeigt.

Ist kein Kontextmenü vorhanden, wird keine Menüaktion ausgelöst. Der Zustand des **image**-Objekts wird dennoch entsprechend angepasst und das zum Zustand gehörende Bild aus dem *.picture[enum]*-Attribut angezeigt.

Der genaue Ablauf der Aktion ist abhängig vom jeweiligen Fenstersystem. Auch die Behandlung des Fokus erfolgt so, wie es für Menüs auf dem jeweiligen Fenstersystem üblich ist.

Verfügbarkeit

Nur MICROSOFT WINDOWS.

listview

Dieses Attribut bestimmt die Darstellungsart des **listviews**.

Beim Schreibzugriff („set“) können Werte vom Datentyp *integer*, *string* und **text**-Ressourcen angegeben werden. **text**-Ressourcen werden automatisch zu *string* gewandelt.

Lesezugriff („get“) liefert immer einen Wert vom Datentyp *integer*.

Wertebereich

0 | "icon" | "picture"

Symbolansicht mit großen Symbolen.

Anordnung der Listeneinträge zuerst von links nach rechts, dann von oben nach unten.

Das Symbol wird über der Beschriftung des Eintrags angezeigt.

1 | "smallicon" | "smallpicture"

Symbolansicht mit kleinen Symbolen.

Anordnung der Listeneinträge zuerst von links nach rechts, dann von oben nach unten.

Das Symbol wird links von der Beschriftung des Eintrags angezeigt.

2 | "list"

Listenansicht mit kleinen Symbolen.

Anordnung der Listeneinträge zuerst von oben nach unten, dann von links nach rechts.

Das Symbol wird links von der Beschriftung des Eintrags angezeigt.

3 | "detail" | "report"

Liste mit kleinen Symbolen, bei der die Listeneinträge untereinander angeordnet werden.

Links neben der Beschriftung wird ein kleines Symbol angezeigt.

Zu jedem Listeneintrag werden Detailinformationen in mehreren Spalten angezeigt.

Die Darstellung ähnelt einer Tabelle.

4 | "tile"

Darstellung der Listeneinträge als Kacheln mit großem Symbol links sowie Beschriftung rechts.

Anordnung der Listeneinträge untereinander.

Bei ungültigen Werten wird der Standardwert 0 verwendet. Der Attributwert wird allerdings nicht geändert.

menuitem (Menüeintrag)

Das Attribut definiert die Art des Menüeintrags.

Beim Schreibzugriff („set“) können Werte vom Datentyp *integer* und *class* angegeben werden. Lesezugriff („get“) liefert immer einen Wert vom Datentyp *integer*.

Bei **menuitems**, die direktes Kind eines **Fensters** sind, wird das *.style*-Attribut ignoriert.

Wertebereich

0 | *pushbutton*

Normaler Menüeintrag, der üblicherweise zum Auslösen einer Aktion verwendet wird.

1 | *radiobutton*

Menüeintrag mit zwei Zuständen und dem Verhalten eines Radiobuttons.

Im Allgemeinen werden mehrere Menüeinträge dieses Stils verwendet, um über das Menü eine von mehreren sich gegenseitig ausschließenden Optionen auszuwählen.

Eine Gruppe von **menuitems** mit *.style = radiobutton*, in der jeweils nur einer der Menüeinträge aktiviert sein kann, wird durch den Menüanfang, das Menüende oder **Menüseparatoren** begrenzt.

2 | *checkbox*

Menüeintrag mit zwei Zuständen und dem Verhalten einer Checkbox, der in der Regel dazu verwendet wird über das Menü eine Einstellung vorzunehmen.

menusep (Menüseparator)

Das Attribut definiert das Aussehen der Menütrennlinie.

Wertebereich

- 0**
Standard-Trennlinie des jeweiligen Fenstersystems.
- 1**
Einfache Linie.
- 2**
Doppelte Linie.

Hinweis zum IDM für Windows

Das *.style*-Attribut des **Menüseparator**s wird vom IDM FÜR WINDOWS ignoriert.

2.347.2 *datetime*, *poptext* und *toolbar*

datetime

Das Attribut bestimmt das Aussehen und die Art der Bedienung des **datetime**-Objekts.

Wertebereich

- poptext***
Zum Auswählen eines Datums kann ein Kalender geöffnet werden.
- spinbox***
Werte können über eine Spinbox eingestellt werden.

Hinweis

Das Ändern des Attributs im sichtbaren Zustand führt zu einem Zurücksetzen des Objekts und sollte vermieden werden.

poptext (Combobox)

Beim **poptext** bestimmt das *.style*-Attribut wie die Liste dargestellt wird und ob der Inhalt des Eingabefelds vom Anwender bearbeitet werden kann.

Wertebereich

- edittext***
Die Liste ist geschlossen und kann vom Anwender auf- und zugeklappt werden. Der Inhalt des Eingabefelds lässt sich bearbeiten.
Der Anwender kann einen vorhandenen Eintrag aus der Liste auswählen oder einen anderen Wert eingeben.

listbox

Die Liste ist immer geöffnet und der Inhalt des Eingabefelds lässt sich bearbeiten.
Der Anwender kann einen vorhandenen Eintrag aus der Liste auswählen oder einen anderen Wert eingeben.

poptext

Die Liste ist geschlossen und kann vom Anwender auf- und zugeklappt werden. Der Inhalt des Eingabefelds lässt sich **nicht** bearbeiten.
Der Anwender kann nur einen Eintrag auswählen, der bereits in der Liste vorhanden ist.

toolbar

Die **toolbar** kann unter Qt zwei unterschiedliche Ausprägungen haben, die über das Attribut *.style* eingestellt werden. Standardmäßig ist der Stil **toolbar** aktiv, der optisch den bekannten Toolbars entspricht.

Wertebereich

notepage

Toolbars verwenden einen Dock-Bereich, der zwischen Toolbars und innerem Bereich liegt.
„Tabbed Toolbars“ und „Nested Toolbars“ sind möglich.

toolbar

Herkömmliche Toolbar, entsprechend der bekannten IDM-Toolbar.

Der Stil *notepage* erlaubt ein Verschachteln mehrerer **toolbars** in einem Dockbereich und ihre Anordnung in Form von Registerkarten (siehe Kapitel „Besonderheiten des Fensters unter Qt“ beim Objekt *window* in der „Objektreferenz“ für die entsprechenden Steuerungsoptionen). Die Toolbars haben dann eine Titelzeile und können über ihre Titelschaltflächen ausgedockt und geschlossen werden.

toolbars unterschiedlichen Stils, die im selben Dockbereich definiert werden, können nicht gemischt werden und werden entsprechend ihrem Stil gruppiert angeordnet. Dabei werden Toolbars mit dem Stil **toolbar** immer am äußeren Fensterrand positioniert und Toolbars mit dem Stil *notepage* immer zwischen dem Clientbereich des Fensters und den Toolbars mit dem Stil **toolbar** (siehe „Abbildung 2“).

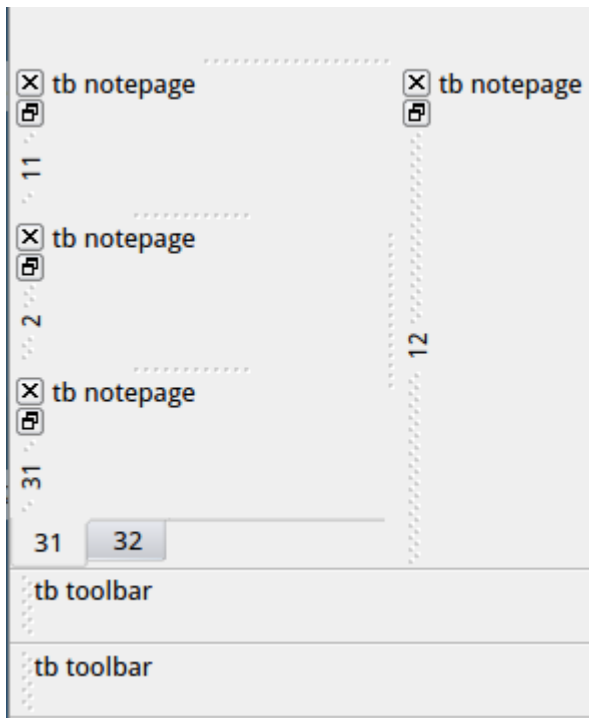


Abbildung 2: Toolbars verschiedenen Stils im unteren Dockbereich (dock_down)

Verfügbarkeit

Das Attribut `.style` wird bei der **toolbar** nur vom IDM FÜR QT unterstützt.

Siehe auch

Objekte toolbar, window

Attribut `.options[enum]`

2.347.3 filereq (Filerequestor)

Das Attribut bestimmt den angezeigten System-Dialog und damit den Verwendungszweck (Modus) des **Filerequestors**.

Wertebereich

fr_directory

System-Dialog zur Auswahl eines Verzeichnisses.

fr_load

System-Dialog zum Öffnen von Dateien.

fr_save

System-Dialog zum Speichern von Dateien.

Besonderheiten des IDM für Windows

Für die verschiedenen Modi werden unterschiedliche Systemdialoge verwendet. Deren Merkmale wie Beschriftung (z.B. der Schaltflächen), Farbe und Font sind überwiegend für den jeweiligen Verwendungszweck vorgegeben und nicht redefinierbar.

Der Modus *fr_directory* ist restriktiver in der Behandlung der Attribute *.directory* und *.pattern*. Außerdem erlaubt er keine Eingabe eines noch nicht vorhandenen Verzeichnisses.

2.347.4 font

Bei der **font**-Ressource bestimmt das Attribut *.style*, welcher Schriftstil (Schriftschnitt) des ausgewählten Zeichensatzes verwendet wird. Um bessere Unabhängigkeit und Kombinationen zwischen Face/Slant und Weight zu gewährleisten, wurde das Attribut *.style* auf die beiden Attribute *.face* und *.weight* umgestellt. Das *.style*-Attribute kann aber weiterhin verwendet werden.

Wertebereich

face_default

Normale, unveränderte Zeichendarstellung.

face_light

Leichgewichtige Zeichendarstellung.

face_normal

Normale, unveränderte Zeichendarstellung.

face_medium

Schriftstärke zwischen *face_normal* und *face_demibold*.

face_demibold

Schriftstärke zwischen *medium* und *bold*.

face_bold

Fette Zeichendarstellung.

face_black

Extra fette Zeichendarstellung.

face_italic

Kursive Zeichendarstellung.

Im Unterschied zu *face_oblique* werden meist spezielle, kursive Zeichen verwendet.

face_oblique

Geneigte, schräge Zeichendarstellung.

Im Unterschied zu *face_italic* werden die geneigten Zeichendarstellungen meist aus den regulären Zeichen abgeleitet.

face_oblique entspricht unter MICROSOFT WINDOWS *face_italic*.

face_roman

Aufrechte, gerade Zeichendarstellung (wird unter MICROSOFT WINDOWS ignoriert).

2.347.5 spinbox

Bei der **spinbox** definiert das Attribut *.style* die Art der angezeigten Werte.

Wertebereich

integer

Es werden die Zahlenwerte aus dem Intervall *.minvalue* ... *.maxvalue* angezeigt.

Der aktuelle Wert kann mit dem Attribut *.curvalue* abgefragt und gesetzt werden.

string

Es werden die Texte aus dem Attribut *.text[integer]* angezeigt.

Der Index des aktuellen Wertes kann mit dem Attribut *.activeitem* abgefragt und gesetzt werden.

void

Die Anzeige im Kindobjekt wird von der Anwendung gesteuert.

Das Weiterschalten der Werte erfolgt wie bei *.style = integer*, allerdings wird der im Kindobjekt (***edittext*** bzw. ***statictext***) angezeigte Wert nicht automatisch geändert sondern muss von der Anwendung gesetzt werden.

2.348 .style[enum]

Mit diesem Attribut werden bestimmte Darstellungseigenschaften der **progressbar** bzw. des **treeview** festgelegt.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_style	Bezeichner: AT-style
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	progressbar, treeview

progressbar

Das Attribut *.style[enum]* bestimmt das Aussehen der Fortschrittsanzeige.

Indexbereich

style_continuous

Gibt an, ob der Balken kontinuierlich (*true*, Standardwert) oder in Blöcken (*false*) gezeichnet wird.

Hinweis zum IDM FÜR QT

.style[continuous] wird nicht unterstützt. QT zeichnet den Fortschrittsbalken abhängig vom verwendeten UI-Stil („cleanlooks“, „plastique“ usw.). Standardmäßig wird ein durchgehender Balken gezeichnet, bei dem eine Unterteilung durch Farbwechsel angedeutet wird. Sobald man *.fgc* setzt, wird der Balken jedoch durchgehend in dieser Farbe gezeichnet.

style_labeled

Gibt an, ob der Balken eine ganzzahlige Prozentangabe als Beschriftung erhält (*true*, Standardwert) oder keine Beschriftung besitzt (*false*).

Die Beschriftung wird nur bei horizontaler Ausrichtung dargestellt.

treeview

Dieses Attribut des **treeview** erlaubt es, bestimmte Darstellungseigenschaften zu aktivieren. Standardmäßig sind diese Eigenschaften ausgeschaltet.

Indexbereich

style_buttons

Allen Einträgen außer dem obersten Vater wird ein +/- Knopf zum Auf- und Zuklappen von Teilbäumen vorangestellt.

style_lines

Zwischen Vater und Kindern werden Linien gezeichnet.

style_root

Wenn *style_lines* aktiv ist, werden die obersten Väter durch Linien miteinander verbunden.
Ist *style_buttons* aktiv, erhalten auch die obersten Väter einen +/- Knopf.

Diese unterschiedlichen Stile ergeben dann folgendes Aussehen:

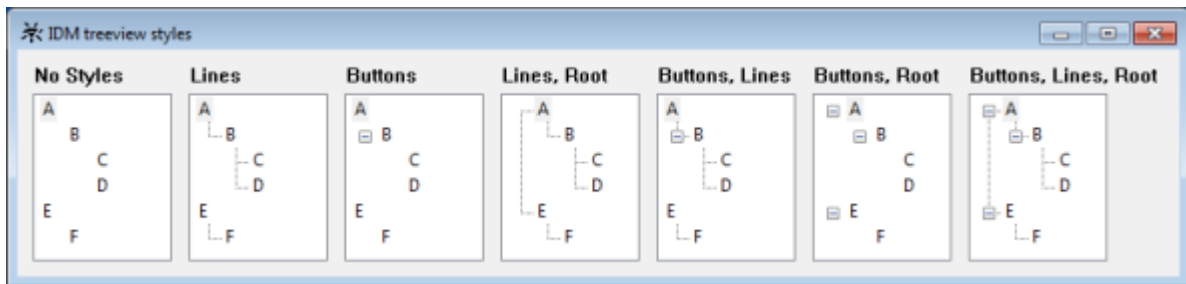


Abbildung 3: Möglichkeiten der treeview-Darstellung

Anmerkung

Die drei Stile werden nur von der Windows-Version des DM unterstützt. Die textuelle Simulation des Treeviews auf den anderen Plattformen bietet nur die Option *.style[style_buttons]* an. Diese ist vergleichbar mit der Darstellung "Buttons, Root" aus dem obigen Bild. Auch hier wird "+" bzw. "-" zur Darstellung eines geschlossenen bzw. offenen Teilbaums mit Kindern verwendet und "." um einen Eintrag ohne Kinder darzustellen.

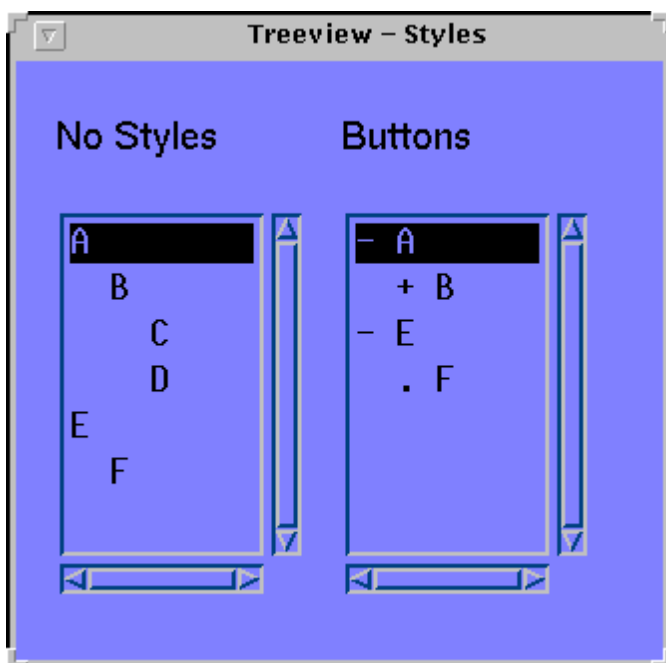


Abbildung 4: treeview unter Motif

2.349 .subcontrol[integer]

Dieses Attribut greift auf das I-te **Subcontrol** eines **Control**- oder **Subcontrol**-Objekts zu.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

C
Bezeichner: AT_subcontrol
Datentyp: DT_object

COBOL
Bezeichner: AT-subcontrol
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.350 .subcontrolcount

Dieses Attribut erfragt die Anzahl der **Subcontrols** in einem **Control**- oder **Subcontrol**-Objekt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

C
Bezeichner: AT_subcontrolcount
Datentyp: DT_integer

COBOL
Bezeichner: AT-subcontrolcount
Datentyp: DT-integer

Klassifizierung
Hierarchieattribut

2.351 .sysmodal

Das Attribut gibt an, ob eine Dialogbox oder **messagebox** vor allen anderen Fenstern auf dem Desktop dargestellt wird.

Bei *.sysmodal = true* wird der jeweilige IDM-Dialog bis zur Quittierung der Dialogbox oder **messagebox** durch den Anwender gesperrt, d.h. der Anwender kann nicht auf ein Fenster dieser Anwendung zugreifen. Andere Applikationen können hingegen noch bedient werden, allerdings bleibt die Dialogbox oder **messagebox** dabei permanent im Vordergrund, kann also nicht verdeckt werden.

Bei *.sysmodal = false* (Standardwert) beschränkt sich die Modalität auf den jeweiligen IDM-Dialog. Die Dialogbox oder **messagebox** kann also von anderen Applikationen verdeckt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_sysmodal	Bezeichner: AT-sysmodal
Datentyp: DT_boolean	Datentyp: DT-boolean

Standardwert
false

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	messagebox, window

Anmerkung zum IDM für Motif

Der IDM versucht das oberste modale Fenster (**Dialogbox**, **filereq** oder **messagebox**) vor anderen Fenstern der jeweiligen IDM-Applikation zu halten. Der IDM kann keine Reihenfolge im Hinblick auf andere Fenster und Dialogboxen garantieren, da die „Stacking-Order“ vom Fenstermanager bestimmt wird.

Anmerkung zum IDM für Qt

Das Attribut *.sysmodal* wird von der **messagebox** des IDM FÜR QT **nicht** unterstützt.

2.352 .systemerror

Dieses Attribut wird beim Auftreten eines Fehlers bei Benutzung der Applikationsfunktionalität mit dem Fehlerstring, den das System liefert, gesetzt und kann dann im Fehlerfall abgefragt werden. Dies wird nur für die Fehlerarten *error_network* und *error_file* unterstützt; in allen anderen Fehlerarten wird ein Leerstring zurückgegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_systemerror	Bezeichner: AT-systemerror
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Es ist zu beachten, dass der Fehlerstring in seiner Formatierung und Sprache nicht vom IDM beeinflusst werden kann. Ebenso wenig kann der IDM sicherstellen, dass das System für jeden Fehlerzustand einen adäquaten Fehlerstring bereitstellt.

Siehe auch

Attribut *.errorcode*

2.353 .systemid

Systemkennung des DOM-Knotens. Das Attribut ist nur verfügbar, wenn das Attribut *.nodetype* entweder den Wert *nodetype_entity* oder den Wert *nodetype_notation* besitzt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	<i>get</i>

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_systemid	Bezeichner: AT-systemid
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	doccursor

Das Attribut ist für den XML-Cursor verfügbar. Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

Siehe auch

Attribut *.nodetype*

2.354 .tabalignment

Mit Hilfe dieses Attributes wird unter MICROSOFT WINDOWS beim Objekt **notebook** definiert, wie der Text in seinen Reitern (Tabs) ausgerichtet ist.

Vom IDM FÜR MOTIF wird dieses Attribut **nicht** ausgewertet.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_tabalignment	Bezeichner: AT-tabalignment
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	notebook

Wertebereich

0

zentriert

Wenn *.majortabwidth* und *.majortabheight* ebenfalls beide auf den Wert 0 gesetzt sind, werden die Breite und die Höhe individuell, also für jeden Tab einzeln, gesetzt.

1 (Standard)

linksbündig

Alle Tabs erhalten die selbe Breite und Höhe.

Hinweis

Ein für die gesetzte Tabbreite zu langer Text wird unabhängig von den getroffenen Einstellungen immer linksbündig dargestellt (also niemals zentriert).

Siehe auch

Attribute *.majortabheight*, *.majortabwidth*

2.355 .tabshape

Definiert die Form der Major- und Minortabs.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_tabshape	Bezeichner: AT-tabshape
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	notebook

Wertebereich

shape_square (Standard)

Die Reiter (Tabs) sind rechteckig.

shape_rounded

Die Reiter (Tabs) haben abgerundete Ecken.

shape_polygon

Die Reiter (Tabs) haben abgeschrägte Ecken.

shape_chamfered

Die Reiter (Tabs) sind schräg (nur IDM FÜR WINDOWS).

2.356 .tabtype

Mit Hilfe dieses Attributes wird beim Objekt **notepage** der Typ eines Tabs beschrieben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
enum	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_tabtype	Bezeichner: AT-tabtype
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	notepage

Wertebereich

tab_major (Standard)

Der Reiter (Tab) ist ein Majortab oder Hauptindex.

tab_minor

Der Reiter (Tab) ist ein Minortab oder Nebenindex.

2.357 .target

Dieses Attribut bestimmt das Verhalten des Objekts als Ziel einer Drag & Drop oder Clipboard-Operation. Als Werte werden Ressourcen vom Typ **target** verwendet.

Beim **doccursor** enthält das Attribut den Namen der Instruktion eines DOM-Knotens. Der Wert entspricht dem Wert des Attributes *name*. Das Attribut ist nur verfügbar, wenn das Attribut *nodetype* den Wert *nodetype_processing_instruction* besitzt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
object [target]	get, set	ja
string (doccursor)	get (doccursor)	

C

Bezeichner: AT_target

Datentyp: DT_target

Datentyp: DT_string (**doccursor**)

COBOL

Bezeichner: AT-target

Datentyp: DT-target

Datentyp: DT-string (**doccursor**)

Klassifizierung

Standardattribut

doccursor

Dieses Attribut wird nicht vererbt, da es sich auf eine Laufzeiteigenschaft bezieht.

Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.358 .terminal

Mit Hilfe dieses Setup-Objekt-Attributes kann bei ALPHAWINDOWS die Art des Terminals abgefragt werden, z.B. *vt200*.

Definition

Datentyp
string

Zugriff
get

C
Bezeichner: AT_terminal
Datentyp: DT_string

COBOL
Bezeichner: AT-terminal
Datentyp: DT-string

Klassifizierung
Objektspezifisches Attribut

Objekte
setup

2.359 .terminaltype

Mit Hilfe dieses Setup-Objekt-Attributes kann bei ALPHAWINDOWS der Typ des Terminals abgefragt werden, z.B. */dev/tty*.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	get

C
Bezeichner: AT_terminaltype
Datentyp: DT_string

COBOL
Bezeichner: AT-terminaltype
Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

2.360 .text

Bei diesem Attribut handelt es sich um den statischen Text eines Objektes.

Bei einem **image** wird dabei der Text definiert, der bei dem Bild erscheinen soll.

Bei einer **notepage** wird mit dem Attribut *.text* (Default null) wird der Text definiert, der in der Statuszeile erscheinen soll. Falls kein Text definiert ist, enthält die betreffende Notepage keine Statuszeile.

Beim **doccursor** enthält das Attribut die Werte aller Unterknoten des DOM-Knotens. Es wird ein String geliefert, der den Text aller Unterknoten repräsentiert. Das Attribut ist hauptsächlich hilfreich, wenn man den Text eines XML-Elements benötigt und nicht erst zu dem Kindknoten, der den Text enthält, navigieren möchte.

Bei einem **statictext**, der ein Kindobjekt einer **spinbox** ist, darf das Attribut *.text* nicht genutzt werden. Die Wertsetzung erfolgt ausschließlich über die entsprechenden Attribute des **spinbox**-Objekts selbst.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string, object [text]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_text	Bezeichner: AT-text
Datentyp: DT_string, DT_text	Datentyp: DT-string, DT-text

Klassifizierung
Textattribut

doccursor

Das Setzen dieses Attributs löscht alle Kindknoten und fügt einen neuen Textknoten an.

Dieses Attribut wird nicht vererbt, da es sich auf eine Laufzeiteigenschaft bezieht.

Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.361 .text[enum]

Dieses indizierte Attribut erlaubt es, die Beschriftung von Dialogelementen eines **filereq**-Objektes vorzunehmen.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_text	Bezeichner: AT-text
Datentyp: DT_string	Datentyp: DT-string

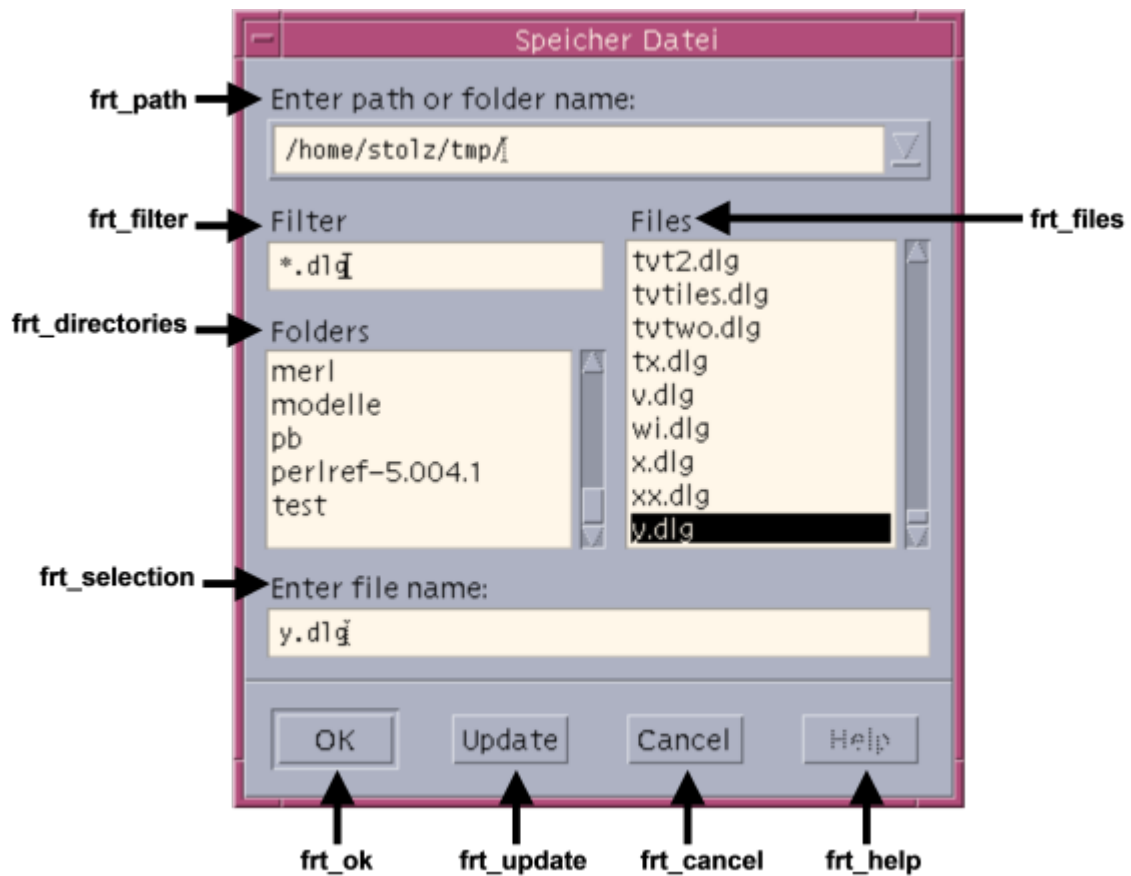
<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	filereq

Indexbereich

<i>frt_cancel</i>	Beschriftung der „Abbrechen“-Schaltfläche.
<i>frt_directories</i>	Beschriftung der Verzeichnisliste.
<i>frt_files</i>	Beschriftung der Dateiliste.
<i>frt_help</i>	Beschriftung der „Hilfe“-Schaltfläche.
<i>frt_nomatch</i>	Text, der in der Dateiliste angezeigt wird, wenn es kein Element gibt, das dem Namensmuster entspricht.
<i>frt_ok</i>	Beschriftung der „OK“-Schaltfläche.
<i>frt_path</i>	Beschriftung des Eingabefelds für den Verzeichnispfad.
<i>frt_pattern</i>	Beschriftung des Eingabefelds für das Namensmuster.
<i>frt_selection</i>	Beschriftung des Eingabefelds für das ausgewählte Element.
<i>frt_update</i>	Beschriftung der „Aktualisieren“-Schaltfläche.

Besonderheiten

Dieses Attribut wird nur von MOTIF unterstützt. Die folgende Abbildung zeigt die Zuordnung der einzelnen Einträge zu den Elementen des Fensters. Die Zuordnung gilt für die Dateiauswahl und die Verzeichnisauswahl.



2.362 .text[integer]

Mit diesem Attribut können einzelne Texte eines **poptexts** (Combobox) gesetzt werden.

Wenn die Werte des Index ≥ 1 und $\leq .itemcount$ sind, erhält man den Text des Eintrags. Beim Wert 0 erhält man den aktuell ausgewählten Text.

Bei einer **spinbox** definiert das Attribut die Werte, die bei `.style = string` durchlaufen und im zugehörigen **edittext** bzw. **statictext** angezeigt werden.

Definition

<i>Datentyp</i> <i>string</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_text Datentyp: DT_string		<i>COBOL</i> Bezeichner: AT-text Datentyp: DT-string
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> poptext, spinbox	

2.363 .textbgc

Mithilfe dieses Attributs kann die Hintergrundfarbe einer Beschriftung – soweit vom Objekt unterstützt – angegeben werden. Ist der Wert gleich *null* (es wurde kein Wert gesetzt) so wird die Standardfarbe des Systems benutzt.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

C
Bezeichner: AT_textbgc
Datentyp: DT_color

COBOL
Bezeichner: AT-textbgc
Datentyp: DT-color

Klassifizierung
Objektspezifisches Attribut

2.364 .textfgc

Mithilfe dieses Attributs kann die Farbe einer Beschriftung – soweit vom Objekt unterstützt – angegeben werden. Ist der Wert gleich *null* (es wurde kein Wert gesetzt) so wird die Standardfarbe des Systems benutzt.

Definition

<i>Datentyp</i> <i>object</i> [color]	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_textfgc Datentyp: DT_color		<i>COBOL</i> Bezeichner: AT-textfgc Datentyp: DT-color
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> progressbar	

2.365 .textwidth

Mit dem Attribut *.textwidth* kann bei einem **edittext** mit Formatierungen (RTF-Edittext) die maximale Textbreite in Pixeln gesetzt und abgefragt werden.

.textwidth ist ein Attribut des RTF-Edittextes und wie der RTF- Modus des Edittextes nur unter MICROSOFT WINDOWS verfügbar.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_textwidth	Bezeichner: AT-textwidth
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	edittext

Wird *.textwidth* auf 0 gesetzt, ergibt sich die maximale Textbreite wie folgt:

- » Ist *.hsb_visible = true* wird die Textbreite durch die breiteste Zeile im formatierten Text bestimmt.
- » Ist *.hsb_visible = false* wird der Text automatisch umgebrochen und die Textbreite entspricht der Breite des Edittextes ohne Ränder.

Ein Wert > 0 setzt die maximale Textbreite auf diesen Wert. Bei Bedarf ist eine horizontale Scrollbar vorhanden, deren Anzeige über das Attribut *.hsb_visible* gesteuert wird.

Siehe auch

Kapitel „Editierbarer Text mit Formatierungen (RTF-Edittext)“ in der „Objektreferenz“

2.366 .tile

Bei **Gruppierungsobjekten** gibt das Attribut *.tile* eine **tile**-Ressource an, die als Hintergrund gezeichnet wird. Die Art und Weise wie das Bild gezeichnet wird, wird durch das Attribut *.tilestyle* bestimmt.

Am **setup**-Objekt kann mit diesem Attribut die Mustervariante gesetzt und abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object [tile]</i>	get, set	ja
<i>integer (setup)</i>		nein (setup)
 <i>C</i>		 <i>COBOL</i>
Bezeichner: AT_tile		Bezeichner: AT-tile
Datentyp: DT_tile		Datentyp: DT-tile
Datentyp: DT_integer (setup)		Datentyp: DT-integer (setup)
 <i>Klassifizierung</i>	 <i>Objekte</i>	
Objektspezifisches Attribut	groupbox, layoutbox, notebook, notepage, setup, spinbox, splitbox, statusbar, toolbar, window	

Hinweis für das Fenster-Objekt (window)

Ein Hintergrundbild wirkt sich nur auf die eigentliche Fensterfläche, nicht jedoch auf Titelleiste, Menüleiste, Statusbar und eingedockte Toolbar aus.

Siehe auch

Attribut *.tilestyle*

2.367 .tiledpi

Über dieses Attribut kann der Anwender angeben für welche DPI-Auflösung die Tiles der Anwendung entworfen wurden. Dies ist insbesondere dann nötig, wenn Grafiken für eine bestimmte (und abweichende) Auflösung erstellt wurden und man sicher gehen möchte, dass diese in der Anwendung auch korrekt dargestellt werden. Die Größe eines Bildes/Musters wird dann ausgehend von diesem Wert auf den aktuell geltenden DPI-Wert skaliert.

Ebenso ist es natürlich auch möglich die Grafiken für eine hohe Auflösung zu erstellen und diese dann durch den IDM für niedrigere DPIs herunter skalieren zu lassen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i> (kann 0 sein)	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_tiledpi	Bezeichner: AT-tiledpi
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Das .tiledpi-Attribut wurde zur IDM-Version A.06.03.a im Rahmen der Unterstützung für hohe Auflösungen (HighDPI-Support) eingeführt.

Hinweis

Dieses Attribut kann nur nach erfolgter IDM-Initialisierung nicht aber in der laufenden Anwendung gesetzt werden.

Verfügbarkeit

Ab IDM-Version A.06.03.a

Siehe auch

Kapitel „HighDPI UnterstützungSupport“ im Handbuch „Programmiertechniken“

2.368 .tilestyle

Für **Gruppierungsobjekte** gibt das Attribut *.tilestyle* die Art und Weise an wie ein Hintergrundbild gezeichnet wird.

Das Attribut beschreibt die Position des Bildes (*.picture*) innerhalb des **image**-Objektes. Kein Bild wird hierbei behandelt wie ein Bild mit Breite und Höhe 0.

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
enum	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_tilestyle	Bezeichner: AT-tilestyle
Datentyp: DT_enum	Datentyp: DT-enum

Klassifizierung
Layoutattribut

Gruppierungsobjekte

Wertebereich

tilestyle_tiled

Das Hintergrundbild wird so oft nebeneinander und untereinander gezeichnet, bis der Hintergrund ausgefüllt ist.

tilestyle_centered (nicht unter MOTIF)

Das Hintergrundbild wird in der Mitte gezeichnet. Ist eine virtuelle Größe des Inhaltsbereichs gegeben, so wird diese zur Positionsrechnung herangezogen.

tilestyle_stretched (nicht unter MOTIF)

Das Hintergrundbild wird so in der Größe geändert, dass die gesamte Fläche ausgefüllt ist. Ist eine virtuelle Größe des Inhaltsbereichs gegeben, wird die virtuelle Fläche aufgefüllt.

tilestyle_parent_tile

Es wird die Definition des Vaterobjektes übernommen. Es wird der Eindruck erweckt, als ob das Objekt durchsichtig ist. Dieser Wert ist für die Objekte **statusbar**, **toolbar** und **window** nicht zulässig.

Anmerkung

Ist bei den Werten *tilestyle_centered* bzw. *tilestyle_stretched* eine Positionsrechnung bzw. Auffüllung bei vorhandener virtueller Größe nicht gewünscht, so sollte hier *tilestyle_parent_tile* verwendet werden und das Hintergrundbild am Vaterobjekt gesetzt werden (eventuell eine Groupbox als „Zwischenobjekt“ einfügen).

image

Wertebereich

tilestyle_icon

Das Bild befindet sich mit Abstand *.spacing* über dem Text. Es wird horizontal zentriert, der Text wird gemäß *.alignment* horizontal im **image**-Objekt angebunden (normal: zentriert). Vertikal wird Bild und Text als Einheit gesehen und diese zentriert. Sollte das Bild skalierbar sein, dann befindet sich der Text am unteren Rand und das Bild füllt den übrigen Raum aus.

tilestyle_left

Das Bild befindet sich am linken Rand des **image**-Objektes. Der Darstellungsbereich für den Text beginnt mit Abstand *.spacing* rechts daneben und reicht bis zum rechten Rand. Der Text wird gemäß *.alignment* in seinem Darstellungsbereich angebunden. Vertikal wird sowohl das Bild als auch der Text zentriert. Sollte das Bild skalierbar sein, dann befindet sich der Text ganz am rechten Rand und *.alignment* hat keine Bedeutung mehr. Das Bild füllt den übrigen Raum aus.

tilestyle_right

Das Bild befindet sich am rechten Rand des **image**-Objektes. Der Darstellungsbereich für den Text beginnt mit Abstand *.spacing* links daneben und reicht bis zum linken Rand. Der Text wird gemäß *.alignment* in seinem Darstellungsbereich angebunden. Vertikal wird sowohl das Bild als auch der Text zentriert. Sollte das Bild skalierbar sein, dann befindet sich der Text ganz am linken Rand und *.alignment* hat keine Bedeutung mehr. Das Bild füllt den übrigen Raum aus.

tilestyle_top

Das Bild befindet sich am oberen Rand des **image**-Objektes. Der Text wird mit Abstand *.spacing* darunter dargestellt. Das Bild wird horizontal zentriert, der Text wird gemäß *.alignment* horizontal im **image**-Objekt angebunden. Sollte das Bild skalierbar sein, dann befindet sich der Text am unteren Rand und das Bild füllt den übrigen Raum aus.

tilestyle_bottom

Das Bild befindet sich am unteren Rand des **image**-Objektes. Der Text wird mit Abstand *.spacing* darüber dargestellt. Das Bild wird horizontal zentriert, der Text wird gemäß *.alignment* horizontal im **image**-Objekt angebunden. Sollte das Bild skalierbar sein, dann befindet sich der Text am oberen Rand und das Bild füllt den übrigen Raum aus.

tilestyle_background

Das Bild wird als Hintergrund dargestellt. Der Text wird vor dem Bild dargestellt. Das Bild wird horizontal zentriert, der Text wird gemäß *.alignment* horizontal im **image**-Objekt angebunden. Vertikal wird sowohl das Bild als auch der Text zentriert. Sollte das Bild skalierbar sein füllt es den gesamten Raum aus.

Siehe auch

Attribute *.alignment*, *.spacing*

2.369 .timeout

Mit dem Attribut *.timeout* wird definiert, nach welcher Zeit (in Sekunden) ein Timeout wirksam werden soll, wenn der Benutzer nicht mit dem Programm arbeitet. Die Angabe der Zeitdauer erfolgt dabei nach demselben Schema wie beim **timer**-Objekt die Incrementaltime definiert wird.

Ist *.timeout* des Dialogs nicht 0, so setzt der IDM jedes mal, wenn er im Fenstersystem blockierend liest, diesen Timer auf. Wird dieser wirksam, wird für den **dialog** ein *deactivate*-Ereignis verschickt. Jedes Ereignis – außer anderen Timer – setzt diesen Timeout wieder zurück.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_timeout		Bezeichner: AT-timeout
Datentyp: DT_integer		Datentyp: DT-integer
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	dialog	

Mit Hilfe von Timeouts soll der IDM-Anwender in die Lage versetzt werden, kritische Operationen nach Ablauf einer gewissen Zeit abubrechen oder sonstige Aktionen einzuleiten. Dabei bezieht sich der Timeout immer auf Aktionen des Benutzers, nicht auf interne Rechenoperationen. Damit kann der Anwender z.B. beim Öffnen eines kritischen Fensters oder einer Leitung zu einem anderen Rechner sich einen Timeout aufsetzen. Dieser wird dann wirksam, wenn der Benutzer für eine gewisse Zeit nicht mit dem Programm arbeitet. Daraufhin kann das Programm das kritische Fenster schließen oder die Leitung wieder freigeben.

Anmerkung

Da mehrere Dialoge definiert werden können, können Sie auch mehrere Timeouts erhalten, die in den verschiedenen Dialogen definiert wurden.

Siehe auch

Objekt timer

2.370 .title

Dieses Attribut steht für den Text eines Objektes, meist den Text in der Titelzeile eines Fensters oder Dialogs.

Bei einer **notepage** definiert dieses Attribut die Beschriftung der Tabs. Defaultwert = 0. Ein Text ist nur vorhanden, wenn hier ein Wert $\neq 0$ angegeben ist. Die Beschriftung kann hierbei ein Text oder ein Bitmap sein.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string, object [text]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_title	Bezeichner: AT-title
Datentyp: DT_string, DT_text	Datentyp: DT-string, DT-text

Klassifizierung
Textattribut

Hinweis zur notepage

Bitte beachten Sie, dass im Normalfall ein Text angegeben werden sollte. Abhängig vom Fenstersystem kann es sein, dass Bitmaps nicht unterstützt werden, nur bestimmte Bitmaps zulässig sind oder immer ein Text vorhanden sein muss.

Hinweis zum filereq unter Microsoft Windows

Der Titel erscheint bei der Verzeichnisauswahl (Modus *fr_directory*) nur als Überschrift und nicht als Fenstertitel.

2.371 .titlebar

Das Attribut *.titlebar* definiert, ob das Fenster eine Titelleiste besitzt oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_titlebar	Bezeichner: AT-titlebar
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Hinweis

Unter MOTIF muss zusätzlich *.iconifiable* auf *false* gesetzt werden, damit keine Titelleiste unter Verwendung des Window Managers MWM erscheint.

2.372 .titlebgc

Mit diesem Attribut kann die Hintergrundfarbe der Titelleiste am **Fenster** definiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_titlebgc	Bezeichner: AT-titlebgc
Datentyp: DT_color	Datentyp: DT-color

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Anmerkung

In den aktuellen Fenstersystemen ist dieses Attribut wirkungslos. Das Aussehen des Fensterrahmens lässt sich von einer Applikation aus nicht beeinflussen.

2.373 .titlefgc

Mit diesem Attribut kann die Vordergrundfarbe der Titelleiste am **Fenster** definiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [color]	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_titlefgc	Bezeichner: AT-titlefgc
Datentyp: DT_color	Datentyp: DT-color

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	window

Anmerkung

In den aktuellen Fenstersystemen ist dieses Attribut wirkungslos. Das Aussehen des Fensterrahmens lässt sich von einer Applikation aus nicht beeinflussen.

2.374 .today

Das Attribut bestimmt, ob am unteren Rand des aufklappbaren Kalenders das heutige Datum angezeigt wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_today	Bezeichner: AT-today
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Standardwert</i>	<i>Vererbung</i>
<i>true</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	datetime

Wertebereich

- false*
Im Kalender wird das heutige Datum nicht angezeigt.
- true*
Im Kalender wird am unteren Rand das heutige Datum angezeigt.

2.375 .todaymarker

Dieses Attribut legt fest, ob das heutige Datum im aufklappbaren Kalender markiert wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_todaymarker	Bezeichner: AT-todaymarker
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Standardwert</i>	<i>Vererbung</i>
<i>true</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	datetime

Wertebereich

false

Im Kalender wird das heutige Datum nicht markiert.

true

Im Kalender wird das heutige Datum markiert.

2.376 .toolbar

Mit Hilfe dieses Attributes kann auf die Toolbar zugegriffen werden, zu der das Objekt gehört. Wenn das Objekt nicht zu einer Toolbar gehört, wird als Ergebnis das *null*-Objekt geliefert.

Definition

Datentyp
object

Zugriff
get

C
Bezeichner: AT_toolbar
Datentyp: DT_object

COBOL
Bezeichner: AT-toolbar
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.377 .toolbar[integer]

Mit Hilfe dieses Attributes kann auf die n-te Toolbar eines Fensters zugegriffen werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	ja

C
Bezeichner: AT_toolbar
Datentyp: DT_object

COBOL
Bezeichner: AT-toolbar
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.378 .toolbarcount

Mit Hilfe dieses Attributes kann die Anzahl der vorhandenen Toolbars in einem Fenster erfragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

C
Bezeichner: AT_toolbarcount
Datentyp: DT_integer

COBOL
Bezeichner: AT-toolbarcount
Datentyp: DT-integer

Klassifizierung
Hierarchieattribut

2.379 .toolhelp

Wenn dieses Attribut gesetzt ist, erscheint ein kleines Popup mit dem spezifizierten Text, sobald sich der Mauszeiger für kurze Zeit bewegungslos innerhalb des jeweiligen Objekts befindet. Das Popup wird automatisch nach einer gewissen Zeitspanne oder bei Bewegung des Mauszeigers wieder ausgeblendet.

Die Popups lassen sich für die Anzeige einer kurzen Erklärung zu einem Objekt einsetzen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i> [text]	get, set	ja

C

Bezeichner: AT_toolhelp

Datentyp: DT_text

COBOL

Bezeichner: AT-toolhelp

Datentyp: DT-text

Klassifizierung

Standardattribut

Unterstützung des Attributs durch Objekte

- » canvas
- » checkbox
- » edittext
- » image
- » listbox
- » poptext
- » pushbutton
- » radiobutton
- » rectangle (nur MICROSOFT WINDOWS)
- » scrollbar
- » spinbox
- » statictext
- » tablefield
- » treeview

Beispiel

```
child pushbutton PbOK
{
    .text      "OK";
    .toolhelp  "accepts file";
}
```

```
}
```

Hinweise zum IDM für Motif

- » Unter Motif lassen sich Vorder- und Hintergrundfarbe mittels X-Ressourcen einstellen, z.B.:

```
IDM*ToolHelpLabel.Background: red
IDM*ToolHelpLabel.Foreground: blue
```

- » Als Standard verwendet der IDM „LightYellow“ als Hintergrundfarbe und „Black“ als Vordergrundfarbe.
- » Der X-Toolkit-Name des Toolhelp-Objekts ist „ToolHelpLabel“.

Hinweise zum IDM für Windows

„WM_MOUSEMOVE“-Ereignisse, die von anderen Anwendungen provoziert werden, können das Ein- und Ausblenden der Toolhelp stören. Zum Beispiel kann es vorkommen, dass sie sich nach dem Umsetzen von *.toolhelp* öffnet, obwohl die Maus nicht bewegt wurde. Es kann auch sein, dass eine sichtbare Toolhelp nicht wieder ausgeblendet wird oder eine Toolhelp gar nicht eingeblendet wird.

Ein anwendungsgesteuertes Öffnen von Toolhelps ist nicht vorgesehen. Wenn dies gewünscht wird, kann mit der Funktion **DM_GetToolkitData()** das Attribut *AT_toolhelp* des **Setup**-Objekts abgefragt werden, um das Windows-Handle des „Tooltip Controls“ zu erhalten, das der ISA Dialog Manager für die Darstellung verwendet.

Mit dem folgenden Beispielcode kann die Toolhelp geöffnet werden:

```
#include <windows.h>
#include <commctrl.h>
#include IDMuser.h

void DML_default DM_ENTRY OpenToolhelp __0() {
    DM_ID idSetup = DM_ParsePath(
        (DM_ID) 0, (DM_ID) 0, "setup", 0, 0);

    if (idSetup != (DM_ID) 0) {
        HWND hwndToolhelp = (HWND) DM_GetToolkitData(
            idSetup, AT_toolhelp);

        if (hwndToolhelp != (HWND) 0) {
            SendMessage(hwndToolhelp, TTM_POPUP,
                (WPARAM) 0, (LPARAM) 0);
        }
    }
}
```

Die Funktion „OpenToolhelp()“ muss im Dialog entsprechend definiert sein. Die Windows-Message TTM_POPUP ist erst ab Version 6 der Common Controls Library (**comctl32.dll**) verfügbar.

2.380 .toolkit

Mit Hilfe dieses Setup-Objekt-Attributes kann die Toolkitart abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>enum</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_toolkit	Bezeichner: AT-toolkit
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Wertebereich

<i>toolkit_motif</i>
Das Toolkit ist MOTIF.
<i>toolkit_qt</i>
Das Toolkit ist QT.
<i>toolkit_windows</i>
Das Toolkit sind die „Common Controls“ von MICROSOFT WINDOWS.

2.381 .toolkit_string

Mit Hilfe dieses Setup-Objekt-Attributs kann das Toolkit abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_toolkit_string	Bezeichner: AT-toolkit-string
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

2.382 .toolkit_version

Mit Hilfe dieses Setup-Objekt-Attributs kann die Toolkit-Version abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_toolkit_version	Bezeichner: AT-toolkit-version
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Hinweis zum IDM für Windows

Ab IDM-Version A.06.01.a liefert das Attribut nicht mehr die Windows-Version für die der IDM kompiliert wurde, sondern die Version der „Common Controls DLL“ (**comctl32.dll**). Diese DLL bestimmt das Aussehen der Oberflächenobjekte, sodass nun ein Wert zurückgeliefert wird, der Aufschluss über das Aussehen und das Layout (z. B. Rahmenbreiten) der Oberfläche gibt.

Die Version wird als Dezimalzahl kodiert:

Majorversion * 100 + Minorversion

Außerdem können noch folgende vordefinierte Werte vorkommen:

- | | |
|--------|---|
| 0 | Keine „Common Controls DLL“ vorhanden. |
| 400 | Version konnte nicht bestimmt werden, da „Common Controls DLL“ älter als Version 4.71 ist. |
| 471 | „Common Controls DLL“ hat Version 4.71 oder Version konnte nicht bestimmt werden, da ein Fehler beim Aufruf von DllGetVersion auftrat. |
| 582 | „Common Controls DLL“ hat Version 5.82 oder Version ist 6.00 oder höher und „Visual Styles“ sind ausgeschaltet. |
| >= 600 | „Visual Styles“ sind eingeschaltet. |

2.383 .top_most

Mit Hilfe dieses Attributes können Fenster als ganz vorne liegend definiert werden. Fenster mit *.top_most = true* werden in der Z-Ordnung aller Toplevel-Fenster möglichst weit oben angeordnet, d.h. nur andere Fenster mit *.top_most = true* können in der Hierarchie weiter oben stehen. Bei Kindfenstern ist das Attribut wirkungslos.

Definition

<i>Datentyp</i> <i>boolean</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
<i>C</i> Bezeichner: AT_top_most Datentyp: DT_boolean		<i>COBOL</i> Bezeichner: AT-top-most Datentyp: DT-boolean
<i>Standardwert</i> <i>false</i>		
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> window	

2.384 .topitem

Dieses Attribut definiert, welcher Eintrag in einer **listbox** oder einem **treeview** an erster Stelle angezeigt wird.

Der Wert von *.topitem* wird korrigiert, wenn der als *.topitem* gesetzte Eintrag nicht bis an den oberen Rand des Objekts gescrollt werden kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_topitem	Bezeichner: AT-topitem
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, treeview

Hinweis zum treeview

Da der *.topitem* Eintrag immer sichtbar ist, kann der Teilbaum mit diesem Eintrag bei einem sichtbaren **treeview** programmatisch nicht geschlossen werden. Interaktiv kann dieser Teilbaum geschlossen werden, weil dafür ein übergeordneter Eintrag erreichbar sein muss. Dadurch ist gewährleistet, dass sich das aktuelle *.topitem* immer oberhalb des gesetzten *.topitem* befindet.

2.385 .tracefile

Mit Hilfe dieses Attributs kann während der Laufzeit über das Setup-Objekt der absolute Pfad des Tracefiles abgefragt werden.

Definition

Datentyp
string

Zugriff
get

C
Bezeichner: AT_tracefile
Datentyp: DT_string

COBOL
Bezeichner: AT-tracefile
Datentyp: DT-string

Klassifizierung
Objektspezifisches Attribut

Objekte
setup

2.386 .tracetime

Mit Hilfe dieses Attributs kann während der Laufzeit über das Setup-Objekt die Zeitausgabe in das Tracefile beeinflusst werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_tracetime	Bezeichner: AT-tracetime
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Wertebereich

0

Im Tracefile werden keine Zeiten aufgeführt

1

Dieser Wert kennzeichnet den Starttime-Modus. Bei diesem Modus werden alle Start- und Endezeiten mitprotokolliert, der Zeitverbrauch für einen einzelnen Aufbau kann dann aus der Differenz berechnet werden. Dabei wird in diesem Modus ausschließlich die System- und die Benutzerzeit betrachtet.

In diesem Modus erscheinen im Tracefile am Zeilenanfang die Zeiten im Format [hh:m-m:ss:uuu]:

- » hh = Stunden
- » mm = Minuten
- » ss = Sekunden
- » uuu = Millisekunden

2

Dieser Wert kennzeichnet den Tracetime-Modus. In diesem Modus wird der Zeitunterschied zum letzten mitprotokollierten Aufruf gegeben. In diesem Modus kann also relativ einfach erkannt werden, wie viel Zeit für einzelne Aktionen verbraucht wird.

In diesem Modus erscheinen im Tracefile die Zeitdifferenz zur letzten Trace-Ausgabe im Format [ss:uuu] am Zeilenanfang:

- » ss = Sekunden
- » uuu = Millisekunden

3

Dieser Wert kennzeichnet den Realtime-Modus. In diesem Fall wird für jede zu protokollierende Aktion die reale Zeit (Uhrzeit) im Tracefile abgelegt.

In diesem Modus erscheinen im Tracefile die Realzeit im Format [hh:mm:ss] am Zeilenanfang:

- » hh = Stunden
- » mm = Minuten
- » ss = Sekunden

Anmerkung

Die Option **-IDMtracetime <nr>** erlaubt auch das Setzen des Tracemodus über die Kommandozeile.

2.387 .tracing

Mit Hilfe dieses Attributes kann angegeben werden, ob das Tracing vollständig durchgeführt werden soll oder nicht.

Dieses Attribut hat nur Auswirkung, wenn die Startoption **-IDMtracefile** angegeben wurde. Sie können also nur das Tracing einschränken, falls überhaupt ein Tracefile erzeugt wird.

Warnung

Dieses Attribut sollte mit Vorsicht und nur temporär verwendet werden!

Es ist dafür gedacht, dass z.B. Passwörter, die vom Dialog Manager an Applikationsfunktionen übergeben werden, nicht im Tracefile erscheinen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_tracing	Bezeichner: AT-tracing
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Beispiel

```
setup.tracing := false; // switch off tracing
// actions
setup.tracing := true;  // switch on tracing again
```

.tracing[string]

Wenn das Attribut mit einem String indiziert ist, der einen Trace-Code darstellt, wird das Tracing für bestimmte Operationen (definiert durch den Trace-Code) ein- (*true*) und ausgeschaltet (*false*).

Siehe auch

Kapitel „Tracing (Ablaufverfolgung)“ im Handbuch „Entwicklungsumgebung“

2.388 .trailingdates

Dieses Attribut bestimmt, ob im aufklappbaren Kalender die an den aktuellen Monat angrenzenden Tage angezeigt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_trailingdates	Bezeichner: AT-trailingdates
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Standardwert</i>	<i>Vererbung</i>
<i>true</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	datetime

Wertebereich

false

Im Kalender werden nur Tage des eingestellten Monats angezeigt.

true

Im Kalender werden auch Tage des vorherigen und folgenden Monats angezeigt, sodass Wochen immer vollständig angezeigt werden.

2.389 .transformer[integer]

Über das *.transformer*-Attribut kann auf die **transformer** zugegriffen werden. Das Attribut wird mit dem Objektindex indiziert (ähnlich zu *.child*).

Das Attribut ist wie das *.record*-Attribut zu verwenden.

Das Attribut ist an den Klassen verfügbar, die auch **records** als Kinder haben können.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>object</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_transformer	Bezeichner: AT-transformer
Datentyp: DT_transformer	Datentyp: DT-transformer

Klassifizierung
Standardattribut

2.390 .transport

Mit Hilfe dieses Attributes wird definiert, welchen intern vorhandenen Transportmechanismus die Kommunikationsschicht verwenden soll. Dabei gibt es folgende Möglichkeiten: *"tcpip"*, *"dynlib"* usw.

Um die Verbindung über **SSL** durchzuführen kann bei der Spezifikation des Protokolls über *.transport* das Schema *"ssl://"* vorangestellt werden (Beispiel *.transport "ssl://tcpip"*;). Wird kein Protokoll vorgegeben kann „*://*“ entfallen (also *.transport "ssl"*;).

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_transport	Bezeichner: AT-transport
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Beispiel

Ohne SSL	Mit SSL
<pre>application App11 { .connect "localhost:4711"; } application App12 { .transport "tcpip-winsock"; .connect "localhost:4711"; }</pre>	<pre>application App11 { .transport "ssl"; .connect "localhost:4711"; } application App12 { .transport "ssl://tcpip-winsock"; .connect "localhost:4711"; }</pre>

Anmerkungen

- » Die Attribute *.transport*, *.connect* und *.exec* können nur geändert werden, wenn *.active = false* ist.
- » Auch beim Attribut *.connect* kann das Schema *"ssl://"* angegeben werden. Wenn bei beiden Attributen ein Schema angegeben ist, muss es identisch sein. Ein einmal angegebenes Schema *"ssl://"* lässt sich nicht mehr abstellen.

2.391 .type

Dieses Attribut definiert den Datentyp einer globalen Variablen oder eines benutzerdefinierten Attributs.

Bei **globalen Variablen** und **benutzerdefinierten Attributen** definiert das Attribut den Datentyp der Variablen bzw. des benutzerdefinierten Attributes.

Die Abfrage bei benutzerdefinierten Attributen erfolgt durch Angabe des zu erfragenden Attributes als Index, z.B. *.type[<benutzerdefiniertes Attribut>]*.

Bei **Regeln** und **Funktionen** liefert das Attribut

- » mit Index 0 oder ohne Index:
den Typ des Rückgabewertes
- » mit Index ≥ 1 :
den Typ des jeweiligen Parameters

Existiert für den Index kein Parameter mehr, so erhalten Sie *void*.

Beim Ereignisobjekt **thisevent** kann der Typ des Ereignisses erfragt werden.

Definition

Datentyp	Zugriff	changed-Ereignis
<i>datatype</i>	get, set	ja
enum (thisevent)	get (function , rule , thisevent)	

C	COBOL
Bezeichner: AT_type	Bezeichner: AT-type
Datentyp: DT_type	Datentyp: DT-type
Datentyp: DT_enum (thisevent)	Datentyp: DT-enum (thisevent)

Klassifizierung

Allgemeines Attribut

Siehe auch

Kapitel „Funktionen (functions)“ und „Benannte Regeln (Unterprogramme)“ im Handbuch „Regelsprache“

Kapitel „Ereignisobjekt thisevent“ im Handbuch „Regelsprache“

2.392 .typescope

Mit diesem Attribut kann der Gültigkeitsbereich des Rückgabetyps von benutzerdefinierten Funktionen und Regeln gelesen werden.

Dieses Attribut ist für die Objektklassen **function** und **rule** verfügbar.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>anyvalue</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_typescope	Bezeichner: AT-typescope
Datentyp: DT_anyvalue	Datentyp: DT-anyvalue

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	function, rule

Siehe auch

Attribute *.indexscope[attribute]*, *.scope[attribute]*, *.typescope[integer]*

Kapitel „Gültigkeitsbereich zur besseren Typ-Prüfung“ im Handbuch „Regelsprache“

2.393 .typescope[integer]

Mit diesem Attribut kann der Gültigkeitsbereich von Parametern gelesen werden.

Der Zugriff auf den Gültigkeitsbereich von benannten Regeln, Ereignisregeln und Funktionen erfolgt über die Indizierung mit dem Datentyp *integer* im Bereich von *1count[.typescope]*.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>anyvalue</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_typescope	Bezeichner: AT-typescope
Datentyp: DT_anyvalue	Datentyp: DT-anyvalue

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	function, rule

Siehe auch

Attribute *.indexscope[attribute]*, *.scope[attribute]*, *.typescope*

Kapitel „Gültigkeitsbereich zur besseren Typ-Prüfung“ im Handbuch „Regelsprache“

2.394 .userdata

In diesem Objektattribut kann beliebige Information jeglicher Datentypen abgespeichert werden. Die Kontrolle darüber liegt bei der Anwendung. Der IDM stellt lediglich den dafür notwendigen Speicherplatz und die Verwaltung zur Verfügung.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>anyvalue</i>	get, set	ja

C

Bezeichner: AT_userdata

Datentyp: DT_anyvalue (definiert durch Parameterart)

COBOL

Bezeichner: AT-userdata

Datentyp: DT-anyvalue (definiert durch Parameterart)

Klassifizierung

Standardattribut

Anmerkung

Das Attribut *.userdata* kann dazu verwendet werden beliebige Werte zu speichern. Das bedeutet, es werden sowohl andere Objekte als auch Farben, Nummern oder Texte gespeichert. Dieses Attribut wird als *void*-Variable interpretiert, weil es immer den Datentyp des jeweiligen Inhalts übernimmt.

2.395 .userdata[integer]

Mit Hilfe dieses Attributes kann zu jedem Eintrag einer **listbox** oder eines **poptextes** eine Userdata mit beliebigen Daten belegt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>anyvalue</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_userdata	Bezeichner: AT-userdata
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	listbox, poptext

2.396 .userdata[index]

Mit Hilfe dieses Attributes kann zu jedem Feld [row, column] des **tablefields** eine Userdata mit beliebigen Daten belegt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>anyvalue</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_userdata	Bezeichner: AT-userdata
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

2.397 .username

In diesem Attribut kann der Benutzername definiert werden, das zum Starten der Anwendungsseite beim RSH- oder SSH-Protokoll verwendet wird.

Das Attribut wird ausgewertet, wenn beim Kommando im `.exec`-Attribut kein Benutzername angegeben ist. Das Attribut stellt eine Alternative dar, wenn es Probleme bereitet, das Kommando im `.exec`-Attribut korrekt aufzubauen.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>string</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_username	Bezeichner: AT-username
Datentyp: DT_string	Datentyp: DT-string

Vererbung
ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	application

Verfügbarkeit

Ab IDM-Version A.06.02.g

Siehe auch

Attribut `.password`

2.398 .userplaced

Mit dem Attribut *.userplaced* kann das Fenster beim Öffnen willkürlich positioniert werden. Es definiert, ob das Fenster beim Öffnen an der definierten Position erscheinen soll oder ob der Benutzer dieses Fenster interaktiv positionieren muss.

Definition

<i>Datentyp</i> <i>boolean</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
<i>C</i> Bezeichner: AT_userplaced Datentyp: DT_boolean		<i>COBOL</i> Bezeichner: AT-userplaced Datentyp: DT-boolean
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> window	

2.399 .uuid

Mit Hilfe dieses Attributes wird einem **control**-Objekt eine eindeutige ID gegeben. Diese ID wird mit Hilfe des Programms **guidgen.exe** generiert.

Wird das **control**-Objekt als OLE-Client benutzt, kann man in diesem Attribut die UUID des Servers hinterlegen. Dieses Attribut ersetzt dann das Attribut *.name*, in dem normalerweise der Servername hinterlegt wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_uuid	Bezeichner: AT-uuid
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	control

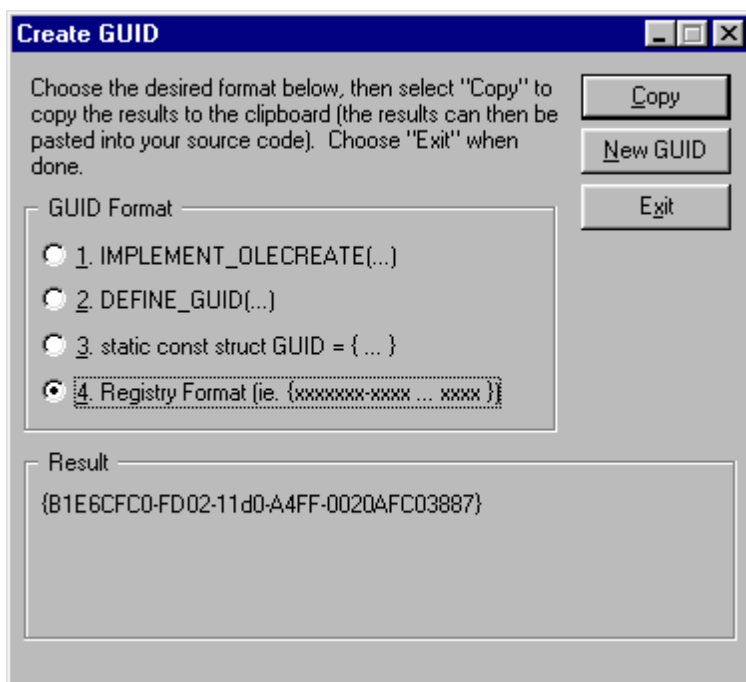


Abbildung 5: Generierung der GUID

2.400 .value

Dieses Attribut erfragt oder setzt den aktuellen Wert des Objekts.

Definition

<i>Datentyp</i> <i>string, object (text)</i> Datentyp des aktuellen Inhalts (variable)	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja nein (doccursor)
<i>C</i> Bezeichner: AT_value Datentyp: DT_string, DT_text Datentyp: Datentyp des aktuellen Inhalts (variable)		<i>COBOL</i> Bezeichner: AT-value Datentyp: DT-string, DT-text Datentyp: Datentyp des aktuellen Inhalts (variable)
<i>Vererbung</i> ja (datetime, filereq) nein (doccursor, variable)		
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> datetime, doccursor, filereq, variable	

2.400.1 datetime

Dieses Attribut legt den Wert des **datetime**-Objekts fest.

Beim Schreibzugriff („set“) können Werte vom Datentyp *string* und **text**-Ressourcen angegeben werden. **text**-Ressourcen werden automatisch zu *string* gewandelt.

Lesezugriff („get“) liefert immer einen Wert vom Datentyp *string*.

Wertebereich

""

Das aktuelle Datum bzw. die aktuelle Uhrzeit werden angezeigt.

<value_string>

Setzt den enthaltenen Datums- oder Zeitwert.

Syntax und Auswertung des Wertstrings

Der Wert wird entsprechend der „Definition der Zeitabstände“ für das timer-Objekt ausgewertet, wobei folgende absolute Zeitdefinition erlaubt ist:

```
<Value>      ::= {<Datevalue>}{ <TimeValue>}  
<DateValue>  ::= {<day>}_ {<month>}_ {<year>}  
<TimeValue>  ::= {<hour>}:{<minute>}:{<second>}}
```

Abweichend von der Definition der Zeitabstände des **timers** ist es also erlaubt, nur ein Datum anzugeben.

Das **datetime** verwendet den gregorianischen Kalender. Der Wertebereich geht von 01.01.1601 00:00:00 bis 31.12.9999 23:59:59. Zweistellige Jahreszahlen werden auf den Zeitraum 1970 bis 2069 abgebildet.

Wenn der Anwender im **datetime** einen Wert auswählt, einstellt oder eingibt, dann enthält *.value* diesen Wert im vollständigen Format *dd.MM.yyyy HH:mm:ss* (siehe Attribut *.format*). Bei einer Änderung tritt ein *select*-Ereignis auf.

Nicht angegebene Teile werden in der Anzeige durch die entsprechenden Werte des aktuellen Systemdatums bzw. der aktuellen Systemzeit ergänzt. Zum Beispiel wird für ein fehlendes Jahr das aktuelle Jahr eingesetzt.

Bei einem syntaktisch falschen Wertstring wird das aktuelle Datum bzw. die aktuelle Uhrzeit angezeigt (wie bei "").

Beispiele

Das aktuelle Datum sei „25.06.2014“ und die aktuelle Uhrzeit „11:53:07“.

Wertstring	Angezeigter Wert
12.1.	„12.01.2014 11:53:07“ Jahr und Zeit werden aus den aktuellen Werten ergänzt.
12.1	„25.06.2014 11:53:07“ Wertstring ist syntaktisch falsch. Aktuelles Datum und aktuelle Zeit werden angezeigt.

Ist der Wertstring zwar syntaktisch korrekt, stellt aber kein gültiges Datum oder keine gültige Zeit dar (zum Beispiel „29.2.2013“ oder „33.33.33“), dann ist undefiniert, was im **datetime** angezeigt wird. Wenn das System einen Fehler liefert, wird das aktuelle Datum bzw. die aktuelle Uhrzeit angezeigt. Ansonsten wird angezeigt, was das System aus dem Wertstring macht.

Wenn *.value* = "" oder fehlerhaft ist und das Attribut *.allowundefined* den Wert *true* hat, dann wird der Wert als unbestimmt (ausgeschaltete Checkbox) angezeigt.

Die Ergänzungen und Korrekturen, die vom Objekt für die Anzeige des Werts durchgeführt werden, ändern den Attributwert nicht. Das heißt, *.value* behält den gesetzten Wert so lange, bis der Anwender im **datetime** einen Wert auswählt, einstellt oder eingibt.

Das aktuelle Datum und die aktuelle Uhrzeit werden beim Sichtbarmachen berechnet. Außerdem werden sie berechnet, wenn *.value* im sichtbaren Zustand auf einen neuen Wert gesetzt wird den das Objekt für die Anzeige ergänzt oder korrigiert (zum Beispiel ""). Auch dabei gilt, dass der Attributwert in der Regel nicht an den angezeigten Wert angepasst wird. Ein wiederholtes Setzen desselben Werts (auch von "") stellt deshalb keine Wertänderung dar, sodass in diesem Fall auch keine Aktualisierung des angezeigten Werts erfolgt.

2.400.2 doccursor (XML-Cursor)

Das Attribut enthält den Wert des DOM-Knotens, auf den der **doccursor** zeigt. Es ist nur verfügbar, wenn das Attribut *.nodetype* einen der folgenden Werte besitzt:

- » *nodetype_attribute*
- » *nodetype_cdata_section*
- » *nodetype_comment*
- » *nodetype_processing_instruction*
- » *nodetype_text*

Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.400.3 filereq (Datei- und Verzeichnisdialoge)

Dieses Attribut enthält nach einer erfolgreichen Selektion (Rückgabewert von **querybox()** war *button_ok*) den vollständigen Pfad der ausgewählten Datei oder des ausgewählten Verzeichnisses.

Es ist zu beachten, dass dieser Pfad systemabhängige Trennzeichen enthält.

Wenn beim **filereq**-Objekt die Mehrfachselektion eingeschaltet ist (*.multisel = true*), dann ist die Liste der angewählten Dateien im indizierten Attribut *.value[integer]* enthalten.

Außerdem wird mit dem Attribut *.value* der Wert für die initiale Auswahl in Datei- und Verzeichnisdialogen mit *.startsel = true* definiert.

Siehe auch

Attribute *.multisel*, *.startsel*, *.value[integer]*

2.400.4 Globale Variablen

Erfragt den aktuellen Wert einer globalen Variablen oder weist ihr einen neuen Wert zu.

Bei einer Zuweisung muss der Datentyp des Wertes oder Ausdrucks dem Datentyp der Variablen entsprechen.

2.401 .value[integer]

Dieses Attribut des Dateiauswahldialogs (**filereq**) enthält, nach einer erfolgreichen Mehrfachselektion (Attribut `.multisel = true`, Rückgabewert von **querybox()** war `button_ok`), eine Liste mit den vollständigen Pfaden der vom Benutzer ausgewählten Dateien.

Die Anzahl der ausgewählten Dateien kann über einen Zugriff auf `.count[.value]` ermittelt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
string	get, set	ja

C	COBOL
Bezeichner: AT_value	Bezeichner: AT-value
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	filereq

Es ist zu beachten, dass diese Pfade systemabhängige Trennzeichen enthalten.

Im Falle der Einfachselektion (`.multisel = false`) ist die ausgewählte Datei im nicht indizierten Attribut [.value](#) zu finden.

Beispiel

```
rule void PrintValues(object Fr)
{
    variable integer I;
    for I:=1 to Fr.count[.value] do
        print Fr.value[I];
    endfor
}
```

Besonderheiten Microsoft Windows

Der Speicherplatz für die gesamte Liste der selektierten Pfade ist auf 64.000 Bytes beschränkt.

Siehe auch

Attribut [value](#)

2.402 .version[enum]

Das Attribut *.version* dient zur Vorgabe der Version des XML-Toolkits. Ein Wert von *0* bedeutet, dass der Standard verwendet wird.

Das Setzen eines neuen Wertes löscht den gespeicherten DOM-Baum und macht alle bestehenden XML-Cursor ungültig (*.mapped = false*).

Das Setzen des Attributes führt nicht automatisch zum Laden des XML-Toolkits. Wird das Attribut abgefragt, wird der gesetzte Wert zurückgeliefert. Soll dynamisch überprüft werden, ob das XML-Toolkit geladen werden kann, muss das Attribut *.real_version* abgefragt werden.

Als Index dient die *toolkit*-Enumeration.

Momentan wird aber nur der Wert *toolkit_windows* unterstützt, da nur unter Microsoft Windows die Laufzeitversion des MSXML-Controls vorgegeben werden kann. Hierzu muss die Majorversion mit *100* multipliziert und gegebenenfalls die Minorversion addiert werden. Wird *0* angegeben, wird das MSXML-Control versionsunabhängig geladen, was momentan die Version 3.0 ist.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> nein
<i>C</i> Bezeichner: AT_version Datentyp: DT_integer		<i>COBOL</i> Bezeichner: AT-version Datentyp: DT-integer
<i>Klassifizierung</i> Objektspezifisches Attribut	<i>Objekte</i> document	

Beispiel

Es soll MSXML 5.0 verwendet werden, damit XSD als Dokumenttyp verwendet werden kann:

```
this.version[toolkit_windows] := 500;
```

Anschließend wird überprüft, ob MSXML 5.0 geladen werden kann:

```
if (this.real_version[toolkit_windows] = 500) then  
    // OK  
endif
```

Hinweis

Sämtliche Versionen des MSXML-Controls unterscheiden sich in den unterstützten Eigenschaften. Die genauen Details sind der Dokumentation des MSXML-Controls zu entnehmen (MSDN-Library Kapitel „Win32- und COM- Entwicklung / Data Access and Storage / MSXML / MSXML SDK Overview / What's New in MSXML“). Hier sollen nur einige wichtige Unterschiede genannt werden:

MSXML 6.0

- » Einige Eigenschaften, die unsicher sind, wie DTD's und eingebettete Schemas, sind standardmäßig ausgeschaltet.
- » XDR Schema Unterstützung ist entfernt.
- » XML Digitale Signatur ist entfernt.

MSXML 5.0 für Microsoft Office Anwendungen

- » Unterstützung von XML Digitale Signaturen.
- » Unterstützung von eingebetteten XSD Schemas.

MSXML 4.0

- » Unterstützung von XML Schemas (XSD)
- » Unterstützung von Schema Object Model (SOM)
- » Ersetzungsmodus bei der Installation entfernt, alte Version des MSXML Controls werden nicht mehr überschrieben.
- » Versionsunabhängige ProgID entfernt.
- » Legacy Code entfernt:
- » Nicht konformes XSL ersetzt durch XSLT 1.0
- » Nicht konforme XSLPattern Sprache ersetzt durch XPath 1.0.

MSXML 3.0

- » Übereinstimmung mit XSLT 1.0 und XPath 1.0 Spezifikationen.
- » Unterstützung von Namespaces bei Abfragen mit XPath.

Siehe auch

Attribut *.real_version[enum]*

2.403 .version_string

Mit Hilfe dieses Setup-Objekt-Attributes kann der IDM-Versionsstring abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	<i>get</i>

C
Bezeichner: AT_version_string
Datentyp: DT_string

COBOL
Bezeichner: AT-version-string
Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

2.404 .vheight

Mit Hilfe dieses Attributes wird definiert, wie hoch das Objekt intern sein soll („virtuelle Höhe“).

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_vheight
Datentyp: DT_integer

COBOL
Bezeichner: AT-vheight
Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Kapitel „Scrollbarattribute“

2.405 .visible

Dieses Attribut legt fest, ob ein Objekt auf dem Bildschirm sichtbar ist oder nicht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_visible	Bezeichner: AT-visible
Datentyp: DT_boolean	Datentyp: DT-boolean

Klassifizierung
Standardattribut

Besonderheiten der Attribute .sensitive und .visible

Im Gegensatz zu den anderen Objektattributen erhalten die Attribute *.visible* und *.sensitive* erst im Zusammenhang mit der Objekthierarchie ihre endgültige Ausprägung, denn die entsprechenden Attribute des zugehörigen Vaterobjektes haben auf diese entscheidenden Einfluss. Konkret heißt das, dass ein Objekt nur dann sichtbar sein kann, wenn es selbst sichtbar **und** alle seine Väter sichtbar sind. Dasselbe gilt für die Selektierbarkeit.

Anmerkung

Die Anweisung *Window.visible := true* bewirkt Folgendes:

- » Wenn das Fenster sichtbar ist, tritt es nach Ausführung der Anweisung hervor, wird also zum vordersten Fenster.
- » Wenn das Fenster unsichtbar ist, erscheint es nach Ausführung der Anweisung auf dem Bildschirm.

Siehe auch

Attribut *.sensitive*

2.406 .vsb_arrows

Das Attribut *.vsb_arrows* definiert, ob die vertikale Scrollbar Pfeile an ihren Enden hat.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_vsb_arrows	Bezeichner: AT-vsb-arrows
Datentyp: DT_boolean	Datentyp: DT-boolean

Klassifizierung
Scrollbarattribut

Wertebereich

<i>true</i>	Scrollbar mit Pfeilen
<i>false</i>	Scrollbar ohne Pfeile

Verfügbarkeit

Das Attribut wird nur unter Motif und erst ab IDM-Version A.05.02.h unterstützt.

Siehe auch

Attribut *.hsb_arrows*

2.407 .vsb_linemotion

Dieses Attribut definiert, um wie viele Einheiten sich der Wert von *.yorigin* ändert, wenn der Benutzer zeilenweise vertikal scrollt. Einheiten entsprechen Pixeln, wenn *.sizeraster = false* ist, und Rasterereinheiten bei *.sizeraster = true*.

Definition

<i>Datentyp</i> <i>integer</i>	<i>Zugriff</i> get, set	<i>changed-Ereignis</i> ja
-----------------------------------	----------------------------	-------------------------------

C
Bezeichner: AT_vsb_linemotion
Datentyp: DT_integer

COBOL
Bezeichner: AT-vsblinemotion
Datentyp: DT-integer

Klassifizierung
Scrollbarattribut

Siehe auch

Kapitel „Scrollbarattribute“

2.408 .vsb_optional

Dieses Attribut definiert, dass die vertikale Scrollbar nur angezeigt wird, wenn sie tatsächlich benötigt wird.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C

Bezeichner: AT_vsb_optional

Datentyp: DT_boolean

COBOL

Bezeichner: AT-*vsb*-optional

Datentyp: DT-boolean

Klassifizierung

Scrollbarattribut

tablefield

Wird das Attribut auf *true* gesetzt, entscheidet der Dialog Manager, ob die vertikale Scrollbar aktuell benötigt wird oder nicht. Können z.B. alle Zeilen eines Tablefields komplett in dem zur Verfügung stehenden Raum dargestellt werden und ist die vertikale Scrollbar optional gesetzt, wird diese Scrollbar nicht angezeigt.

Siehe auch

Kapitel „Scrollbarattribute“

2.409 .vsb_pagemotion

Dieses Attribut definiert, um wie viele Einheiten sich der Wert von *.yorigin* ändert, wenn der Benutzer seitenweise vertikal scrollt. Einheiten entsprechen Pixeln, wenn *.sizeraster = false* ist, und Rasterereinheiten bei *.sizeraster = true*.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_vsb_pagemotion
Datentyp: DT_integer

COBOL
Bezeichner: AT-vs-b-pagemotion
Datentyp: DT-integer

Klassifizierung
Scrollbarattribut

Siehe auch

Kapitel „Scrollbarattribute“

2.410 .vsb_visible

Dieses Attribut definiert die Sichtbarkeit der vertikalen Scrollbar.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

C
Bezeichner: AT_vsb_visible
Datentyp: DT_boolean

COBOL
Bezeichner: AT-vsb-visible
Datentyp: DT-boolean

Klassifizierung
Scrollbarattribut

Siehe auch

Kapitel „Scrollbarattribute“

2.411 .vscreen_height

Mit Hilfe dieses setup-Objekt-Attributes kann die Bildschirmhöhe des virtuellen Desktops bzw. Arbeitsbereichs (in Pixel) abgefragt werden.

Dieses Attribut steht nur unter WINDOWS zur Verfügung, und ist für Multi-Monitor-Umgebungen gedacht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_vscreen_height	Bezeichner: AT-vscreen-height
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.412 .vscreen_width

Mit Hilfe dieses setup-Objekt-Attributes kann die Bildschirmbreite des virtuellen Desktops bzw. Arbeitsbereichs (in Pixel) abgefragt werden.

Dieses Attribut steht nur unter WINDOWS zur Verfügung, und ist für Multi-Monitor-Umgebungen gedacht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_vscreen_width	Bezeichner: AT-vscreen-width
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.413 .vscreen_x

Mit Hilfe dieses setup-Objekt-Attributes kann die X-Koordinate des Ursprungs des virtuellen Desktops bzw. Arbeitsbereichs (in Pixel) abgefragt werden.

Dieses Attribut steht nur unter WINDOWS zur Verfügung, und ist für Multi-Monitor-Umgebungen gedacht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	<i>get</i>

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_vscreen_x	Bezeichner: AT-vscreen-x
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.414 .vscreen_y

Mit Hilfe dieses setup-Objekt-Attributes kann die Y-Koordinate des Ursprungs des virtuellen Desktops bzw. Arbeitsbereichs (in Pixel) abgefragt werden.

Dieses Attribut steht nur unter WINDOWS zur Verfügung, und ist für Multi-Monitor-Umgebungen gedacht.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_vscreen_y	Bezeichner: AT-vscreen-y
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.415 .vwidth

Mit Hilfe dieses Attributes wird definiert, wie breit das Objekt intern sein soll („virtuelle Breite“).

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_vwidth
Datentyp: DT_integer

COBOL
Bezeichner: AT-vwidth
Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Kapitel „Scrollbarattribute“

2.416 .weeknumbers

Dieses Attribut bestimmt, ob im aufklappbaren Kalender die Nummern der Kalenderwochen angezeigt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_weeknumbers	Bezeichner: AT-weeknumbers
Datentyp: DT_boolean	Datentyp: DT-boolean

<i>Standardwert</i>	<i>Vererbung</i>
<i>false</i>	ja

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	datetime

Wertebereich

false

Im Kalender werden keine Kalenderwochen angezeigt

true

Im Kalender werden Kalenderwochen angezeigt.

Hinweis

Die erste Kalenderwoche eines Jahres ist die erste Woche, bei der mindestens 4 Tage im jeweiligen Jahr liegen.

2.417 .weight

Mit Hilfe dieses Attributes kann die Schriftstärke / Dicke / Gewicht eines Fonts definiert werden.

Anmerkung:

Es ist zu beachten, dass es vom gewählten Zeichensatz abhängt ob und in welcher Umfang die ausgewählten Modifikatoren angewendet werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>enum</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_weight	Bezeichner: AT-weight
Datentyp: DT_enum	Datentyp: DT-enum

Klassifizierung

Layoutattribut

Wertebereich

<i>face_default</i>	Normale, unveränderte Zeichendarstellung.
<i>face_light</i>	Leichgewichtige Zeichendarstellung.
<i>face_normal</i>	Normale, unveränderte Zeichendarstellung.
<i>face_medium</i>	Schriftstärke zwischen <i>face_normal</i> und <i>face_demibold</i> .
<i>face_demibold</i>	Schriftstärke zwischen <i>medium</i> und <i>bold</i> .
<i>face_bold</i>	Fette Zeichendarstellung.
<i>face_black</i>	Extra fette Zeichendarstellung.

Verfügbarkeit

Ab IDM-Version A.06.03.a

2.418 .width

Dieses Attribut gibt die aktuelle Breite des Objekts an.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i> (kann 0 sein)	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_width	Bezeichner: AT-width
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

.width definiert die Breite des gesamten **notebook**-Objekts mit all seinen Elementen. Wird die Breite zu klein definiert, wird ein vom Fenstersystem abhängiger Minimalwert genommen. Die Breite darf nicht 0 sein, da die benötigte Breite nicht berechnet werden kann.

Beim **tablefield** kann das Attribut auch den Wert 0 haben. Dies bedeutet, dass der Dialog Manager die entsprechende Größe so berechnen soll, dass alle Elemente in der entsprechenden Richtung dargestellt werden können, ohne dass eine Scrollbar notwendig ist. Wird z.B. die Breite des **tablefields** auf 0 gesetzt, wird es so breit, dass alle Spalten vollständig in dem Objekt dargestellt werden können.

Bei der **toolbar** definiert *.height* (ohne Index) den Standardwert für die nicht mit *.width[class]* gesetzten Werte.

Siehe auch

Attribute *.height*, *.real_width*, *.xauto*, *.xleft*, *.xright*

2.419 .width[class]

Das Attribut gibt die Breite als angedockte Toolbar bzw. als Toolwindow an.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_width	Bezeichner: AT-width
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	toolbar

Indexbereich

<i>toolbar</i>
Breite, wenn die Toolbar angedockt ist
<i>window</i>
Breite, wenn die Toolbar abgedockt ist (Toolwindow)

Ohne einen Index gibt das Attribut den Standardwert für beide Andockzustände zurück oder legt ihn fest.

Siehe auch

Attribut *.height[class]*

2.420 .width[enum]

Mit diesem Attribut können Korrekturfaktoren für die Berechnung der Rasterbreite aus dem **font** definiert werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	nein

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_width	Bezeichner: AT-width
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>
Geometrieattribut

Indexbereich

scale_factor

Mit diesem Indexwert definiert das Attribut einen Prozentsatz, der als Skalierungsfaktor mit dem der Basiswert multipliziert wird.

Der Basiswert ist die aus der Font-Metrik oder dem Referenzstring ermittelte Zeichenbreite des Zeichensatzes..

scale_offset

Mit diesem Indexwert definiert das Attribut einen Pixelwert, der als Konstante zum skalierten Basiswert addiert wird.

Siehe auch

Attribute *.height[enum]*, *.refstring*, *.real_xraster*, *.xraster*

Kapitel „Berechnung der Rastergröße aus einem Referenzfont“ im Handbuch „Ressourcenreferenz“

2.421 .window

Mit Hilfe dieses Attributes kann das Fenster erfragt werden, zu dem das Objekt gehört.

Definition

Datentyp
object

Zugriff
get

C
Bezeichner: AT_window
Datentyp: DT_object

COBOL
Bezeichner: AT-window
Datentyp: DT-object

Klassifizierung
Hierarchieattribut

2.422 .winsys

Mit Hilfe dieses Setup-Objekt-Attributes kann die Fenstersystemart abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>enum</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_winsys	Bezeichner: AT-winsys
Datentyp: DT_enum	Datentyp: DT-enum

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Wertebereich

winsys_none

Das Fenstersystem kann nicht identifiziert werden.

winsys_windows

Das Fenstersystem ist eine Version von MICROSOFT WINDOWS.

winsys_x11

Das Fenstersystem ist eine X11-kompatible Version des X WINDOW SYSTEM (X Windows).

2.423 .winsys_string

Mit Hilfe dieses Setup-Objekt-Attributes kann das Fenstersystem abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>string</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_winsys_string	Bezeichner: AT-winsys-string
Datentyp: DT_string	Datentyp: DT-string

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Attribut *.winsys_version*

2.424 .winsys_version

Mit Hilfe dieses Setup-Objekt-Attributes kann die Fenstersystem-Version abgefragt werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_winsys_version	Bezeichner: AT-winsys-version
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Siehe auch

Attribut *.winsys_string*

2.425 .wrap

Bei der **layoutbox** kann mit diesem Attribut der Umbruch ein- und ausgeschaltet werden. Umbruch heißt, dass bei einer Größenänderung oder Änderung des *.visible*-Attributs der **layoutbox** die Kinder gegebenenfalls neu sortiert werden, damit die Kinder alle sichtbar, bzw. erreichbar bleiben.

Bei der **spinbox** kann über dieses Attribut gesteuert werden, ob ein zirkuläres Spinning erfolgen soll (*.wrap = true*), oder ob beim Erreichen der Unter- bzw. der Obergrenze das Spinning abgebrochen werden soll (*.wrap = false*). Der interne Default für dieses Attribut ist *true*.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>boolean</i>	get, set	ja
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_wrap		Bezeichner: AT-wrap
Datentyp: DT_boolean		Datentyp: DT-boolean
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	layoutbox, spinbox	

layoutbox

Wenn der Umbruch ausgeschaltet wird, werden keine Änderungen mehr durchgeführt (z.B. Änderungen an den Attributen der **layoutbox** *direction*, *.ymargin*, *.xmargin*... oder deren Kindern).

Falls der Umbruch von Anfang an ausgeschaltet ist, werden die Objekte von der **layoutbox** nicht angeordnet.

2.426 .x

Mit Hilfe dieses Attributes kann beim Ereignisobjekt ***thisevent*** die Mauskoordinate X erfragt werden (gültig nur beim *select*-Ereignis bei Fenster und Groupbox; relativ zum Objekt, auf das sich das Ereignis bezieht).

Definition

Datentyp
integer

Zugriff
get

C
Bezeichner: AT_x
Datentyp: DT_integer

COBOL
Bezeichner: AT-x
Datentyp: DT-integer

Klassifizierung
Objektspezifisches Attribut

Objekte
thisevent

Siehe auch

Attribut .y

Kapitel „Ereignisobjekt thisevent“ im Handbuch „Regelsprache“

2.427 .xalignment[index]

Dieses Attribut des **tablefields** definiert für die Zelle in Zeile I und Spalte J die horizontale Ausrichtung des Inhalts (links, zentriert, rechts) innerhalb der Zelle.

Der Wert in *.xalignment[0,0]* wird für alle Zellen genommen, für die keine horizontale Ausrichtung angegeben ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_xalignment	Bezeichner: AT-xalignment
Datentyp: DT_integer	Datentyp: DT-integer

Standardwert
1

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Wertebereich

<i>-1</i>	rechtsbündig
<i>0</i>	horizontal zentriert
<i>1</i>	linksbündig
Standardwert, wenn nicht durch <i>.xalignment[0,0]</i> überschrieben	

Hinweis

Zur Benutzung dieses Attributs muss *.options[opt_new_align] = true* sein.

2.428 .xauto

Mit diesem Attribut wird definiert, ob der IDM die Objektbreite, den Abstand des Objekts vom linken Rand oder den Abstand vom rechten Rand des Vaterobjekts automatisch bestimmen soll.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_xauto	Bezeichner: AT-xauto
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Wertebereich

- 1
Der Abstand von linken Rand des Vaterobjekts wird vom IDM berechnet.
Der Abstand vom rechten Rand und die Objektbreite (*.xright* und *.width*) sind vom Entwickler zu definieren, wobei der IDM bei einigen Objekten auch die Breite automatisch aus dem Inhalt berechnen kann.
- 0
Die Objektbreite wird vom IDM berechnet. Die tatsächliche Breite kann mit dem Attribut *.real_width* abgefragt werden.
Die Abstände vom linken und rechten Rand des Vaterobjekts (*.xleft* und *.xright*) sind vom Entwickler zu definieren.
- 1
Der Abstand von rechten Rand des Vaterobjekts wird vom IDM berechnet.
Der Abstand vom linken Rand und die Objektbreite (*.xleft* und *.width*) sind vom Entwickler zu definieren, wobei der IDM bei einigen Objekten auch die Breite automatisch aus dem Inhalt berechnen kann.

Siehe auch

Attribute *.width*, *.xleft*, *.xright*

Kapitel „Geometrieattribute“

2.429 .xdpi

Mit Hilfe dieses Setup-Objekt-Attributes können die DPI („dots per inch“) in X-Richtung abgefragt werden.

Es wird bei Multiscreendialogen (nur Motif) der Wert für den Default-Screen zurückgegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_xdpi	Bezeichner: AT-xdpi
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

2.430 .xdpi[integer]

Mit Hilfe dieses Setup-Objekt-Attributes können die DPI („dots per inch“) in X-Richtung abgefragt werden.

Das indizierte Attribut ist nur bei Multiscreendialogen (nur Motif) verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_xdpi	Bezeichner: AT-xdpi
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung. Es ist zu beachten, dass mit Screen-Index nicht die Screen-Nummer gemeint ist, die z.B. mit dem Programm **xdpyinfo** erhalten wird.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

2.431 .xleft

Dieses Attribut gibt den Abstand zur linken Seite des übergeordneten Objektes an.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_xleft	Bezeichner: AT-xleft
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Attribute *.width*, *.xauto*, *.xright*

Kapitel „Geometrieattribute“

2.432 .xmargin

Dieses Attribut bestimmt den Abstand der Kinder (**layoutbox**) bzw. des Inhalts (**edittext**, **poptext**, **image**) vom linken und rechten Rand des Objekts.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_xmargin	Bezeichner: AT-xmargin
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	edittext, layoutbox, poptext, image

layoutbox

Wertebereich 0...65536

Standardwert 0

Mit diesem Attribut kann der linke und rechte Abstand zwischen der **layoutbox** und den Kindern angegeben werden.

image

Das Attribut gibt den horizontalen Abstand vom Rand (Umrandung) zum eigentlichen Inhalt eines Image-Objekts in Pixel an. Änderungen an den xmargin-Werten wirken sich immer auf den Abstand zum linken Rand und zum rechten Rand aus.

Siehe auch Kapitel „HighDPI UnterstützungSupport“ im Handbuch „Programmiertechniken“

Ab IDM-Version A.06.03.a

edittext

Das Attribut gibt den Abstand zwischen Rand (Umrandung) und dem Text in Pixel an. Änderungen an den xmargin-Werten wirken sich immer auf den Abstand zum linken Rand und zum rechten Rand aus. Wenn zusätzlich *.width = 0* und *.xauto <> 0* ist wirkt sich eine Änderung des Attributs auch auf die Breite des Objekts aus.

Zu beachten ist, dass die Default-Werte dieses Attributs bei den WSIs (also z.B. Motif, Windows) unterschiedlich sind. Der Wertebereich ist -127 ... +127. Defaultwert für Motif ist 5, für Windows 1.

Unter Microsoft Windows muss zusätzlich am Dialog das Attribut *.options[opt_et_margin] = false* gesetzt sein (Defaultwert), damit das Attribut *.xmargin* ausgewertet wird. Außerdem kann es unter Microsoft Windows (Patchlevel-Abhängig) bei einem einzeiligen Edittext mit überlangem Text sein, dass der Abstand zum rechten Rand nicht beachtet wird.

poptext

Microsoft Windows

Der **poptext** beachtet bei `.style = edittext` bzw. `.style = listbox` das Attribut `.xmargin`. Das Verhalten entspricht dem Verhalten des **edittext**-Objekts (siehe oben).

Motif

Das **poptext**-Objekt unterstützt unter Motif 2.x ab IDM-Version A.05.02.d die Attribute `.xmargin` und `.ymargin` um den horizontalen und vertikalen Abstand zwischen Text und äußerem Rahmen zu beeinflussen.

Da der **poptext** unter Motif ein zusammengesetztes Objekt ist, beeinflussen die Attribute den Abstand zwischen Text und Textfeld-Rahmen und damit nur indirekt den Abstand zwischen Textfeld und äußerem Rahmen. Das `.ymargin`-Attribut beeinflusst die Höhe des **poptexts**, wenn keine Höhe vorgegeben wurde (`.height = 0` und `.yauto <> 0`). Zu große Werte können bei vorgegebener Höhe dazu führen dass der Text nicht mehr sichtbar ist. Dagegen wirkt das `.xmargin`-Attribut auch wenn eine Breite angegeben wurde und verändert damit den darstellbaren Textbereich und Rahmenabstand. Bei negativen Attributwerten werden die Standardwerte des Systems verwendet.

Besonderheiten

- » Von einer Verwendung von `.xmargin` wird abgeraten, da Motif im Rahmenbereich seinen Cursor zeichnet wenn er links bzw. rechts vom Text steht.
- » Der Textfeldrahmen ist unter Motif normalerweise nur für die styles **edittext** und **listbox** sichtbar.
- » Der Wert -2 für `.xmargin` und `.ymargin` ist ein Kompatibilitätsmodus zur Vorgängerversion, die bei `.height = 0` eine leicht andere Höhenberechnung und Textpositionierung hatte.
- » Der Standardwert ist 5 und unterscheidet sich damit von den Windows-Plattformen.

2.433 .xml

Über dieses Attribut kann man die String-Darstellung des ***XML-Dokuments*** oder ***XML-Cursors*** erfragen. Wenn beim XML-Dokument ein neuer Wert gesetzt wird, wird der gespeicherte DOM-Baum gelöscht und ein neuer DOM-Baum aus dem gesetzten Wert aufgebaut. Alle bestehenden XML-Cursor werden ungültig. Bei einem ungültigen XML-Cursor besitzt das Attribut *.mapped* den Wert *false*.

Definition

Datentyp
string, object [text]

Zugriff
get (***doccursor***)
get, set (***document***)

C
Bezeichner: AT_xml
Datentyp: DT_string, DT_text

COBOL
Bezeichner: AT-xml
Datentyp: DT-string, DT-text

Vererbung
nein

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	doccursor, document

Das Attribut ist für das XML-Dokument und den XML-Cursor verfügbar. Es ist zu beachten, dass ein XML-Cursor, dessen Attribut *.mapped* den Wert *false* besitzt, automatisch auf die Wurzel des DOM-Dokuments positioniert wird.

2.434 .xorigin

Dieses Attribut gibt den Wert an, um wie viel der eigentliche Inhalt des Objektes in X-Richtung verschoben ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_xorigin
Datentyp: DT_integer

COBOL
Bezeichner: AT-xorigin
Datentyp: DT-integer

Klassifizierung
Geometrieattribut

2.435 .xraster

Dieses Attribut bildet die zugrundeliegende Einheit des Rasters in X-Richtung.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_xraster	Bezeichner: AT-xraster
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Anmerkungen

- » Die Attribute *.reffont*, *.xraster* und *.yraster* haben vor allem Bedeutung, wenn die Anwendung so realisiert werden soll, dass sie auf möglichst vielen unterschiedlichen Hardware-Umgebungen ablauffähig ist. Mit Hilfe dieser Attribute wird eine zugrundeliegende Objektgrößeneinheit definiert, die unabhängig von den bisher verwendeten Pixeleinheiten ist. Zur neuen Einheit wird die Größe eines Buchstabens des ausgewählten Referenzfonts oder die Angaben, die der Benutzer direkt ohne Angabe eines Zeichensatzes gemacht hat. Alle Dimensionen oder Positionen beziehen sich dann auf diese Einheit.

Bezeichner	Datentyp	Bedeutung
<i>.reffont</i>	<i>object (font)</i>	Name des Zeichensatzes, auf den sich diese Größe beziehen soll.
<i>.xraster</i>	<i>integer</i>	Zugrundeliegende Einheit in X-Richtung.
<i>.yraster</i>	<i>integer</i>	Zugrundeliegende Einheit in Y-Richtung.

- » Wird ein Referenzfont angegeben, so rechnet der IDM die Werte für *.xraster* und *.yraster* automatisch aus, d.h. die Angabe von *.reffont* überschreibt die Angaben von *.xraster* und *.yraster*.

Beispiel

```
.xraster      8;  
.yraster      16;  
...  
.posraster    true;  
.xleft        10;  
.ytop         4;
```

Dieses so definierte Objekt hat in Pixelkoordinaten die Position 80, 64, da die Positionsangabe zur Basis 8 bzw. 16 erfolgte.

Siehe auch

Attribute *.reffont*, *.yraster*

2.436 .xright

Dieses Attribut gibt den Abstand zur rechten Seite des übergeordneten Objektes an.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_xright	Bezeichner: AT-xright
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Attribute *.width*, *.xauto*, *.xleft*

Kapitel „Geometrieattribute“

2.437 .xspacing

Mit diesem Attribut kann der **horizontale** Abstand zwischen den Kindern der **layoutbox** angegeben werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_xspacing	Bezeichner: AT-xspacing
Datentyp: DT_integer	Datentyp: DT-integer

<i>Wertebereich</i>	<i>Standardwert</i>
0...65536	0

<i>Klassifizierung</i>	<i>Objekte</i>
Geometrieattribut	layoutbox

Siehe auch

Attribut *.yspacing*

2.438 .y

Mit Hilfe dieses Attributes kann beim Ereignisobjekt ***thisevent*** die Mauskoordinate Y erfragt werden (gültig nur beim *select*-Ereignis bei Fenster und Groupbox; relativ zum Objekt, auf das sich das Ereignis bezieht).

Definition

Datentyp
integer

Zugriff
get

C
Bezeichner: AT_y
Datentyp: DT_integer

COBOL
Bezeichner: AT-y
Datentyp: DT-integer

Klassifizierung
Objektspezifisches Attribut

Objekte
thisevent

Siehe auch

Attribut .x

Kapitel „Ereignisobjekt thisevent“ im Handbuch „Regelsprache“

2.439 .yalignment[index]

Dieses Attribut des **tablefields** definiert für die Zelle in Zeile I und Spalte J die vertikale Ausrichtung des Inhalts (oben, zentriert, unten) innerhalb der Zelle.

Der Wert in *.yalignment[0,0]* wird für alle Zellen genommen, für die keine vertikale Ausrichtung angegeben ist.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_yalignment	Bezeichner: AT-yalignment
Datentyp: DT_integer	Datentyp: DT-integer

Standardwert
1

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	tablefield

Wertebereich

- 1*
unten bündig
 - 0*
vertikal zentriert
 - 1*
oben bündig
- Standardwert, wenn nicht durch *.yalignment[0,0]* überschrieben

Hinweis

Zur Benutzung dieses Attributs muss *.options[opt_new_align] = true* sein.

2.440 .yauto

Mit diesem Attribut wird definiert, ob der IDM die Objekthöhe, den Abstand des Objekts vom oberen Rand oder den Abstand vom unteren Rand des Vaterobjekts automatisch bestimmen soll.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_yauto	Bezeichner: AT-yauto
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Wertebereich

-1

Der Abstand vom oberen Rand des Vaterobjekts wird vom IDM berechnet.
Der Abstand vom unteren Rand und die Objekthöhe (*.ybottom* und *.height*) sind vom Entwickler zu definieren, wobei der IDM bei einigen Objekten auch die Höhe automatisch aus dem Inhalt berechnen kann.

0

Die Objekthöhe wird vom IDM berechnet. Die tatsächliche Höhe kann mit dem Attribut *.real_height* abgefragt werden.
Die Abstände vom oberen und unteren Rand des Vaterobjekts (*.ytop* und *.ybottom*) sind vom Entwickler zu definieren.

1

Der Abstand vom unteren Rand des Vaterobjekts wird vom IDM berechnet.
Der Abstand vom oberen Rand und die Objekthöhe (*.ytop* und *.height*) sind vom Entwickler zu definieren, wobei der IDM bei einigen Objekten auch die Höhe automatisch aus dem Inhalt berechnen kann.

Siehe auch

Attribute *.height*, *.ybottom*, *.ytop*

Kapitel „Geometrieattribute“

2.441 .ybottom

Dieses Attribut gibt den Abstand zur unteren Seite des übergeordneten Objektes an.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_ybottom
Datentyp: DT_integer

COBOL
Bezeichner: AT-ybottom
Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Attribute *.height*, *.yauto*, *.ytop*

Kapitel „Geometrieattribute“

2.442 .ydpi

Mit Hilfe dieses Setup-Objekt-Attributes können die DPI („dots per inch“) in Y-Richtung abgefragt werden.

Es wird bei Multiscreendialogen (nur Motif) der Wert für den Default-Screen zurückgegeben.

Definition

<i>Datentyp</i>	<i>Zugriff</i>
<i>integer</i>	get

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_ydpi	Bezeichner: AT-ydpi
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	setup

2.443 .ydpi[integer]

Mit Hilfe dieses Setup-Objekt-Attributes können die DPI („dots per inch“) in Y-Richtung abgefragt werden.

Das indizierte Attribut ist nur bei Multiscreendialogen (nur Motif) verfügbar, wobei der Index im Bereich 1 ... *setup.screencount* liegen kann.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	
<i>integer</i>	get	
<i>C</i>		<i>COBOL</i>
Bezeichner: AT_ydpi		Bezeichner: AT-ydpi
Datentyp: DT_integer		Datentyp: DT-integer
<i>Klassifizierung</i>	<i>Objekte</i>	
Objektspezifisches Attribut	setup	

Anmerkung Motif:

Der IDM FÜR MOTIF besitzt Multi-Screen Unterstützung. Es ist zu beachten, dass mit Screen-Index nicht die Screen-Nummer gemeint ist, die z.B. mit dem Programm **xdpyinfo** erhalten wird.

Siehe auch Kapitel „Multiscreen Ssupport untder Motif“ im Handbuch „Programmiertechniken“

Ein Beispieldialog ist bei der display-Ressource zu finden.

Anmerkung Windows:

Der IDM FÜR WINDOWS hat ab IDM Version **A.06.03.a** Multi-Monitor Unterstützung.

Siehe auch: Kapitel „Multi-Mmonitor Ssupport untder Windows“ im Handbuch „Programmiertechniken“.

2.444 .ymargin

Dieses Attribut bestimmt den Abstand der Kinder (**layoutbox**) bzw. des Inhalts (**edittext**, **poptext**, **image**) vom oberen und unteren Rand des Objekts.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_ymargin	Bezeichner: AT-ymargin
Datentyp: DT_integer	Datentyp: DT-integer

<i>Klassifizierung</i>	<i>Objekte</i>
Objektspezifisches Attribut	edittext, layoutbox, poptext, image

layoutbox

Wertebereich 0...65536

Standardwert 0

Mit diesem Attribut kann der obere und untere Abstand zwischen der **layoutbox** und den Kindern angegeben werden.

image

Das Attribut gibt den vertikalen Abstand vom Rand (Umrandung) zum eigentlichen Inhalt eines Image-Objekts in Pixel an. Änderungen an den ymargin-Werten wirken sich immer auf den Abstand zum oberen Rand und zum unteren Rand aus.

Siehe auch Kapitel „HighDPI UnterstützungSupport“ im Handbuch „Programmiertechniken“

Ab IDM-Version A.06.03.a

edittext (nur Motif)

Dieses Attribut gibt den Abstand zwischen Rand (Umrandung) und dem Text in Pixel an. Änderungen an den .ymargin-Werten wirken sich immer auf den Abstand zum oberen Rand und zum unteren Rand aus. Wenn zusätzlich die Attribute .height = 0 und .yauto <> 0 sind, wirkt sich eine Änderung des Attributs auch auf die Höhe des Objekts aus.

Der Wertebereich ist -127 ... +127. Defaultwert für Motif ist 5.

Das Attribut wird vom **edittext** unter MICROSOFT WINDOWS nicht unterstützt.

poptext (nur Motif)

Das **poptext**-Objekt unterstützt unter Motif 2.x ab IDM-Version A.05.02.d die Attribute .xmargin und .ymargin um den horizontalen und vertikalen Abstand zwischen Text und äußerem Rahmen zu beeinflussen.

Da der **poptext** unter Motif ein zusammengesetztes Objekt ist, beeinflussen die Attribute den Abstand zwischen Text und Textfeld-Rahmen und damit nur indirekt den Abstand zwischen Textfeld und äußerem Rahmen. Das `.ymargin`-Attribut beeinflusst die Höhe des **poptexts**, wenn keine Höhe vorgegeben wurde (`.height = 0` und `.yauto <> 0`). Zu große Werte können bei vorgegebener Höhe dazu führen dass der Text nicht mehr sichtbar ist. Dagegen wirkt das `.xmargin`-Attribut auch wenn eine Breite angegeben wurde und verändert damit den darstellbaren Textbereich und Rahmenabstand. Bei negativen Attributwerten werden die Standardwerte des Systems verwendet.

Besonderheiten (Motif)

- » Der Textfeldrahmen ist unter Motif normalerweise nur für die styles *edittext* und *listbox* sichtbar.
- » Der Wert -2 für `.xmargin` und `.ymargin` ist ein Kompatibilitätsmodus zur Vorgängerversion, die bei `.height = 0` eine leicht andere Höhenberechnung und Textpositionierung hatte.
- » Der Standardwert ist 5 und unterscheidet sich damit von den Windows-Plattformen.

2.445 .yorigin

Über dieses Attribut wird der Wert angegeben, um wie viel der Inhalt in Y-Richtung verschoben wurde.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

C
Bezeichner: AT_yorigin
Datentyp: DT_integer

COBOL
Bezeichner: AT-yorigin
Datentyp: DT-integer

Klassifizierung
Geometrieattribut

2.446 .yraster

Dieses Attribut bildet die zugrundeliegende Einheit des Rasters in Y-Richtung.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_yraster	Bezeichner: AT-yraster
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Anmerkungen

- » Die Attribute *.reffont*, *.xraster* und *.yraster* haben vor allem Bedeutung, wenn die Anwendung so realisiert werden soll, dass sie auf möglichst vielen unterschiedlichen Hardware-Umgebungen ablauffähig ist. Mit Hilfe dieser Attribute wird eine zugrundeliegende Objektgrößeneinheit definiert, die unabhängig von den bisher verwendeten Pixeleinheiten ist. Zur neuen Einheit wird die Größe eines Buchstabens des ausgewählten Referenzfonts oder die Angaben, die der Benutzer direkt ohne Angabe eines Zeichensatzes gemacht hat. Alle Dimensionen oder Positionen beziehen sich dann auf diese Einheit.

Bezeichner	Datentyp	Bedeutung
<i>.reffont</i>	<i>object (font)</i>	Name des Zeichensatzes, auf den sich diese Größe beziehen soll.
<i>.xraster</i>	<i>integer</i>	Zugrundeliegende Einheit in X-Richtung.
<i>.yraster</i>	<i>integer</i>	Zugrundeliegende Einheit in Y-Richtung.

- » Wird ein Referenzfont angegeben, so rechnet der IDM die Werte für *.xraster* und *.yraster* automatisch aus, d.h. die Angabe von *.reffont* überschreibt die Angaben von *.xraster* und *.yraster*.

Beispiel

```
.xraster 8;  
.yraster 16;  
...  
.posraster true;  
.xleft 10;  
.ytop 4;
```

Dieses so definierte Objekt hat in Pixelkoordinaten die Position 80, 64, da die Positionsangabe zur Basis 8 bzw. 16 erfolgte.

Siehe auch

Attribute *.reffont*, *.xraster*

2.447 .yspacing

Mit diesem Attribut kann der **vertikale** Abstand zwischen den Kindern der **layoutbox** angegeben werden.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_yspacing	Bezeichner: AT-yspacing
Datentyp: DT_integer	Datentyp: DT-integer

<i>Wertebereich</i>	<i>Standardwert</i>
0...65536	0

<i>Klassifizierung</i>	<i>Objekte</i>
Geometrieattribut	layoutbox

Siehe auch

Attribut *.xspacing*

2.448 .ytop

Dieses Attribut gibt den Abstand zur oberen Seite des übergeordneten Objektes an.

Definition

<i>Datentyp</i>	<i>Zugriff</i>	<i>changed-Ereignis</i>
<i>integer</i>	get, set	ja

<i>C</i>	<i>COBOL</i>
Bezeichner: AT_ytop	Bezeichner: AT-ytop
Datentyp: DT_integer	Datentyp: DT-integer

Klassifizierung
Geometrieattribut

Siehe auch

Attribute *.height*, *.yauto*, *.ybottom*

Kapitel „Geometrieattribute“

Index

.

.scale 408

.width 545

.xmargin 559

.ymargin 573

A

.acc_label 25

.acc_text 27

accelerator 237

.accelerator 29

Accelerator

 Identifikator 29

.action 30

action_copy 30

action_cut 30

action_paste 30

active 31, 35

active[index] 34

active[integer] 33

activeitem 35

activeobject 31, 37

ActiveX-Control 340

alignment 38

allowundefined 40

Anwendung

 lokal 255

 Netz 255

Anwendungssymbol 216

application 41, 316

applikationsmodal 133

arrows 42

AT-acc-label 25

AT-acc-text 27

AT-accelerator 29

AT-action 30

AT-active 31, 33-34

AT-activeitem 35

AT-activeobject 37

AT-alignment 38

AT-allowundefined 40

AT-application 41

AT-arrows 42

AT-attribute 43-44

AT-autoalign 46

AT-autosize 47

AT-barwidth 49

AT-bgc 50-51, 239

AT-binding 48, 52

AT-bordercolor 53

AT-borderraster 54

AT-borderstyle 59

AT-borderwidth 62

AT-button 63

AT-bw 64

AT-calendaralignment 66

AT-canvasfunc 67

AT-certificatefile 68

AT-changedir 69

AT-child [70](#)
AT-childcount [71](#)
AT-class [72](#)
AT-closeable [73](#)
AT-codepage [74](#)
AT-colalignment [77](#)
AT-colcount [78](#)
AT-colfirst [79](#)
AT-colheader [80](#)
AT-colheadfgc [81](#)
AT-colheadfont [82](#)
AT-colheadshadow [83](#)
AT-colheadvisible [84](#)
AT-collinewidth [85](#)
AT-color [86](#)
AT-color-type [87-88](#)
AT-colorcount [89-90](#)
AT-colorname [91](#)
AT-colsizeable [92](#)
AT-coltile [93](#)
AT-colvisible [94](#)
AT-colwidth [95](#)
AT-configurable [97](#)
AT-connect [98](#)
AT-constant [101](#)
AT-content [102](#), [104-105](#)
AT-contentfunc [107](#)
AT-control [108](#)
AT-count [109](#)
AT-cursor [110](#)
AT-cursorname [111](#)
AT-curvalue [112](#)
AT-cut-pending [113](#)
AT-cut-pending-changed [114](#)
AT-data [115](#)
AT-dataget [116](#)
AT-dataindex [117](#)
AT-datamap [118](#)
AT-datamodel [119](#)
AT-dataoptions [120](#)
AT-dataset [122](#)
AT-datasetattr [124](#)
AT-datasetcount [125](#)
AT-datasettype [126](#)
AT-dataset [128](#)
AT-defbutton [129](#)
AT-deltavalue [130](#)
AT-depth [131](#)
AT-dialog [132](#)
AT-dialogbox [133](#)
AT-direction [135](#)
AT-directory [139](#)
AT-display [140](#)
AT-doccursor [141](#)
AT-dock-line [142](#)
AT-dock-offset [143](#)
AT-dockable [144](#)
AT-docking [146](#)
AT-document [147](#)
AT-editable [148](#), [150](#)
AT-editpos [151](#)
AT-edittext [152](#)
AT-edittype [153](#)
AT-endsel [154](#)

AT-env [155](#)
 AT-envvar [156](#)
 AT-errfile [157](#)
 AT-errorcode [158](#)
 AT-event [159](#)
 AT-event-code [160](#)
 AT-eventcount [161](#)
 AT-exec [162](#)
 AT-extension [165](#)
 AT-external [166-167](#)
 AT-extevent [168](#)
 AT-face [169](#)
 AT-fgc [170-171](#)
 AT-field [172](#)
 AT-fieldactive [173](#)
 AT-fieldfocus [174-175](#)
 AT-fieldfocusable [176](#)
 AT-fieldshadow [177](#)
 AT-filled [178](#)
 AT-firstchar [179](#)
 AT-firstchild [180](#)
 AT-firstmenu [181](#)
 AT-firstrecord [182](#)
 AT-firstsubcontrol [183](#)
 AT-firsttoolbar [184](#)
 AT-focus [185](#)
 AT-focus-on-click [188](#)
 AT-focusitem [190](#)
 AT-font [191-192](#)
 AT-fontname [193](#)
 AT-format [194, 197](#)
 AT-formatfunc [198](#)
 AT-function [199](#)
 AT-gradient [199](#)
 AT-grey [201](#)
 AT-groupbox [202](#)
 AT-height [203-204](#)
 AT-help [206](#)
 AT-helpmenu [207](#)
 AT-helppos [208](#)
 AT-hls [210](#)
 AT-hsb-arrows [211](#)
 AT-hsb-linemotion [212](#)
 AT-hsb-optional [213](#)
 AT-hsb-pagemotion [214](#)
 AT-hsb-visible [215](#)
 AT-icon [216](#)
 AT-iconic [218](#)
 AT-iconifyable [219](#)
 AT-idispatch [220](#)
 AT-ignorecursor [221](#)
 AT-imagebgc [222](#)
 AT-imagefgc [223](#)
 AT-incrtime [224](#)
 AT-index [225](#)
 AT-indexscope [227](#)
 AT-input [228](#)
 AT-instance [229](#)
 AT-interfaceid [230](#)
 AT-is-applet [231](#)
 AT-itemcount [232](#)
 AT-itemorder [233](#)
 AT-ixmlomdocument2 [234](#)
 AT-ixmlomnode [235](#)

AT-ixmlDOMNodeList [236](#)
AT-keyboard [237](#)
AT-label [209](#), [238](#)
AT-language [242-243](#)
AT-lastChild [244](#)
AT-lastMenu [245](#)
AT-lastRecord [246](#)
AT-lastSubControl [247](#)
AT-lastToolBar [248](#)
AT-layoutBox [249](#)
AT-level [250](#)
AT-license-key [252](#)
AT-lineMotion [253](#)
AT-load [254](#)
AT-local [255](#)
AT-logFile [256](#)
AT-majorTabHeight [257](#)
AT-majorTabWidth [258](#)
AT-mapped [259](#)
AT-mapping [260](#)
AT-masterApplication [261](#)
AT-maxChars [265-266](#)
AT-maxHeight [267](#)
AT-maximized [268](#)
AT-maxSize [269](#)
AT-maxValue [270](#)
AT-maxWidth [272](#)
AT-member [273](#)
AT-memberCount [274](#)
AT-menu [275-276](#)
AT-menuBgc [277](#)
AT-menuCount [278](#)
AT-menuFgc [279](#)
AT-message [280](#)
AT-minColWidth [281-282](#)
AT-minHeight [283](#)
AT-minOrTabHeight [284](#)
AT-minOrTabWidth [285](#)
AT-minRowHeight [286](#)
AT-minSize [287](#)
AT-minValue [288](#)
AT-minWidth [290](#)
AT-mode [291](#)
AT-model [292](#)
AT-module [293](#)
AT-mouseButtons [294](#)
AT-mouseOver [295](#)
AT-moveable [296](#)
AT-msgBoxText [297](#)
AT-multiline [299](#)
AT-multisel [300](#)
AT-mustExist [303](#)
AT-name [304](#)
AT-navigable [306](#)
AT-navigation [308](#)
AT-nextActive [309-310](#)
AT-nodetype [311](#)
AT-notePage [312](#)
AT-open [313](#)
AT-opsys-string [314](#)
AT-opsys-type [315](#)
AT-options [316](#)
AT-order [340](#)
AT-output [342](#)

AT-overridecursor [343](#)
 AT-pagemotion [345](#)
 AT-parent [346](#)
 AT-password [347](#)
 AT-path [348](#)
 AT-pattern [349](#)
 AT-picheight [350](#)
 AT-picture [352-353, 357](#)
 AT-picture-hilite [359](#)
 AT-picwidth [360](#)
 AT-pointer-height [362-363](#)
 AT-pointer-width [364-365](#)
 AT-posraster [366](#)
 AT-preedit [367](#)
 AT-preeditssel [368](#)
 AT-privatekeyfile [369](#)
 AT-propscale [370](#)
 AT-publicid [371](#)
 AT-publickeyfile [372](#)
 AT-real-height [373](#)
 AT-real-modified [374](#)
 AT-real-screen [375](#)
 AT-real-sensitive [376](#)
 AT-real-shadowobject [377](#)
 AT-real-size [378](#)
 AT-real-version [379](#)
 AT-real-visible [380](#)
 AT-real-width [381](#)
 AT-real-x [382](#)
 AT-real-xraster [383](#)
 AT-real-y [384](#)
 AT-real-yraster [385](#)
 AT-record [386](#)
 AT-recordcount [387](#)
 AT-reffont [391](#)
 AT-refstring [393](#)
 AT-rgb [393](#)
 AT-root [395](#)
 AT-rowalignment [396](#)
 AT-rowcount [397](#)
 AT-rowfirst [398](#)
 AT-rowheader [399](#)
 AT-rowheadfgc [400](#)
 AT-rowheadfont [401](#)
 AT-rowheadshadow [402](#)
 AT-rowheadvisible [403](#)
 AT-rowheight [404](#)
 AT-rowlinewidth [405](#)
 AT-rowsizeable [406](#)
 AT-rowvisible [407](#)
 AT-scale [408](#)
 AT-scalestyle [410](#)
 AT-scope [412-413](#)
 AT-screen [414-415](#)
 AT-screen-height [416-417](#)
 AT-screen-width [418-423, 538-541](#)
 AT-screencount [424](#)
 AT-searchpath [425](#)
 AT-selected [426](#)
 AT-selection [427](#)
 AT-self [428](#)
 AT-selstyle [429](#)
 AT-sensitive [431, 433-434](#)
 AT-shadowattr [435](#)

AT-shadowindex [436](#)
AT-shadowinstance [437](#)
AT-shadowobject [439](#)
AT-shortdaynames [440](#)
AT-showitem [441](#)
AT-size [441-442](#)
AT-sizeable [443-444](#)
AT-sizeraster [445](#)
AT-smallpicheight [446](#)
AT-smallpicture [447](#)
AT-smallpicwidth [448](#)
AT-source [449](#)
AT-spacing [450](#)
AT-specified [451](#)
AT-startsel [452](#)
AT-starttime [454](#)
AT-state [455](#)
AT-static [456](#)
AT-statusbar [457](#)
AT-statushelp [458](#)
AT-style [459](#), [467](#)
AT-subcontrol [469](#)
AT-subcontrolcount [470](#)
AT-sysmodal [471](#)
AT-systemerror [472](#)
AT-systemid [473](#)
AT-tabalignment [474](#)
AT-tabshape [475](#)
AT-tabtype [476](#)
AT-target [477](#)
AT-terminal [478](#)
AT-terminaltype [479](#)
AT-text [480-481](#), [483](#)
AT-textbgc [484](#)
AT-textfgc [485](#)
AT-textwidth [486](#)
AT-tile [487](#)
AT-tiledpi [488](#)
AT-tilestyle [489](#)
AT-timeout [491](#)
AT-title [492](#)
AT-titlebar [493](#)
AT-titlebgc [494](#)
AT-titlefgc [495](#)
AT-today [496](#)
AT-todaymarker [497](#)
AT-toolbar [498-499](#)
AT-toolbarcount [500](#)
AT-toolhelp [501](#)
AT-toolkit [503](#)
AT-toolkit-string [504](#)
AT-toolkit-version [505](#)
AT-top-most [506](#)
AT-topitem [507](#)
AT-tracefile [508](#)
AT-tracetime [509](#)
AT-tracing [511](#)
AT-trailingdates [512](#)
AT-transformer [513](#)
AT-transport [514](#)
AT-type [515](#)
AT-typescope [516-517](#)
AT-userdata [518-520](#)
AT-username [521](#)

AT-userplaced [522](#)
AT-uuid [523](#)
AT-value [524](#), [527](#)
AT-version [528](#)
AT-version-string [530](#)
AT-vheight [531](#)
AT-visible [532](#)
AT-vsbs-arrows [533](#)
AT-vsbs-linemotion [534](#)
AT-vsbs-optional [535](#)
AT-vsbs-pagemotion [536](#)
AT-vsbs-visible [537](#)
AT-vwidth [542](#)
AT-weeknumbers [543](#)
AT-weight [544](#)
AT-width [545-546](#)
AT-window [548](#)
AT-winsys [549](#)
AT-winsys-string [550](#)
AT-winsys-version [551](#)
AT-wrap [552](#)
AT-x [553](#)
AT-xalignment [554](#)
AT-xauto [555](#)
AT-xdpi [556-557](#)
AT-xleft [558](#)
AT-xmargin [559](#)
AT-xml [561](#)
AT-xorigin [562](#)
AT-xraster [563](#)
AT-xright [565](#)
AT-xspacing [566](#)

AT-y [567](#)
AT-yalignment [568](#)
AT-yauto [569](#)
AT-ybottom [570](#)
AT-ydpi [571-572](#)
AT-ymargin [573](#)
AT-yorigin [575](#)
AT-yraster [576](#)
AT-yspacing [578](#)
AT-ytop [579](#)
AT_acc_label [25](#)
AT_acc_text [27](#)
AT_accelerator [29](#)
AT_action [30](#)
AT_active [31](#), [33-34](#)
AT_activeitem [35](#)
AT_activeobject [37](#)
AT_alignment [38](#)
AT_allowundefined [40](#)
AT_application [41](#)
AT_arrows [42](#)
AT_attribute [43-44](#)
AT_autoalign [46](#)
AT_autosize [47](#)
AT_barwidth [49](#)
AT_bgc [50-51](#), [239](#)
AT_binding [48](#), [52](#)
AT_bordercolor [53](#)
AT_borderraster [54](#)
AT_borderstyle [59](#)
AT_borderwidth [62](#)
AT_button [63](#)

AT_bw 64
AT_calendaralignment 66
AT_canvasfunc 67
AT_certificatefile 68
AT_changedir 69
AT_child 70
AT_childcount 71
AT_class 72
AT_closeable 73
AT_codepage 74
AT_colalignment 77
AT_colcount 78
AT_colfirst 79
AT_colheader 80
AT_colheadfgc 81
AT_colheadfont 82
AT_colheadshadow 83
AT_colheadvisible 84
AT_collinewidth 85
AT_color 86
AT_color_type 87-88
AT_colorcount 89-90
AT_colorname 91
AT_colsizeable 92
AT_coltitle 93
AT_colvisible 94
AT_colwidth 95
AT_configurable 97
AT_connect 98
AT_constant 101
AT_content 102, 104-105
AT_contentfunc 107
AT_control 108
AT_count 109
AT_cursor 110
AT_cursorname 111
AT_curvalue 112
AT_cut_pending 113
AT_cut_pending_changed 114
AT_data 115
AT_dataget 116
AT_dataindex 117
AT_datamap 118
AT_datamodel 119
AT_dataoptions 120
AT_dataselect 122
AT_dataselectattr 124
AT_dataselectcount 125
AT_dataselecttype 126
AT_dataset 128
AT_defbutton 129
AT_deltavalue 130
AT_depth 131
AT_dialog 132
AT_dialogbox 133
AT_directory 139
AT_display 140
AT_doccursor 141
AT_dock_line 142
AT_dock_offset 143
AT_dockable 144
AT_docking 146
AT_document 147
AT_editable 148, 150

AT_editpos [151](#)
 AT_edittest [152](#)
 AT_editttype [153](#)
 AT_endsel [154](#)
 AT_env [155](#)
 AT_envvar [156](#)
 AT_errfile [157](#)
 AT_errorcode [158](#)
 AT_event [159](#)
 AT_event_code [160](#)
 AT_eventcount [161](#)
 AT_exec [162](#)
 AT_extension [165](#)
 AT_external [166-167](#)
 AT_extevent [168](#)
 AT_face [169](#)
 AT_fgcolor [170-171](#)
 AT_field [172](#)
 AT_fieldactive [173](#)
 AT_fieldfocus [174-175](#)
 AT_fieldfocusable [176](#)
 AT_fieldshadow [177](#)
 AT_filled [178](#)
 AT_firstchar [179](#)
 AT_firstchild [180](#)
 AT_firstmenu [181](#)
 AT_firstrecord [182](#)
 AT_firstsubcontrol [183](#)
 AT_firsttoolbar [184](#)
 AT_focus [185](#)
 AT_focus_on_click [188](#)
 AT_focusitem [190](#)
 AT_font [191-192](#)
 AT_fontname [193](#)
 AT_format [194, 197](#)
 AT_formatfunc [198](#)
 AT_function [199](#)
 AT_gradient [199](#)
 AT_grey [201](#)
 AT_groupbox [202](#)
 AT_height [203-204](#)
 AT_help [206](#)
 AT_helpmenu [207](#)
 AT_helppos [208](#)
 AT_hls [210](#)
 AT_hsb_arrows [211](#)
 AT_hsb_linemotion [212](#)
 AT_hsb_optional [213](#)
 AT_hsb_pagemotion [214](#)
 AT_hsb_visible [215](#)
 AT_icon [216](#)
 AT_iconic [218](#)
 AT_iconifiable [219](#)
 AT_idispatch [220](#)
 AT_ignorecursor [221](#)
 AT_imagebgc [222](#)
 AT_imagefgc [223](#)
 AT_incrtime [224](#)
 AT_index [225](#)
 AT_indexscope [227](#)
 AT_input [228](#)
 AT_instance [229](#)
 AT_interfaceid [230](#)
 AT_irection [135](#)

AT_is_applet [231](#)
AT_itemcount [232](#)
AT_itemorder [233](#)
AT_ixmlomdocument2 [234](#)
AT_ixmlomnode [235](#)
AT_ixmlomodelist [236](#)
AT_keyboard [237](#)
AT_label [209](#), [238](#)
AT_language [242-243](#)
AT_lastchild [244](#)
AT_lastmenu [245](#)
AT_lastrecord [246](#)
AT_lastsubcontrol [247](#)
AT_lasttoolbar [248](#)
AT_layoutbox [249](#)
AT_level [250](#)
AT_license_key [252](#)
AT_linemotion [253](#)
AT_load [254](#)
AT_local [255](#)
AT_logfile [256](#)
AT_majortabheight [257](#)
AT_majortabwidth [258](#)
AT_mapped [259](#)
AT_mapping [260](#)
AT_masterapplication [261](#)
AT_maxchars [265-266](#)
AT_maxheight [267](#)
AT_maximized [268](#)
AT_maxsize [269](#)
AT_maxvalue [270](#)
AT_maxwidth [272](#)
AT_member [273](#)
AT_membercount [274](#)
AT_menu [275-276](#)
AT_menubgc [277](#)
AT_menucount [278](#)
AT_menufgc [279](#)
AT_message [280](#)
AT_mincolwidth [281-282](#)
AT_minheight [283](#)
AT_minortabheight [284](#)
AT_minortabwidth [285](#)
AT_minrowheight [286](#)
AT_minsize [287](#)
AT_minvalue [288](#)
AT_minwidth [290](#)
AT_mode [291](#)
AT_model [292](#)
AT_module [293](#)
AT_mouse_buttons [294](#)
AT_mouseover [295](#)
AT_moveable [296](#)
AT_msgboxtext [297](#)
AT_multiline [299](#)
AT_multisel [300](#)
AT_mustexist [303](#)
AT_name [304](#)
AT_navigable [306](#)
AT_navigation [308](#)
AT_nextactive [309-310](#)
AT_nodetype [311](#)
AT_notepage [312](#)
AT_open [313](#)

AT_opsys_string [314](#)
AT_opsys_type [315](#)
AT_options [316](#)
AT_order [340](#)
AT_output [342](#)
AT_overridecursor [343](#)
AT_pagemotion [345](#)
AT_parent [346](#)
AT_password [347](#)
AT_path [348](#)
AT_pattern [349](#)
AT_picheight [350](#)
AT_picture [352-353, 357](#)
AT_picture_hilite [359](#)
AT_picwidth [360](#)
AT_pointer_height [362-363](#)
AT_pointer_width [364-365](#)
AT_posraster [366](#)
AT_preedit [367](#)
AT_preeditssel [368](#)
AT_privatekeyfile [369](#)
AT_propscale [370](#)
AT_publicid [371](#)
AT_publickeyfile [372](#)
AT_real_height [373](#)
AT_real_modified [374](#)
AT_real_screen [375](#)
AT_real_sensitive [376](#)
AT_real_shadowobject [377](#)
AT_real_size [378](#)
AT_real_version [379](#)
AT_real_visible [380](#)
AT_real_width [381](#)
AT_real_x [382](#)
AT_real_xraster [383](#)
AT_real_y [384](#)
AT_real_yraster [385](#)
AT_record [386](#)
AT_recordcount [387](#)
AT_reffont [391](#)
AT_refstring [393](#)
AT_rgb [393](#)
AT_root [395](#)
AT_rowalignment [396](#)
AT_rowcount [397](#)
AT_rowfirst [398](#)
AT_rowheader [399](#)
AT_rowheadfgc [400](#)
AT_rowheadfont [401](#)
AT_rowheadshadow [402](#)
AT_rowheadvisible [403](#)
AT_rowheight [404](#)
AT_rowlinewidth [405](#)
AT_rowsizeable [406](#)
AT_rowvisible [407](#)
AT_scale [408](#)
AT_scalestyle [410](#)
AT_scope [412-413](#)
AT_screen [414-415](#)
AT_screen_height [416-417](#)
AT_screen_width [418-423, 538-541](#)
AT_screencount [424](#)
AT_searchpath [425](#)
AT_selected [426](#)

AT_selection [427](#)
AT_self [428](#)
AT_selstyle [429](#)
AT_sensitive [431](#), [433-434](#)
AT_shadowattr [435](#)
AT_shadowindex [436](#)
AT_shadowinstance [437](#)
AT_shadowobject [439](#)
AT_shortdaynames [440](#)
AT_showitem [441](#)
AT_size [441-442](#)
AT_sizeable [443-444](#)
AT_sizeraster [445](#)
AT_smallpicheight [446](#)
AT_smallpicture [447](#)
AT_smallpicwidth [448](#)
AT_source [449](#)
AT_spacing [450](#)
AT_specified [451](#)
AT_startsel [452](#)
AT_starttime [454](#)
AT_state [455](#)
AT_static [456](#)
AT_statusbar [457](#)
AT_statushelp [458](#)
AT_style [459](#), [467](#)
AT_subcontrol [469](#)
AT_subcontrolcount [470](#)
AT_sysmodal [471](#)
AT_systemerror [472](#)
AT_systemid [473](#)
AT_tabalignment [474](#)
AT_tabshape [475](#)
AT_tabtype [476](#)
AT_target [477](#)
AT_terminal [478](#)
AT_terminaltype [479](#)
AT_text [480-481](#), [483](#)
AT_textbgc [484](#)
AT_textfgc [485](#)
AT_textwidth [486](#)
AT_tile [487](#)
AT_tiledpi [488](#)
AT_tilestyle [489](#)
AT_timeout [491](#)
AT_title [492](#)
AT_titlebar [493](#)
AT_titlebgc [494](#)
AT_titlefgc [495](#)
AT_today [496](#)
AT_todaymarker [497](#)
AT_toolbar [498-499](#)
AT_toolbarcount [500](#)
AT_toolhelp [501](#)
AT_toolkit [503](#)
AT_toolkit_string [504](#)
AT_toolkit_version [505](#)
AT_top_most [506](#)
AT_topitem [507](#)
AT_tracefile [508](#)
AT_tracetime [509](#)
AT_tracing [511](#)
AT_trailingdates [512](#)
AT_transformer [513](#)

AT_transport 514
AT_type 515
AT_typescope 516-517
AT_userdata 518-520
AT_username 521
AT_userplaced 522
AT_uuid 523
AT_value 524, 527
AT_version 528
AT_version_string 530
AT_vheight 531
AT_visible 532
AT_vsb_arrows 533
AT_vsb_linemotion 534
AT_vsb_optional 535
AT_vsb_pagemotion 536
AT_vsb_visible 537
AT_vwidth 542
AT_weeknumbers 543
AT_weight 544
AT_width 545-546
AT_window 548
AT_winsys 549
AT_winsys_string 550
AT_winsys_version 551
AT_wrap 552
AT_x 553
AT_xalignment 554
AT_xauto 555
AT_xdpi 556-557
AT_xleft 558
AT_xmargin 559

AT_xml 561
AT_xorigin 562
AT_xraster 563
AT_xright 565
AT_xspacing 566
AT_y 567
AT_yalignment 568
AT_yauto 569
AT_ybottom 570
AT_ydpi 571-572
AT_ymargin 573
AT_yorigin 575
AT_yraster 576
AT_yspacing 578
AT_ytop 579
Attribut
 benutzerdefiniert 377, 435-436, 439, 515
 benutzerdefiniertshadow-Attribut 109
 name 477
attribute 43
attribute[integer] 44
Auflösung 24
autoalign 46
autosize 47

B

backpage 48
barwidth 49
Bedienbarkeit 148, 306, 431
benutzerdefinierte Attribute 273-274, 377, 435-436, 439

benutzerdefiniertes Attribut [515](#)

 Bezeichner [239](#)

Betriebssystem [314-315](#)

Bezeichner [239](#)

 Argument [239](#)

 benutzerdefiniertes Attribut [239](#)

 Funktion [239](#)

 Methode [239](#)

bgc [50](#)

bgc[index] [51](#)

Bild [320](#)

bind_none [52](#)

bind_organizer [52](#)

bind_solid [52](#)

bind_spiral [52](#)

binding [52](#)

Bindung [48, 136](#)

Bitmap [492](#)

border_compat [59](#)

border_none [59](#)

border_plain [59](#)

border_raised [59](#)

border_sunken [59](#)

border_toolkit [59](#)

bordercolor [53](#)

borderraster [54](#)

borderstyle [59](#)

borderwidth [62](#)

bp_bottomleft [48, 136](#)

bp_bottomright [48, 136](#)

bp_topleft [48, 136](#)

bp_topright [48, 136](#)

Button-Kombinationen [63](#)

button[integer] [63](#)

button_abort [63, 297](#)

button_cancel [63, 297](#)

button_ignore [63, 297](#)

button_no [63, 297](#)

button_ok [63, 297](#)

button_retry [63, 297](#)

button_yes [63, 297](#)

Buttonstrings [297](#)

.bw [64](#)

C

Caching

 doccursor [121](#)

calendaralignment [66](#)

canvas [316](#)

Canvas-Funktion [316](#)

canvasfunc [67](#)

CCR_focus [186](#)

.certificatefile [68](#)

changedir [69](#)

checkbox [318](#)

child[integer] [70](#)

childcount [71](#)

class [72](#)

closeable [73](#)

.codepage [74](#)

colalignment[integer] [77](#)

colcount [78](#)

colfirst [79](#)

colheader [80](#)

- colheadfgc [81](#)
- colheadfont [82](#)
- colheadshadow [83](#)
- colheadvisible [84](#)
- collinewidth[integer] [85](#)
- color [86](#)
- color_black [65](#)
- color_blue [393](#)
- color_green [393](#)
- color_hue [210](#)
- color_light [210](#)
- color_red [393](#)
- color_sat [210](#)
- color_type [87](#)
- color_type[integer] [88](#)
- color_white [65](#)
- colorcount [89](#)
- colorcount[integer] [90](#)
- colormname[integer] [91](#)
- colsizeable[integer] [92](#)
- coltitle[integer] [93](#)
- coltype_bw [87-88](#)
- coltype_color [87-88](#)
- coltype_grey [87-88](#)
- colvisible[integer] [94](#)
- colwidth[integer] [95](#)
- Combobox [323](#)
- Composite-Widget [337](#)
- configurable [97](#)
- connect [98](#)
 - SSL [98](#)
- constant [101](#)
- content [102](#)
- content[index] [105](#)
- content[integer] [104](#)
- contentfunc [107](#)
- control [108](#)
 - Hierarchie [108](#)
- Control [340](#)
- count [109](#)
- cp_acp [74](#)
- cp_ascii [74](#)
- cp_cp1252 [74](#)
- cp_cp437 [74](#)
- cp_cp850 [74](#)
- cp_dec169 [75](#)
- cp_euc [75](#)
- cp_hp15 [75](#)
- cp_iso6937 [75](#)
- cp_iso8859 [75](#)
- cp_jap15 [75](#)
- cp_kor15 [75](#)
- cp_prc15 [75](#)
- cp_roc15 [75](#)
- cp_roman8 [75](#)
- cp_ucp [75](#)
- cp_utf16 [75](#)
- cp_utf16b [75](#)
- cp_utf16l [75](#)
- cp_utf8 [76](#)
- cp_utfwin [76](#)
- cp_wcs [76](#)
- cp_winansi [76](#)
- cursor [110](#)

Cursor
 temporär 343
cursorname[integer] 111
curvalue 42, 112
cut_pending 113
cut_pending_changed 114

D

data 115
dataget[attribute] 116
dataindex[attribute] 117
datamap[attribute] 118
Datamodel-Attribut
 Datentyp 126
 Kardinalität 125
datamodel[attribute] 119
dataoptions[enum] 120
dataselect[attribute] 122
dataselectattr[attribute] 124
dataselectcount[attribute] 125
dataselecttype[attribute] 126
dataset[attribute] 128
Datentyp 515
 Datamodel-Attribut 126
deactivate 491
Default 412
Default-Pushbutton 129
defbutton 129
deltavalue 130
depth 131
dialog 132, 318
dialogbox 133

Dialogbox 133, 471
direction 48, 135-136
directory 139
display 140
DM_CanvasUserArgs 317
DM_SetValue 232
doccursor
 Caching 121
doccursor[integer] 141
dock_down 144, 146
dock_left 144, 146
dock_line 142
dock_offset 143
dock_right 144, 146
dock_up 144, 146
dock_window 144, 146
dockable[enum] 144
docking 146
document[integer] 147
DOM-Knoten
 Instruktion 477
 Kindknoten 480
 Unterknoten 480
dopt_apply_on_event 120
dopt_apply_on_unmap 120
dopt_cache_data 121
dopt_propagate_on_changed 121
dopt_propagate_on_start 120
dopt_represent_on_init 120
dopt_represent_on_map 120
dots per inch 556-557, 571-572
DPI 556-557, 571-572

dynlib [514](#)

E

edit_locking [153](#)

edit_offline [153](#)

edit_online [153](#)

editable [148](#), [306](#), [431](#)

editable[index] [150](#)

Editierbarkeit [148](#), [306](#), [431](#)

editpos [151](#)

edittext [152](#), [318](#)

 mehrzeilig [299](#)

edittype [153](#)

Eingabemodus [32](#)

Eingabeverhalten [368](#)

endsel [154](#)

env[string] [155](#)

envvar[string] [156](#)

Ereignis

 select-activate [29](#)

Erreichbarkeit von Objekten [338](#)

errfile [157](#)

error_file [158](#)

error_network [158](#)

error_none [158](#)

error_protocol [158](#)

error_unavail [158](#)

error_version [158](#)

errorcode [158](#)

Escape-Taste [64](#)

event[event] [159](#)

event[integer] [159](#)

event_code [160](#)

eventcount [159](#), [161](#)

exec [162](#)

 SSH [162](#)

explizite Aktivierung [368](#)

export [164](#)

 Vererbung [388](#)

extended [429](#)

extension [165](#)

external [166](#)

external[integer] [167](#)

extevent [168](#)

F

.face [169](#)

face_black [465](#), [544](#)

face_bold [465](#), [544](#)

face_default [169](#), [465](#), [544](#)

face_demibold [465](#), [544](#)

face_italic [169](#), [465](#)

face_light [465](#), [544](#)

face_medium [465](#), [544](#)

face_normal [465](#), [544](#)

face_oblique [169](#), [465](#)

face_roman [169](#), [465](#)

Farbsättigung [210](#)

Farbton [210](#)

Fehlerdatei [157](#)

Fehlermeldung [157](#)

Feld

 Hintergrundfarbe [51](#)

Fenster 333
 vergrößern 443
Fenster schließen 334
fgc 170
fgc[index] 171
field[index] 172
fieldactive[index] 173
fieldfocus 174
fieldfocus[index] 175
fieldfocusable 176
fieldshadow[I,J] 177
filereq 319
filled 178
firstchar 179
firstchild 180
firstmenu 181
firstrecord 182
firstsubcontrol 183
firsttoolbar 184
focus 185, 187
focus [I,J] 187
focus_on_click 188
focusable 189
focusitem 190
Fokus 337
 Erreichbarkeit 338
 Motif 337
 negative Koordinaten 337
 negative Positionsangaben 337
 Objektposition 337
 Sichtbarkeit 337
Fokus-Objekt 186

Fokussierbarkeit 148, 306, 431
font 191
 Stil 465
font raster 339
fontindex] 192
fontname[integer] 193
Fontraster 339
format 194, 198
Format 194
Format-Funktion 194, 198
Format-Ressource 194, 198
Format-Variante 196
format[index] 197
formatfunc 194, 198
Formatstring 195
fr_directory 464
fr_load 464
fr_save 464
fro_createprompt 319
fro_overwriteprompt 319
fro_relativepath 319
frt_cancel 481
frt_directories 481
frt_files 481
frt_help 481
frt_nomatch 481
frt_ok 481
frt_path 481
frt_pattern 481
frt_selection 481
frt_update 481
func_callback 261

func_canvas 261
func_content 262
func_data 262
func_format 262
func_normal 262
func_spinbox 262
function 199
Funktion 199, 515
 Bezeichner 239

G

.gradient 199
.grey 201
Größenraster 445
groupbox 202, 319
 Hierarchie 202

H

Hauptindex 257-258, 476
Header 171, 403
height 203, 376
height[class] 204
.height[enum] 204
Helligkeit 210
help 206
helpmenu 207
Helpmenü 207
helppos 208, 458
hierarchische Vorlage 276
HighDPI 24
Hintergrundfarbe 50, 222
 Feld 51

.hinttext 209
HLS-Farbmodell 210
.hls[enum] 209
host 162
Host 98, 162
hsb_arrows 211
hsb_linemotion 212
hsb_optional 213
hsb_pagemotion 214
hsb_visible 215
hue 210

I

icon 216
icon_asterisk 216
icon_error 216
icon_exclamation 216
icon_hand 216
icon_information 216
icon_query 217
icon_question 217
icon_warning 217
iconic 218
iconifizierbar 219
Iconifiziermechanismus 219
iconifiable 219
Identifikator 3
idispach 220
IDM-Versionsstring 530
IDM_ApplIcon 216
IDM_DeflIcon 216
IDM_FONTRASTER_COMPAT 339

IDMfontraster_compat [339](#)

IDMtracefile [511](#)

ignorecursor [221](#)

image [320](#)

imagebgc [222](#)

imagefgc [223](#)

implizite Aktivierung [368](#)

incremental time [224](#)

Incrementaltime [491](#)

incrtime [224](#)

index [225](#)

indexscope[attribute] [227](#)

input[integer] [228](#)

instance[integer] [229](#)

Instanz [412](#)

Instruktion [477](#)

interfaceid [230](#)

is_applet [231](#)

itemcount [232](#)

itemorder [233](#)

ixmlomdocument2 [234](#)

ixmlomnode [235](#)

ixmlomodelist [236](#)

K

Kalenderwoche [543](#)

Kardinalität

 Datamodel-Attribut [125](#)

keyboard [237](#)

Kind-Objekte [70](#)

Kindknoten [480](#)

Konfigurationsdatei [97](#)

Koordinate

 negative [337](#)

L

label [238](#)

Label [239](#)

.label[anyvalue] [239](#)

lang_c [242](#), [261](#)

lang_cobol [242](#), [261](#)

lang_cobol_cancel [242](#), [261](#)

lang_default [242](#), [261](#)

lang_fortran [261](#)

lang_java [261](#)

lang_pascal [261](#)

language [242](#)

language[integer] [243](#)

lastchild [244](#)

lastmenu [245](#)

lastrecord [246](#)

lastsubcontrol [247](#)

lasttoolbar [248](#)

layoutbox [249](#), [320](#)

level[integer] [250](#)

license_key [252](#)

lightness [210](#)

linemotion [42](#), [253](#)

listbox [321](#)

load [254](#)

local [255](#)

logfile [256](#)

luminance [210](#)

M

- Majortab [48](#), [136](#), [257-258](#), [476](#)
- majortabheight [257](#)
- majortabwidth [258](#)
- mapped [259](#)
- mapping[integer] [260](#)
- .masterapplication[enum] [261](#)
- Mauscursorbreite [364-365](#)
- Mauscursorhöhe [362-363](#)
- maxchars [265](#)
- maxchars[index] [266](#)
- maxheight [267](#)
- maximized [268](#)
- maxsize[integer] [269](#)
- maxvalue [270](#)
- maxwidth [272](#)
- Mehrfachselektion in Motif [301](#)
- member[integer] [273](#)
- membercount [274](#)
- menu [275](#)
- Menü [275](#)
 - Reihenfolge [340](#)
- menu[integer] [276](#)
- menubgc [277](#)
- menubox [322](#)
- MENUBOX [275](#)
- Menübox [207](#)
- menucount [278](#)
- Menüelement [278](#)
- menufgc [279](#)
- menuitem [322](#)
- Menüleiste [207](#), [279](#)
- message[integer] [280](#)
- messagebox [297](#), [343](#)
- Methode
 - Bezeichner [239](#)
- mincolwidth [281](#)
- mincolwidth[integer] [282](#)
- minheight [283](#)
- Minortab [48](#), [136](#), [284-285](#), [476](#)
- minortabheight [284](#)
- minortabwidth [285](#)
- minrowheight [286](#)
- minsize[integer] [287](#)
- minvalue [288](#)
- minwidth [290](#)
- mode [291](#)
- mode_client [291](#)
- mode_none [291](#)
- mode_server [291](#)
- model [292](#)
- Modell [412](#)
- module [293](#)
- Motif
 - Fokussierbarkeit [337](#)
 - Tastaturnavigation [337](#)
- mouse_buttons [294](#)
- mouseover [295](#)
- moveable [296](#)
- msgboxtext[enum] [297](#)
- MSXML [528](#)
- multiline [299](#)
- multiple [429](#)

multisel [300](#)

Mustervariante [487](#)

mustexist [303](#)

N

name [304](#), [477](#)

navi_default [308](#)

navi_grouped [308](#)

navi_tab [308](#)

navigable [148](#), [306](#), [431](#)

navigation [308](#)

nckpage [48](#), [136](#)

Nebenindex [284-285](#), [476](#)

nextactive[index] [310](#)

nextactive[integer] [309](#)

nodetype [311](#), [477](#)

nodetype_attribute [311](#)

nodetype_cdata_section [311](#)

nodetype_comment [311](#)

nodetype_document [311](#)

nodetype_document_fragment [311](#)

nodetype_document_type [311](#)

nodetype_element [311](#)

nodetype_entity [311](#)

nodetype_entity_reference [311](#)

nodetype_notation [311](#)

nodetype_processing_instruction [311](#), [477](#)

nodetype_text [311](#)

noicon [217](#)

notebook [322](#)

Notebook [31](#)

notepage [312](#), [322](#), [480](#), [492](#)

Label [37](#)

Notepage [29](#)

Label [31](#)

O

Objekt

Hintergrundfarbe [50](#)

Objekthierarchie [431](#), [532](#)

Objektidentifikator [292](#)

Objektklasse [72](#)

open[integer] [313](#)

opsys_string [314](#)

opsys_type [315](#)

opt_accept_child [316](#)

opt_addevents [316](#)

opt_animated [333](#)

opt_auto_close [334](#)

opt_balloon_toolhelp [327](#)

opt_center_toolhelp [317-323](#), [325-329](#),
[332](#), [334](#)

opt_cert_required [316](#)

opt_enable_tearoff [322](#)

opt_et_margin [318](#)

opt_focus_frame [317](#)

opt_fontraster_compat [328](#), [339](#)

opt_graphicsview [317](#)

opt_help [334](#)

opt_hscroll [324](#)

opt_html [318](#)

opt_mnemonic [324](#)

opt_motif_shadow [317](#)

- opt_nested_docks 335
- opt_new_align 329
- opt_new_colwidth 95, 330
- opt_no_ssl_v2 316
- opt_old_colwidth 330
- opt_old_select 324, 331
- opt_push_like 318, 326
- opt_quick_navigate 321, 325
- opt_rightclick_selects 332
- opt_rtf 319
- opt_scroll_on_focus 320-321, 323, 333, 335, 337
- opt_scroll_pixels 322
- opt_sim_insensitive 320
- opt_sort 325
- opt_tabbed_docks 336
- opt_use_widget 318, 322, 326, 329
- opt_verify_peer 316
- opt_w2kprefsize_compat 338
- opt_window_size 336
- opt_wntsizebug_compat 338
- opt_xmborder_compat 331
- opt_yi_monitoring 327
- options[enum] 316
- order 340
- os_nt 315
- os_unix 315
- os_w64 315
- output[integer] 342
- overridecursor 343

P

- pagemotion 345
- parent 346
- .password 347
- path 162, 348
- pattern 349
- Pfeile 42
- picheight 350
- picture 352
- picture[enum] 353
- picture[integer] 357
- picture_hilite[integer] 359
- picwidth 360
- Pixelkoordinaten 392, 564, 577
- pointer_height 362
- pointer_height[integer] 363
- pointer_width 364
- pointer_width[integer] 365
- poptext 323
- Popup-Menü 275
- Positionsangabe
 - negative 337
- Positions raster 366
- posraster 366
- .preedit 367
- preedit_offthespot 367
- preedit_onthespot 367
- preedit_overthespot 367
- preedit_root 367
- .preedit sel 368
- presel_all 368

presel_begin 368
presel_default 368
presel_end 368
.privatekeyfile 369
Processing-Instruction 477
Programmiersprache 242
progressbar 325
.propscale 370
publicid 371
.publickeyfile 372
pushbutton 326

Q

QFrame 317
QGraphicsScene 317
QGraphicsView 317

R

radiobutton 326
Raster 366, 445
raster width 339
Rasterattribute 391, 563, 576
Rasterbreite 339, 383
Rasterhöhe 385
Rasterkoordinaten 257-258, 284-285
real_height 373
real_modified 374
real_screen 375
real_sensitive 376
real_shadowobject 377
real_size[integer] 378
real_version[enum] 379

real_visible 380
real_width 381
real_x 382
real_xraster 383
real_y 384
real_yraster 385
Rechteck 326
record[integer] 386
recordcount 387
rectangle 275, 326
reexport 388
 Vererbung 388
reference string 339
Referenzfont 391, 563, 576
Referenzstring 339
reffont 21, 391, 563, 576
refstring 393
Regel 515
Regelsyntax 185
Ressource 72
.rgb[enum] 393
root 395
rowalignment[integer] 396
rowcount 397
rowfirst 398
rowheader 399
rowheadfgc 400
rowheadfont 401
rowheadshadow 402
rowheadvisible 403
rowheight[integer] 404
rowlinewidth[integer] 405

rowsizeable[integer] 406
rowvisible[integer] 407

S

Sättigung 210
saturation 210
savedfocus 185
scale_factor 205, 547
scale_offset 205, 547
.scalestyle 410
Schieberegler 42
schließen
 Fenster 334
Schriftschnitt 465
Schriftstil 465
scope 412
scope[attribute] 413
screen 414
screen[integer] 415
screen_height 416
screen_height[integer] 417
screen_width 418
screen_width[integer] 419
.screen_x 420
.screen_x[integer] 421
.screen_y 422
.screen_y[integer] 423
screencount 424
Scroll-Position
 horizontale 179
scrollbar 327
Scrollbarslider 112

.searchpath 425
sel_column 427
sel_header 427
sel_row 427
sel_single 427
selected[integer] 426
selection[enum] 427
Selektierbarkeit 148, 306, 431
Selektion
 einfach 35
 mehrfach 35
Selektionsmuster 122
self 428
selstyle 429
sensitive 148, 306, 431, 532
sensitive[index] 434
sensitive[integer] 433
Server 98
setup 327
shadow-Attribut 435-436, 439
shadowattr 435
shadowindex 436
shadowinstance 437
shadowobject 439
shape_chamfered 475
shape_polygon 475
shape_rounded 475
shape_square 475
shortdaynames 440
showitem 441
Sichtbarmachen des Objekts 186
single 429

- `.size` [441](#)
- `size[integer]` [442](#)
- `sizeable` [443](#)
- `sizeable[class]` [444](#)
- `sizeraster` [95](#), [257-258](#), [284-285](#), [445](#)
- `Skalierung` [24](#)
- `Sliderposition` [270](#)
- `smallpicheight` [446](#)
- `smallpicture[integer]` [447](#)
- `smallpicwidth` [448](#)
- `source` [449](#)
- `SPACE` [302](#)
- `spacing` [450](#)
- `specified` [451](#)
- `spinbox` [328](#)
- `splitbox` [328](#)
- `Sprachvariante` [242](#)
- `startsel` [452](#)
- `starttime` [454](#)
- `state` [455](#)
- `state_checked` [455](#)
- `state_indeterminate` [455](#)
- `state_unchecked` [455](#)
- `static` [456](#)
- `statictext` [329](#)
- `statusbar` [457](#)
- `statushelp` [458](#)
- `Statuszeile` [480](#)
- `Stil`
 - `font` [465](#)
- `style` [459](#)
- `style[enum]` [467](#)

- `style_buttons` [467](#)
- `style_continuous` [467](#)
- `style_labeled` [467](#)
- `style_lines` [467](#)
- `style_root` [467](#)
- `subcontrol[integer]` [469](#)
- `subcontrolcount` [470](#)
- `sysmodal` [471](#)
- `systemerror` [472](#)
- `Systemformat` [195](#)
- `systemid` [473](#)

T

- `tab_major` [476](#)
- `tab_minor` [476](#)
- `tabalignment` [474](#)
- `tablefield` [329](#)
 - `Eingabeverhalten` [368](#)
 - `explizite Aktivierung` [368](#)
 - `implizite Aktivierung` [368](#)
- `tabshape` [475](#)
- `tabtype` [476](#)
- `target` [477](#)
- `Tastatur-Cursor` [302](#)
- `Tastaturfokus` [337](#)
- `Tastaturnavigation` [337](#)
- `tcpip` [514](#)
- `terminal` [478](#)
- `terminaltype` [479](#)
- `text` [480](#)
- `Text` [492](#)
 - `Ausrichtung` [38](#)

text[enum] 481
text[integer] 483
textbgc 484
textfgc 485
Textknoten 480
Textvariante 242
textwidth 486
thisevent 29, 109
tile 487
tile_active 353
tile_active_mouse_over 353
tile_default 353
tile_insensitive 353
tile_mouse_over 353
.tiledpi 488
tilestyle 489
tilestyle_background 490
tilestyle_bottom 490
tilestyle_centered 489
tilestyle_icon 490
tilestyle_left 490
tilestyle_parent_tile 489
tilestyle_right 490
tilestyle_stretched 489
tilestyle_tiled 489
tilestyle_top 490
timeout 491
Timeout 491
timer 109
Timer 491
Titelleiste 493
title 492
titlebar 493
titlebgc 494
titlefgc 495
today 496
todaymarker 497
toolbar 332, 498
toolbar[integer] 499
toolbarcount 500
toolhelp 501
Toolhelp 317-323, 325-329, 332, 334
toolkit 503
toolkit-Enumeration 379, 528
toolkit_motif 503
toolkit_qt 503
toolkit_string 504
toolkit_version 505
toolkit_windows 379, 503, 528
top_most 506
topitem 507
Toplevel-Fenster 133
Trace-Code 511
tracefile 508
Tracefile 511
tracetime 509
tracing 511
tracing[string] 511
trailingdates 512
transformer[integer] 513
transport 514
 SSL 514
treeview 332
type 515

typescope [516](#)
typescope[integer] [517](#)
Typkonvertierung [239](#)

U

Umgebungsvariable [155-156](#)
Unterknoten [480](#)
userdata [518](#)
userdata[index] [520](#)
userdata[integer] [519](#)
.username [521](#)
userplaced [522](#)
USW-Klasse [166-167](#)
uuid [523](#)

V

value [524](#)
value[integer] [527](#)
Variable [515](#)
Vektorenlänge [109](#)
version[enum] [528](#)
version_string [530](#)
vheight [531](#)
visible [431](#), [532](#)
Vordergrundfarbe [223](#)
Vorlagenobjekt [292](#)
vsb_arrows [533](#)
vsb_linemotion [534](#)
vsb_optional [535](#)
vsb_pagemotion [536](#)
vsb_visible [537](#)
.vscreen_height [538](#)

.vscreen_width [539](#)
.vscreen_x [540](#)
.vscreen_y [541](#)
vwidth [542](#)

W

weeknumbers [543](#)
.weight [544](#)
Widget [337](#)
width[class] [546](#)
.width[enum] [546](#)
window [333](#), [548](#)
Window
 schließen [334](#)
winsys [549](#)
winsys_none [549](#)
winsys_string [550](#)
winsys_version [551](#)
winsys_windows [549](#)
winsys_x11 [549](#)
wrap [552](#)

X

x [553](#)
xalignment[index] [554](#)
xauto [555](#)
xdpi [556](#)
xdpi[integer] [557](#)
xleft [558](#)
xml [561](#)
XML-Toolkit [379](#), [528](#)
xorigin [20](#), [562](#)

xraster [391](#), [563](#), [576](#)

xright [565](#)

xspacing [566](#)

Y

y [567](#)

yalignment[index] [568](#)

yauto [569](#)

ybottom [570](#)

ydpi [571](#)

ydpi[integer] [572](#)

yorigin [20](#), [575](#)

yraster [391](#), [563](#), [576](#)

yspacing [578](#)

ytop [579](#)

Z

zusätzliche Geometrieattribute [20](#)