OS-9® Device Descriptor and Configuration Module Reference

Version 4.7
Copyright and publication information

This manual reflects version 4.7 of Microware OS-9. Reproduction of this document, in part or whole, by any means, electrical, mechanical, magnetic, optical, chemical, manual, or otherwise is prohibited, without written permission from RadiSys Microware Communications Software Division, Inc.

Disclaimer

The information contained herein is believed to be accurate as of the date of publication. However, RadiSys Corporation will not be liable for any damages including indirect or consequential, from use of the OS-9 operating system, Microware-provided software, or reliance on the accuracy of this documentation. The information contained herein is subject to change without notice.

Reproduction notice

The software described in this document is intended to be used on a single computer system. RadiSys Corporation expressly prohibits any reproduction of the software on tape, disk, or any other medium except for backup purposes. Distribution of this software, in part or whole, to any other party or on any other system may constitute copyright infringements and misappropriation of trade secrets and confidential processes which are the property of RadiSys Corporation and/or other parties. Unauthorized distribution of software may cause damages far in excess of the value of the copies involved.
# Contents

Low-Level System Configuration Module (cnfgdata) 15

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>16</td>
</tr>
<tr>
<td>cnfgdata Module Field Configuration Options</td>
<td>16</td>
</tr>
<tr>
<td>Direct Modification Advantages</td>
<td>16</td>
</tr>
<tr>
<td>Description File/Rebuild Advantages</td>
<td>16</td>
</tr>
<tr>
<td>Direct Modification</td>
<td>17</td>
</tr>
<tr>
<td>Description File Modification</td>
<td>19</td>
</tr>
<tr>
<td>Low-Level Configuration Module Field Reference</td>
<td>19</td>
</tr>
<tr>
<td>Module Header Fields</td>
<td>20</td>
</tr>
<tr>
<td>_m_group</td>
<td>22</td>
</tr>
<tr>
<td>_m_user</td>
<td>23</td>
</tr>
<tr>
<td>mod_name</td>
<td>24</td>
</tr>
<tr>
<td>m_access</td>
<td>25</td>
</tr>
<tr>
<td>m_tylan</td>
<td>27</td>
</tr>
<tr>
<td>m_attrev</td>
<td>29</td>
</tr>
<tr>
<td>m_edit</td>
<td>31</td>
</tr>
<tr>
<td>Console Device Fields</td>
<td>31</td>
</tr>
<tr>
<td>console_name</td>
<td>33</td>
</tr>
<tr>
<td>cons_vector</td>
<td>34</td>
</tr>
<tr>
<td>cons_priority</td>
<td>35</td>
</tr>
<tr>
<td>cons_level</td>
<td>36</td>
</tr>
<tr>
<td>cons_timeout</td>
<td>37</td>
</tr>
<tr>
<td>cons_parity</td>
<td>38</td>
</tr>
<tr>
<td>cons_baudrate</td>
<td>39</td>
</tr>
<tr>
<td>cons_wordsize</td>
<td>41</td>
</tr>
<tr>
<td>cons_stopbits</td>
<td>42</td>
</tr>
<tr>
<td>cons_flow</td>
<td>43</td>
</tr>
<tr>
<td>Communication Device Fields</td>
<td>43</td>
</tr>
<tr>
<td>comm_name</td>
<td>45</td>
</tr>
<tr>
<td>cons_vector</td>
<td>46</td>
</tr>
<tr>
<td>cons_priority</td>
<td>47</td>
</tr>
<tr>
<td>cons_level</td>
<td>48</td>
</tr>
<tr>
<td>cons_timeout</td>
<td>49</td>
</tr>
<tr>
<td>cons_parity</td>
<td>50</td>
</tr>
<tr>
<td>cons_baudrate</td>
<td>51</td>
</tr>
<tr>
<td>cons_wordsize</td>
<td>53</td>
</tr>
<tr>
<td>cons_stopbits</td>
<td>54</td>
</tr>
<tr>
<td>cons_flow</td>
<td>55</td>
</tr>
<tr>
<td>Debugger Fields</td>
<td>55</td>
</tr>
<tr>
<td>debug_name</td>
<td>57</td>
</tr>
<tr>
<td>debug_call_at_cold</td>
<td>58</td>
</tr>
<tr>
<td>Low-Level Protocol Manager Fields</td>
<td>58</td>
</tr>
</tbody>
</table>
OS-9® Device Descriptor and Configuration Module Reference

Init Module Field Reference ................................................................................................................................................................................. 92

Boot Data Fields........................................................................................................................................................................................................ 79

Configuration Boot Data Fields................................................................................................................................................................. 76

Interface Data Fields........................................................................................................................................................................................................ 63

Description File/Rebuild Advantages ........................................................................................................................................................ 88

Direct Modification Advantages ................................................................................................................................................................. 88

Boot Data Fields........................................................................................................................................................................................................ 79

Notification Services Field................................................................................................................................................................................................ 85

OS-9 Configuration Module (init) 87

Init Module Field Configuration Options.................................................................................................................................................... 88

Direct Modification Advantages ................................................................................................................................................................. 88

Description File/Rebuild Advantages ........................................................................................................................................................ 88

Direct Modification................................................................................................................................................................................................ 88

Description File Modification........................................................................................................................................................................... 91

Module Header Fields................................................................................................................................................................................................ 93

Module Body Fields................................................................................................................................................................................................ 103

maxlpmprotos ........................................................................................................................................................................................................ 60
maxrcmbufs ........................................................................................................................................................................................................ 61
maxlpmcns ......................................................................................................................................................................................................... 62
lpm_count ......................................................................................................................................................................................................... 63
Interface Data Fields........................................................................................................................................................................................................ 63
ip_address ........................................................................................................................................................................................................ 65
subnet_mask ......................................................................................................................................................................................................... 66
brdcst_address ...................................................................................................................................................................................................... 67
gw_address ........................................................................................................................................................................................................ 68
mac_address ......................................................................................................................................................................................................... 69
hwtype ................................................................................................................................................................................................................ 70
if_flags ................................................................................................................................................................................................................ 71
if_name ................................................................................................................................................................................................................ 72
port_address ......................................................................................................................................................................................................... 73
if_vector ............................................................................................................................................................................................................... 74
if_priority ........................................................................................................................................................................................................ 75
if_level ............................................................................................................................................................................................................... 76
boot_count ........................................................................................................................................................................................................ 76
boot_cmdsize ...................................................................................................................................................................................................... 78
boot_abname ........................................................................................................................................................................................................ 80
boot_newab ........................................................................................................................................................................................................ 81
boot_newname ...................................................................................................................................................................................................... 82
boot_automenu .................................................................................................................................................................................................... 83
boot_params ........................................................................................................................................................................................................ 84
autoboot_delay ................................................................................................................................................................................................... 85
max_notifiers .......................................................................................................................................................................................................... 86
maxlpmprotos ........................................................................................................................................................................................................ 60
maxrcmbufs ........................................................................................................................................................................................................ 61
maxlpmcns ......................................................................................................................................................................................................... 62
lpm_count ......................................................................................................................................................................................................... 63
Interface Data Fields........................................................................................................................................................................................................ 63
ip_address ........................................................................................................................................................................................................ 65
subnet_mask ......................................................................................................................................................................................................... 66
brdcst_address ...................................................................................................................................................................................................... 67
gw_address ........................................................................................................................................................................................................ 68
mac_address ......................................................................................................................................................................................................... 69
hwtype ................................................................................................................................................................................................................ 70
if_flags ................................................................................................................................................................................................................ 71
if_name ................................................................................................................................................................................................................ 72
port_address ......................................................................................................................................................................................................... 73
if_vector ............................................................................................................................................................................................................... 74
if_priority ........................................................................................................................................................................................................ 75
if_level ............................................................................................................................................................................................................... 76
boot_count ........................................................................................................................................................................................................ 76
boot_cmdsize ...................................................................................................................................................................................................... 78
boot_abname ........................................................................................................................................................................................................ 80
boot_newab ........................................................................................................................................................................................................ 81
boot_newname ...................................................................................................................................................................................................... 82
boot_automenu .................................................................................................................................................................................................... 83
boot_params ........................................................................................................................................................................................................ 84
autoboot_delay ................................................................................................................................................................................................... 85
max_notifiers .......................................................................................................................................................................................................... 86

OS-9® Device Descriptor and Configuration Module Reference

maxlpmprotos ........................................................................................................................................................................................................ 60
maxrcmbufs ........................................................................................................................................................................................................ 61
maxlpmcns ......................................................................................................................................................................................................... 62
lpm_count ......................................................................................................................................................................................................... 63
Interface Data Fields........................................................................................................................................................................................................ 63
ip_address ........................................................................................................................................................................................................ 65
subnet_mask ......................................................................................................................................................................................................... 66
brdcst_address ...................................................................................................................................................................................................... 67
gw_address ........................................................................................................................................................................................................ 68
mac_address ......................................................................................................................................................................................................... 69
hwtype ................................................................................................................................................................................................................ 70
if_flags ................................................................................................................................................................................................................ 71
if_name ................................................................................................................................................................................................................ 72
port_address ......................................................................................................................................................................................................... 73
if_vector ............................................................................................................................................................................................................... 74
if_priority ........................................................................................................................................................................................................ 75
if_level ............................................................................................................................................................................................................... 76
boot_count ........................................................................................................................................................................................................ 76
boot_cmdsize ...................................................................................................................................................................................................... 78
boot_abname ........................................................................................................................................................................................................ 80
boot_newab ........................................................................................................................................................................................................ 81
boot_newname ...................................................................................................................................................................................................... 82
boot_automenu .................................................................................................................................................................................................... 83
boot_params ........................................................................................................................................................................................................ 84
autoboot_delay ................................................................................................................................................................................................... 85
max_notifiers .......................................................................................................................................................................................................... 86

OS-9® Device Descriptor and Configuration Module Reference

maxlpmprotos ........................................................................................................................................................................................................ 60
maxrcmbufs ........................................................................................................................................................................................................ 61
maxlpmcns ......................................................................................................................................................................................................... 62
lpm_count ......................................................................................................................................................................................................... 63
Interface Data Fields........................................................................................................................................................................................................ 63
ip_address ........................................................................................................................................................................................................ 65
subnet_mask ......................................................................................................................................................................................................... 66
brdcst_address ...................................................................................................................................................................................................... 67
gw_address ........................................................................................................................................................................................................ 68
mac_address ......................................................................................................................................................................................................... 69
hwtype ................................................................................................................................................................................................................ 70
if_flags ................................................................................................................................................................................................................ 71
if_name ................................................................................................................................................................................................................ 72
port_address ......................................................................................................................................................................................................... 73
if_vector ............................................................................................................................................................................................................... 74
if_priority ........................................................................................................................................................................................................ 75
if_level ............................................................................................................................................................................................................... 76
boot_count ........................................................................................................................................................................................................ 76
boot_cmdsize ...................................................................................................................................................................................................... 78
boot_abname ........................................................................................................................................................................................................ 80
boot_newab ........................................................................................................................................................................................................ 81
boot_newname ...................................................................................................................................................................................................... 82
boot_automenu .................................................................................................................................................................................................... 83
boot_params ........................................................................................................................................................................................................ 84
autoboot_delay ................................................................................................................................................................................................... 85
max_notifiers .......................................................................................................................................................................................................... 86

OS-9® Device Descriptor and Configuration Module Reference

maxlpmprotos ........................................................................................................................................................................................................ 60
maxrcmbufs ........................................................................................................................................................................................................ 61
maxlpmcns ......................................................................................................................................................................................................... 62
lpm_count ......................................................................................................................................................................................................... 63
Interface Data Fields........................................................................................................................................................................................................ 63
ip_address ........................................................................................................................................................................................................ 65
subnet_mask ......................................................................................................................................................................................................... 66
brdcst_address ...................................................................................................................................................................................................... 67
gw_address ........................................................................................................................................................................................................ 68
mac_address ......................................................................................................................................................................................................... 69
hwtype ................................................................................................................................................................................................................ 70
if_flags ................................................................................................................................................................................................................ 71
if_name ................................................................................................................................................................................................................ 72
port_address ......................................................................................................................................................................................................... 73
if_vector ............................................................................................................................................................................................................... 74
if_priority ........................................................................................................................................................................................................ 75
if_level ............................................................................................................................................................................................................... 76
boot_count ........................................................................................................................................................................................................ 76
boot_cmdsize ...................................................................................................................................................................................................... 78
boot_abname ........................................................................................................................................................................................................ 80
boot_newab ........................................................................................................................................................................................................ 81
boot_newname ...................................................................................................................................................................................................... 82
boot_automenu .................................................................................................................................................................................................... 83
boot_params ........................................................................................................................................................................................................ 84
autoboot_delay ................................................................................................................................................................................................... 85
max_notifiers .......................................................................................................................................................................................................... 86
SCF Device Descriptors 149

SCF Field Configuration Options .............................................................................................................. 150
  Direct Modification Advantages .................................................................................................................. 150
  Description File/Rebuild Advantages .......................................................................................................... 150
  Direct Modification .................................................................................................................................. 150
  Description File Modification ...................................................................................................................... 153

SCF Device Descriptor Field Reference ........................................................................................................ 154

Module Header Fields .................................................................................................................................. 154
  _m_group .................................................................................................................................................. 156
  _m_user .................................................................................................................................................. 157
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>mod_name</td>
<td>158</td>
</tr>
<tr>
<td>m_access</td>
<td>159</td>
</tr>
<tr>
<td>m_tylan</td>
<td>161</td>
</tr>
<tr>
<td>m_attrev</td>
<td>163</td>
</tr>
<tr>
<td>m_edit</td>
<td>165</td>
</tr>
<tr>
<td>Device Descriptor Data Definition Fields</td>
<td>165</td>
</tr>
<tr>
<td>dd_port</td>
<td>167</td>
</tr>
<tr>
<td>dd_lun</td>
<td>168</td>
</tr>
<tr>
<td>dd_pd_size</td>
<td>169</td>
</tr>
<tr>
<td>dd_type</td>
<td>170</td>
</tr>
<tr>
<td>dd_mode</td>
<td>172</td>
</tr>
<tr>
<td>fmgr_name</td>
<td>174</td>
</tr>
<tr>
<td>drv_name</td>
<td>175</td>
</tr>
<tr>
<td>dd_class</td>
<td>176</td>
</tr>
<tr>
<td>SCF Description Block Fields</td>
<td>176</td>
</tr>
<tr>
<td>outdev_name</td>
<td>178</td>
</tr>
<tr>
<td>SCF Logical Unit Static Storage Fields</td>
<td>178</td>
</tr>
<tr>
<td>hardware_vector</td>
<td>180</td>
</tr>
<tr>
<td>v_irqlevel</td>
<td>181</td>
</tr>
<tr>
<td>v_priority</td>
<td>182</td>
</tr>
<tr>
<td>v_pollin</td>
<td>183</td>
</tr>
<tr>
<td>v_pollout</td>
<td>184</td>
</tr>
<tr>
<td>v_lun</td>
<td>185</td>
</tr>
<tr>
<td>v_irqmask</td>
<td>186</td>
</tr>
<tr>
<td>v_maxbuff</td>
<td>187</td>
</tr>
<tr>
<td>v_insize</td>
<td>188</td>
</tr>
<tr>
<td>v_outsize</td>
<td>189</td>
</tr>
<tr>
<td>v_line</td>
<td>190</td>
</tr>
<tr>
<td>v_intr</td>
<td>191</td>
</tr>
<tr>
<td>v_quit</td>
<td>194</td>
</tr>
<tr>
<td>v_psch</td>
<td>195</td>
</tr>
<tr>
<td>v_xon</td>
<td>196</td>
</tr>
<tr>
<td>v_xoff</td>
<td>197</td>
</tr>
<tr>
<td>v_baud</td>
<td>198</td>
</tr>
<tr>
<td>v_parity</td>
<td>200</td>
</tr>
<tr>
<td>v_stopbits</td>
<td>201</td>
</tr>
<tr>
<td>v_wordsize</td>
<td>202</td>
</tr>
<tr>
<td>v_rtsstate</td>
<td>203</td>
</tr>
<tr>
<td>v_devspec</td>
<td>204</td>
</tr>
<tr>
<td>SCF Path Option Fields</td>
<td>204</td>
</tr>
<tr>
<td>pd_inmap0type</td>
<td>211</td>
</tr>
<tr>
<td>pd_inmap0func_code</td>
<td>212</td>
</tr>
<tr>
<td>pd_inmap0size</td>
<td>214</td>
</tr>
<tr>
<td>pd_inmap0string</td>
<td>215</td>
</tr>
<tr>
<td>pd_inmap1type</td>
<td>216</td>
</tr>
<tr>
<td>pd_inmap1func_code</td>
<td>217</td>
</tr>
<tr>
<td>pd_inmap1size</td>
<td>218</td>
</tr>
<tr>
<td>pd_inmap1string</td>
<td>219</td>
</tr>
<tr>
<td>pd_inmap2type</td>
<td>220</td>
</tr>
</tbody>
</table>
pd_inmap14size ........................................................................................................................................................................................... 270
pd_inmap14string ........................................................................................................................................................................................... 271
pd_inmap15type .......................................................................................................................................................................................... 272
pd_inmap15func_code ................................................................................................................................................................................... 273
pd_inmap15size ........................................................................................................................................................................................... 274
pd_inmap15string ........................................................................................................................................................................................... 275
pd_inmap16type .......................................................................................................................................................................................... 276
pd_inmap16func_code ................................................................................................................................................................................... 277
pd_inmap16size ........................................................................................................................................................................................... 278
pd_inmap16string ........................................................................................................................................................................................... 279
pd_inmap17type .......................................................................................................................................................................................... 280
pd_inmap17func_code ................................................................................................................................................................................... 281
pd_inmap17size ........................................................................................................................................................................................... 282
pd_inmap17string ........................................................................................................................................................................................... 283
pd_inmap18type .......................................................................................................................................................................................... 284
pd_inmap18func_code ................................................................................................................................................................................... 285
pd_inmap18size ........................................................................................................................................................................................... 286
pd_inmap18string ........................................................................................................................................................................................... 287
pd_inmap19type .......................................................................................................................................................................................... 288
pd_inmap19func_code ................................................................................................................................................................................... 289
pd_inmap19size ........................................................................................................................................................................................... 290
pd_inmap19string ........................................................................................................................................................................................... 291
pd_inmap20type .......................................................................................................................................................................................... 292
pd_inmap20func_code ................................................................................................................................................................................... 293
pd_inmap20size ........................................................................................................................................................................................... 294
pd_inmap20string ........................................................................................................................................................................................... 295
pd_inmap21type .......................................................................................................................................................................................... 296
pd_inmap21func_code ................................................................................................................................................................................... 297
pd_inmap21size ........................................................................................................................................................................................... 298
pd_inmap21string ........................................................................................................................................................................................... 299
pd_inmap22type .......................................................................................................................................................................................... 300
pd_inmap22func_code ................................................................................................................................................................................... 301
pd_inmap22size ........................................................................................................................................................................................... 302
pd_inmap22string ........................................................................................................................................................................................... 303
pd_inmap23type .......................................................................................................................................................................................... 304
pd_inmap23func_code ................................................................................................................................................................................... 305
pd_inmap23size ........................................................................................................................................................................................... 306
pd_inmap23string ........................................................................................................................................................................................... 307
pd_inmap24type .......................................................................................................................................................................................... 308
pd_inmap24func_code ................................................................................................................................................................................... 309
pd_inmap24size ........................................................................................................................................................................................... 310
pd_inmap24string ........................................................................................................................................................................................... 311
pd_inmap25type .......................................................................................................................................................................................... 312
pd_inmap25func_code ................................................................................................................................................................................... 313
pd_inmap25size ........................................................................................................................................................................................... 314
pd_inmap25string ........................................................................................................................................................................................... 315
pd_inmap26type .......................................................................................................................................................................................... 316
pd_inmap26func_code ................................................................................................................................................................................... 317
pd_inmap26size ........................................................................................................................................................................................... 318
SBF Device Descriptors 357

SBF Field Configuration Options ................................................................. 358
Direct Modification Advantages ................................................................. 358
Description File/Rebuild Advantages ....................................................... 358
Direct Modification .................................................................................. 358
Description File Configuration ................................................................. 361

SBF Device Descriptor Field Reference .................................................. 361
Module Header Fields ............................................................................ 362
_m_group ................................................................................................. 363
_m_user ................................................................................................. 364
mod_name ............................................................................................. 365
m_access .............................................................................................. 366
m_tylan ................................................................................................. 368
m_attrev ........................................................................................................................................................................... 370
m_edit ........................................................................................................................................................................................................ 372
Device Descriptor Data Definition Fields................................................................................................................................ 372
dd_port .................................................................................................................................................................................................. 374
dd_lun .................................................................................................................................................................................................. 375
dd_pd_size .................................................................................................................................................................................................. 376
dd_type .................................................................................................................................................................................................. 377
dd_mode .................................................................................................................................................................................................. 379
fmgr_name .................................................................................................................................................................................................. 381
drv_name .................................................................................................................................................................................................. 382
dd_class .................................................................................................................................................................................................. 383
SBF Path Options Fields........................................................................................................................................................................ 383
pd_bkksz .................................................................................................................................................................................................. 385
pd_flags .................................................................................................................................................................................................. 386
pd_dmmode .................................................................................................................................................................................................. 387
pd_sci_id .................................................................................................................................................................................................. 388
pd_scslun .................................................................................................................................................................................................. 389
SBF Logical Unit Status Fields.......................................................................................................................................................... 389
sbf_vector .................................................................................................................................................................................................. 390
sbf_irqlevel .................................................................................................................................................................................................. 391
sbf_priority .................................................................................................................................................................................................. 392
sbf_dflag .................................................................................................................................................................................................. 393
RBF Device Descriptors 395
RBF Field Configuration Options.................................................................................................................................................... 396
Direct Modification Advantages........................................................................................................................................................ 396
Description File/Rebuild Advantages................................................................................................................................................ 396
Direct Modification.................................................................................................................................................................................................. 396
Description File Configuration.......................................................................................................................................................... 399
RBF Device Descriptor Field Reference............................................................................................................................................. 399
Module Header Fields........................................................................................................................................................................... 400
_m_group .................................................................................................................................................................................................. 401
_m_user .................................................................................................................................................................................................. 402
mod_name .................................................................................................................................................................................................. 403
m_access .................................................................................................................................................................................................. 404
m_tylan .................................................................................................................................................................................................. 406
m_attrev .................................................................................................................................................................................................. 408
m_edit .................................................................................................................................................................................................. 410
Device Descriptor Data Definition Fields........................................................................................................................................... 410
dd_port .................................................................................................................................................................................................. 412
dd_lun .................................................................................................................................................................................................. 413
dd_pd_size .................................................................................................................................................................................................. 414
dd_type .................................................................................................................................................................................................. 415
dd_mode .................................................................................................................................................................................................. 417
fmgr_name .................................................................................................................................................................................................. 419
drv_name .................................................................................................................................................................................................. 420
dd_class .................................................................................................................................................................................................. 421
RBF Path Option Fields........................................................................................................................................................................... 421
pd_sid .................................................................................................................................................................................................. 423
pd_vfy .................................................................................................................................................................................................. 424
PCF Device Descriptors 451

PCF Field Configuration Options .......................................................................................................................... 452
  Direct Modification Advantages .......................................................................................................................... 452
  Description File/Rebuild Advantages .................................................................................................................. 452
  Direct Modification ............................................................................................................................................. 452
  Description File Configuration ........................................................................................................................... 455

PCF Device Descriptor Field Reference .................................................................................................................. 455

Module Header Fields ................................................................................................................................................. 456
  _m_group ........................................................................................................................................................... 457
  _m_user ............................................................................................................................................................ 458
  mod_name ........................................................................................................................................................ 459
  m_access .......................................................................................................................................................... 460
  m_tylan ........................................................................................................................................................... 462
  m_attrev ........................................................................................................................................................... 464
  m_edit ............................................................................................................................................................. 466

Device Descriptor Data Definition Fields .................................................................................................................. 466
  dd_port ............................................................................................................................................................ 468
  dd_lun ............................................................................................................................................................. 469
  dd_pd_size .................................................................................................................................................... 470
  dd_type ......................................................................................................................................................... 471
  dd_mode ....................................................................................................................................................... 473
  fmgr_name ................................................................................................................................................... 475
  drvr_name .................................................................................................................................................... 476
Pipe Device Descriptors 507

Pipe Device Descriptor Field Configuration Options ......................................................... 508

Direct Modification Advantages ......................................................................................... 508

Description File/Rebuild Advantages .................................................................................. 508

Direct Modification .............................................................................................................. 508

Description File Modification ............................................................................................ 510

Pipe Device Descriptor Field Reference ............................................................................. 511

Module Header Fields ......................................................................................................... 512

_m_group .......................................................................................................................... 513

_m_user ............................................................................................................................. 514

mod_name ........................................................................................................................ 515

m_access .......................................................................................................................... 516

m_tylan ............................................................................................................................. 518

m_attrrev ........................................................................................................................ 520

m_edit ............................................................................................................................... 522

Device Descriptor Data Definition Fields ............................................................................ 522

dd_port ............................................................................................................................. 524

dd_lun ............................................................................................................................... 525

dd_pd_size ....................................................................................................................... 526
Low-Level System Configuration Module (cnfgdata)

This chapter includes the following topics:

- Overview
- cnfgdata Module Field Configuration Options
- Low-Level Configuration Module Field Reference
  - Module Header Fields
  - Console Device Fields
  - Communication Device Fields
  - Debugger Fields
  - Low-Level Protocol Manager Fields
  - Interface Data Fields
  - Configuration Boot Data Fields
  - Boot Data Fields
  - Notification Services Field
Overview

The `cnfgdata` module contains configuration data used by the low-level system modules. The following subsystems are configured in the `cnfgdata` module:

- Low-level system console
- Low-level auxiliary communication
- Debugger
- Low-level protocol manager and interface data
- Booters and boot services
- Notification services

The next section in this chapter provides a detailed example of the configuration options you can use to change configuration values for this module.

The rest of this chapter provides a detailed list of all available `cnfgdata` module fields, including a field description and available values.

`cnfgdata` Module Field Configuration Options

There are two methods you can use to change a `cnfgdata` module configuration field:

1. Use the `EditMod` utility to directly modify existing `cnfgdata` modules either as a stand-alone module or as part of a merged module group (such as a boot image).
2. Modify the description file for the `cnfgdata` module and rebuild it using the makefile provided.

Direct Modification Advantages

The direct modification method has the following advantages:

- **Fast**: No source configuration file rebuilds are necessary.
- **Temporary**: The original module or merged-module group configuration can be easily restored via the appropriate rebuild.
- **Contained**: Changes are limited to the individual boot image modified (merged-module option).

Description File/Rebuild Advantages

The advantage of the description file/rebuild method is the changes are permanent and reproducible. Modifications apply to all subsequent module rebuilds and to all merged-module groups built containing the updated module.

Both methods are documented in this section. These procedures are used with the field descriptions starting with the Module Header Fields. For direct modification, use the `EditMod` LABELS data to navigate the `EditMod` menus. The DESCRIPTION FILE MACRO data identifies the macro you need to define/modify in the configuration sources to rebuild the `cnfgdata` module.
Direct Modification

Use the `EditMod` utility and the following procedures to directly modify fields in the existing `cnfgdata` module. The module can stand-alone or it can be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The field references later in this chapter contain a description of each configurable field, its supported values, and the sequence of menu options required by `EditMod` to modify that field.

Refer to the *Utilities Reference* for a full description of `EditMod`'s capabilities.

![Figure 1-1. Directory Location for Modifying the cnfgdata Module as a Stand-alone Module](image1)

![Figure 1-2. Directory Location for Modifying the cnfgdata Module as Part of a Boot Image](image2)

Refer to your board guide for information about how to modify the module lists and remake the boot images, and for specific boot image names.

**Direct Modification Procedures**

To modify the stand-alone module, complete the following steps:

1. Change to the `CMDS/BOOTOBJ/ROM` directory (see **Figure 1-1**).
2. Use `EditMod` to edit the module:
$EditMod -e -dc_all cnfgdata

To modify the module as part of a merged module group, complete the following steps:

1. Change to the BOOTS/SYSTEMS/PORTBOOT directory (see Figure 1-2).
2. Use EditMod to edit the module:

   $EditMod -e -dc_all cnfgdata -f=<boot image name>

3. Use the menu selections provided in the EditMod LABELS section of the field reference later in this chapter to locate the fields you want to edit.
4. Select a new value for the field from the AVAILABLE VALUES section of the field reference. Enter that value at the EditMod prompt to modify the field.
5. If you want to make additional modifications, use the p command (previous) to step backward through the EditMod menus. Repeat Steps 3 and 4 until you have made all desired modifications to the cnfgdata module.
6. Select the w command (write) to save the changes.
7. Select the q command (quit) to exit EditMod.

Unless you modified the cnfgdata module in your boot image, you should rebuild your boot image to include the new cnfgdata module.

Example EditMod Session

This example modifies cnfgdata as part of the boot image rom.

$ EditMod -e -dc_all cnfgdata -f=rom

1. Module header
2. Configuration data

$Which? [?/1-2/p/t/a/w/q] 2

1. Console port data structure
2. Communication port data structure
3. Debugger data structure
4. Low level protocol manager data structure
5. Boot services data structure
6. Notification services data structure

$Which? [?/1-6/p/t/a/w/q] .
. (desired modifications)
.
Which? [?/1-19/p/t/a/w/q] w

Which? [?/1-19/p/t/a/w/q] q
Description File Modification

You can use these procedures to modify the \texttt{cnfgdata} description file and rebuild the \texttt{cnfgdata} modules for your port directory. The DESCRIPTION FILE MACROS section of the field reference specifies the name of the macro you modify/define in the description file to configure the field. The value used in the define is chosen from the AVAILABLE VALUES specified for the field.

**Figure 1-3. Directory Location for Modifying the cnfgdata Description Files**

![Diagram showing the directory structure for modifying the cnfgdata description files]

**Description File Modification Procedures**

1. Change to the \texttt{ROM/CNFGDATA} directory (see Figure 1-3).
2. Edit the file \texttt{config.des} and read the included comments for more information on using the specific description file provided in your software distribution. The \texttt{config.des} file contains a list of macro names which can be defined to override the global default values for the configuration fields.
3. Refer to the DESCRIPTION FILE MACRO section in the field reference later in this chapter to determine the macro name you define to configure the target field.
4. Read the comments in \texttt{config.des} to determine where to place the define for this macro.
5. Select the value you want to use to configure the field. See the AVAILABLE VALUES section of the field reference data for values or macros that can be used for the definition. Define the macro by entering a definition in the appropriate description files as follows:
   
   ```
   \#define <macro> <value>
   ```
6. Save the changes and rebuild the module by entering the following command from the \texttt{ROM/CNFGDATA} directory:

   ```
   os9make
   ```
7. Rebuild your boot image to include the new \texttt{cnfgdata} module.

**Low-Level Configuration Module Field Reference**

This section contains a list of all configurable fields in the \texttt{cnfgdata} module. Each field entry contains the following information:
•  `<Field name>` - The call name for each field that can be reconfigured in the module.

•  EditMod LABELS - EditMod menu selections for navigating to the proper field in an EditMod session.

•  DESCRIPTION FILE MACRO - The macro name you modify/define in the description file.

•  DESCRIPTION - A brief description of the field's purpose and use.

•  EXAMPLE - An optional example of the description file entry showing how to change the value of this field.

•  PORT GENERIC DEFAULT VALUE - The value set in the port generic description file for this field. This is the value the field is assigned when the module is built, unless the appropriate macro has been defined in the port specific description file to override this default value.

•  PORT SPECIFIC OVERRIDE VALUE - The value set in the port specific description file for this field. If defined, this is the value the field is assigned when the module is built, overriding the port generic default value.

•  AVAILABLE VALUES - Values to which the field can be set through EditMod or the description files. In many cases, this data is presented in a table that maps a description of the value to a numeric value appropriate for entry in EditMod, and to a pre-defined macro available for use in the description file.

The cnfgdata module consists of a module header and six distinct sections of configuration data. Each section is used by a specific low-level sub-system. The reference data in this chapter is divided into sections based on sub-system.

### Module Header Fields

The following section contains the module header fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>_m_group</td>
<td>MH_GROUP</td>
</tr>
<tr>
<td>_m_user</td>
<td>MH_USER</td>
</tr>
<tr>
<td>mod_name</td>
<td>MH_NAME</td>
</tr>
<tr>
<td>m_access</td>
<td>MH_ACCESS</td>
</tr>
<tr>
<td>m_tylan</td>
<td>MH_TYLAN</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------</td>
</tr>
<tr>
<td>m_attrev</td>
<td>MH_ATTREV</td>
</tr>
<tr>
<td>m_edit</td>
<td>MH_EDIT</td>
</tr>
</tbody>
</table>
EditMod Labels
1-module header
1-module owner’s group number

Description
Group ID of the module’s owner. The group number allows people working in the same department or on the same project to share a common identification number.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 65535
EditMod Labels

1-module header
2-module owner’s user number

Description

User ID of the module’s owner. The user number identifies a specific user.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values

0 to 65535
**EditMod Labels**

1-module header
3-module name

**Description**
Contains the module name string.

**Port Generic Default Value**

NULL

**Port Specific Override Value**

Refer to `ROM/CNFGDATA/config.des` (Figure 1-3).

**Available Values**

Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
Chapter 1: Low-Level System Configuration Module (cnfgdata)

m_access
MH_ACCESS

EditMod Labels

1-module header
4-access permissions

Description

Defines the permissible module access by its owner or by other users.

Port Generic Default Value

Macro

MP_OWNER_READ | MP_OWNER_EXEC | MP_GROUP_READ |
MP_GROUP_EXEC | MP_WORLD_READ | MP_WORLD_EXEC

EditMod
0x555

Port Specific Override Value

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values

Module access permission values are located in the header file, module.h, and are listed in Table 1-2.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read permission by owner</td>
<td>MP_OWNER_READ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Write permission by owner</td>
<td>MP_OWNER_WRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Execute permission by owner</td>
<td>MP_OWNER_EXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Owner permission mask</td>
<td>MP_OWNER_MASK</td>
<td>0x000f</td>
</tr>
<tr>
<td>Read permission by group</td>
<td>MP_GROUP_READ</td>
<td>0x0010</td>
</tr>
<tr>
<td>Write permission by group</td>
<td>MP_GROUP_WRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Execute permission by group</td>
<td>MP_GROUP_EXEC</td>
<td>0x0040</td>
</tr>
<tr>
<td>Group permission mask</td>
<td>MP_GROUP_MASK</td>
<td>0x00f0</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>--------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Read permission by world</td>
<td>MP_WORLD_READ</td>
<td>0x0100</td>
</tr>
<tr>
<td>Write permission by world</td>
<td>MP_WORLD_WRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Execute permission by world</td>
<td>MP_WORLD_EXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>World permission mask</td>
<td>MP_WORLD_MASK</td>
<td>0x0f00</td>
</tr>
<tr>
<td>All permissions for owner, group, and</td>
<td>MP_WORLD_ACCESS</td>
<td>0x0777</td>
</tr>
<tr>
<td>world</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System permission mask</td>
<td>MP_SYSTM_MASK</td>
<td>0xf000</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
5-type/language

Description

Contains the module’s type (first byte) and language (second byte). The language codes indicate if the module is executable and which language the run-time system requires for execution, if any.

Port Generic Default Value

Macro

\( (MT\_DATA << 8) + ML\_OBJECT \)

EditMod
0x401

Port Specific Override Value

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values

Module type values and language codes are located in the header file, module.h, and are listed in Table 1-3 and Table 1-4.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not used (wildcard value in system calls)</td>
<td>MT_ANY</td>
<td>0x0000</td>
</tr>
<tr>
<td>Program module</td>
<td>MT_PROGRAM</td>
<td>0x0001</td>
</tr>
<tr>
<td>Subroutine module</td>
<td>MT_SUBROUT</td>
<td>0x0002</td>
</tr>
<tr>
<td>Multi-module (reserved for future use)</td>
<td>MT_MULTI</td>
<td>0x0003</td>
</tr>
<tr>
<td>Data module</td>
<td>MT_DATA</td>
<td>0x0004</td>
</tr>
<tr>
<td>Configuration data block data module</td>
<td>MT_CDBDATA</td>
<td>0x0005</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0xb-0xa</td>
<td>0xb-0xa</td>
</tr>
<tr>
<td>User trap library</td>
<td>MT_TRAPLIB</td>
<td>0x000b</td>
</tr>
</tbody>
</table>
### Table 1-3. *m_tylan* Available Module Type Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>System module</td>
<td>MT_SYSTEM</td>
<td>0x000c</td>
</tr>
<tr>
<td>File manager module</td>
<td>MT_FILEMAN</td>
<td>0x000d</td>
</tr>
<tr>
<td>Physical device driver</td>
<td>MT_DEVDRVR</td>
<td>0x000e</td>
</tr>
<tr>
<td>Device descriptor module</td>
<td>MT_DEVDESC</td>
<td>0x000f</td>
</tr>
<tr>
<td>User definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module type mask</td>
<td>MT_MASK</td>
<td>0xff00</td>
</tr>
</tbody>
</table>

### Table 1-4. *m_tylan* Available Language Code Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unspecified language (wildcard in system calls)</td>
<td>ML_ANY</td>
<td>0x0</td>
</tr>
<tr>
<td>Machine language</td>
<td>ML_OBJECT</td>
<td>0x1</td>
</tr>
<tr>
<td>Basic I-code (reserved for future use)</td>
<td>ML_ICODE</td>
<td>0x2</td>
</tr>
<tr>
<td>Pascal P-code (reserved for future use)</td>
<td>MLPCODE</td>
<td>0x3</td>
</tr>
<tr>
<td>C I-code (reserved for future use)</td>
<td>ML_CCIDE</td>
<td>0x4</td>
</tr>
<tr>
<td>Cobol I-code (reserved for future use)</td>
<td>ML_CBLICODE</td>
<td>0x5</td>
</tr>
<tr>
<td>Fortran</td>
<td>ML_FRTNCODE</td>
<td>0x6</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0x7-0xf</td>
<td>0x7-0xf</td>
</tr>
<tr>
<td>User-definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module language mask</td>
<td>ML_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
Chapter 1: Low-Level System Configuration Module (cnfgdata)

**EditMod Labels**

- 1-module header
- 6-revision/attributes

**Description**

Contains the module’s attributes (first byte) and revision (second byte).

**Port Generic Default Value**

Macro

MA_REENT<<8

EditMod

0x8000

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**

Module attribute and revision codes are located in the header file module.h., and are listed in Table 1-5.

If two modules with the same name are found in the memory search or are loaded into the current module directory, only the module with the highest revision level is kept. This enables easy substitution of modules for update or correction.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is re-entrant (sharable by multiple tasks).</td>
<td>MA_REENT</td>
<td>0x8000</td>
</tr>
<tr>
<td>(shifted left to first byte: MA_REENT&lt;&lt;8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The module is sticky. A sticky module is not removed from memory until its link count becomes -1 or memory is required for another use.</td>
<td>MA_GHOST</td>
<td>0x4000</td>
</tr>
<tr>
<td>(shifted left to first byte: MA_GHOST&lt;&lt;8)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The module is a system-state module. **MA_SUPER**
(shifted left to first byte: `MA_SUPER<<8`) 0x20
(shifted left to first byte: `0x2000`

User-definable revision number 0x0-0xfe 0x0-0xfe

Module attribute mask **MA_MASK** 0xff00

Module revision mask **MR_MASK** 0x00ff

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is a system-state module.</td>
<td>MA_SUPER</td>
<td>0x20 (shifted left to first byte: 0x2000)</td>
</tr>
<tr>
<td>User-definable revision number</td>
<td>0x0-0xfe</td>
<td>0x0-0xfe</td>
</tr>
<tr>
<td>Module attribute mask</td>
<td>MA_MASK</td>
<td>0xff00</td>
</tr>
<tr>
<td>Module revision mask</td>
<td>MR_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
Chapter 1: Low-Level System Configuration Module (cnfgdata)

m_edit
MH_EDITION

EditMod Labels
1-module header
7-edition

Description
Indicates the software release level for maintenance. OS-9® does not use this field. Whenever a program is revised (even for a small change), increase this number. We recommend internal documentation within the source program be keyed to this system.

Port Generic Default Value
1

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 65535

Console Device Fields
The console device fields are in the order they appear during an interactive EditMod session. Defined fields may appear in a different order in config.des. The field values can be changed using the EditMod utility or by modifying the config.des description file. See cnfgdata Module Field Configuration Options for detailed instructions on changing these fields.

Table 1-6. Console Device Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>console_name</td>
<td>CONS_NAME</td>
</tr>
<tr>
<td>cons_vector</td>
<td>CONS_VECTOR</td>
</tr>
<tr>
<td>cons_priority</td>
<td>CONS_PRIORITY</td>
</tr>
<tr>
<td>cons_level</td>
<td>CONS_LEVEL</td>
</tr>
<tr>
<td>cons_timeout</td>
<td>CONS_TIMEOUT</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>cons_parity</td>
<td>CONS_PARITY</td>
</tr>
<tr>
<td>cons_baudrate</td>
<td>CONS_BAUDRATE</td>
</tr>
<tr>
<td>cons_wordsize</td>
<td>CONS_WORDSIZE</td>
</tr>
<tr>
<td>cons_stopbits</td>
<td>CONS_STOPBITS</td>
</tr>
<tr>
<td>cons_flow</td>
<td>CONS_FLOW</td>
</tr>
</tbody>
</table>
EditMod Labels
2-configuration data
1-console port data structure
1-console port name

Description
Contains the console device name string.

Macro Example
#define CONS_NAME "iovcons"

Port Generic Default Value
NULL

Port Specific Override Value
Refer to ROM/CNFGBDATA/config.des (Figure 1-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
EditMod Labels
2-configuration data
1-console port data structure
2-interrupt vector number

Description
This is the vector number of the console device passed to the processor at interrupt time.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295
**EditMod Labels**

2-configuration data  
1-console port data structure  
3-interrupt priority

**Description**

This is the software (polling) priority for the console device on the IRQ polling table.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to ROM/CNFQDATA/config.des (Figure 1-3).

**Available Values**

The interrupt priority value range is 0-65534 (65535 is reserved). A non-zero priority determines the position of the device within the vector. Lower values are polled first.

Some considerations to keep in mind when selecting an interrupt priority:

- A priority of 0 indicates the device desires exclusive use of the vector.
- If the priority is 1, it is polled first and no other device can have a priority of 1 on the vector. For all other priority values, more than one device can share the same priority on a vector. In this case, first-in, first-out (FIFO) scheduling determines the order of precedence in the polling table for the devices.
- OS-9 does not allow a device to claim exclusive use of a vector if another device has already been installed on the vector. Additionally, it does not allow another device to use the vector once the vector has been claimed for exclusive use.
- This value is software dependent.

**See Also**

F_IRQ system call entry in the *OS-9 Technical Manual*. 
**EditMod Labels**

- 2-configuration data
- 1-console port data structure
- 4-interrupt level

**Description**

This is the hardware priority of the console device interrupt. When a device interrupts the processor, the level of the interrupt is used to mask lower priority interrupts.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**

0 to 4294967295. The number of supported interrupt levels is dependent on the processor being used (for example, 1-7 on 680x0 type CPUs).

**See Also**

The *OS-9 Input/Output System* section of the *OS-9 Technical Manual*. 
EditMod Labels
2-configuration data
1-console port data structure
5-polling timeout

Description
Polling time-out value for the console device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295
EditMod Labels
2-configuration data
1-console port data structure
6-parity

Description
Parity mode to be used by the console device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGBDATA/config.des (Figure 1-3).

Available Values
The configuration modules parity values are located in the header file, rom.h, and are listed in Table 1-7.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>No parity</td>
<td>CONS_NOPARITY</td>
<td>0x00</td>
</tr>
<tr>
<td>Odd parity</td>
<td>CONS_ODDPARITY</td>
<td>0x01</td>
</tr>
<tr>
<td>Even parity</td>
<td>CONS_EVENPARITY</td>
<td>0x02</td>
</tr>
<tr>
<td>Mark parity</td>
<td>CONS_MARKPARITY</td>
<td>0x03</td>
</tr>
<tr>
<td>Space parity</td>
<td>CONS_SPACEPARITY</td>
<td>0x04</td>
</tr>
<tr>
<td>Parity mask</td>
<td>CONS_PARITY_MASK</td>
<td>0x0F</td>
</tr>
<tr>
<td>Parity shift</td>
<td>CONS_PARITY_SHIFT</td>
<td>0</td>
</tr>
</tbody>
</table>
Chapter 1: Low-Level System Configuration Module (cnfgdata)

cons_baudrate
CONS_BAUDRATE

EditMod Labels
2-configuration data
1-console port data structure
7-baud rate

Description
Baud rate to be used by the console device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
The configuration modules baud rate values are located in the header file, rom.h, and are listed in Table 1-8.

Table 1-8. cons_baudrate Available Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardwire baud rate</td>
<td>CONS_BAUDRATE_HARWD</td>
<td>0x00</td>
</tr>
<tr>
<td>50 bits per second (bps)</td>
<td>CONS_BAUDRATE_50</td>
<td>0x01</td>
</tr>
<tr>
<td>75 bps</td>
<td>CONS_BAUDRATE_75</td>
<td>0x02</td>
</tr>
<tr>
<td>110 bps</td>
<td>CONS_BAUDRATE_110</td>
<td>0x03</td>
</tr>
<tr>
<td>134.5 bps</td>
<td>CONS_BAUDRATE_134PS</td>
<td>0x04</td>
</tr>
<tr>
<td>150 bps</td>
<td>CONS_BAUDRATE_150</td>
<td>0x05</td>
</tr>
<tr>
<td>300 bps</td>
<td>CONS_BAUDRATE_300</td>
<td>0x06</td>
</tr>
<tr>
<td>600 pbs</td>
<td>CONS_BAUDRATE_600</td>
<td>0x07</td>
</tr>
<tr>
<td>1200 bps</td>
<td>CONS_BAUDRATE_1200</td>
<td>0x08</td>
</tr>
<tr>
<td>1800 bps</td>
<td>CONS_BAUDRATE_1800</td>
<td>0x09</td>
</tr>
</tbody>
</table>
### Table 1-8. `cons_baudrate` Available Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000 bps</td>
<td>CONS_BAUDRATE_2000</td>
<td>0x0A</td>
</tr>
<tr>
<td>2400 bps</td>
<td>CONS_BAUDRATE_2400</td>
<td>0x0B</td>
</tr>
<tr>
<td>3600 bps</td>
<td>CONS_BAUDRATE_3600</td>
<td>0x0C</td>
</tr>
<tr>
<td>4800 bps</td>
<td>CONS_BAUDRATE_4800</td>
<td>0x0D</td>
</tr>
<tr>
<td>7200 bps</td>
<td>CONS_BAUDRATE_7200</td>
<td>0x0E</td>
</tr>
<tr>
<td>9600 bps</td>
<td>CONS_BAUDRATE_9600</td>
<td>0x0F</td>
</tr>
<tr>
<td>19,200 bps</td>
<td>CONS_BAUDRATE_19200</td>
<td>0x10</td>
</tr>
<tr>
<td>31,250 bps</td>
<td>CONS_BAUDRATE_31250</td>
<td>0x11</td>
</tr>
<tr>
<td>38,400 bps</td>
<td>CONS_BAUDRATE_38400</td>
<td>0x12</td>
</tr>
<tr>
<td>56,000 bps</td>
<td>CONS_BAUDRATE_56000</td>
<td>0x13</td>
</tr>
<tr>
<td>57,600 bps</td>
<td>CONS_BAUDRATE_57600</td>
<td>0x14</td>
</tr>
<tr>
<td>64,000 bps</td>
<td>CONS_BAUDRATE_64000</td>
<td>0x15</td>
</tr>
<tr>
<td>115,200 bps</td>
<td>CONS_BAUDRATE_115200</td>
<td>0x16</td>
</tr>
<tr>
<td>No echo</td>
<td>CONS_NOECHO</td>
<td>0x80</td>
</tr>
<tr>
<td>Baud rate mask</td>
<td>CONS_BAUDRATE_MASK</td>
<td>0x3F</td>
</tr>
</tbody>
</table>
Chapter 1: Low-Level System Configuration Module (cnfgdata)

cons_wordsizel
CONS_WORDSIZE

EditMod Labels
2-configuration data
1-console port data structure
8-character size

Description
Bits-per-byte to be used by the console device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFCDATA/config.des (Figure 1-3).

Available Values
The configuration modules word size values are located in the header file, rom.h, and are listed in Table 1-9.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bit word size</td>
<td>CONS_8BITS</td>
<td>0x00</td>
</tr>
<tr>
<td>7 bit word size</td>
<td>CONS_7BITS</td>
<td>0x40</td>
</tr>
<tr>
<td>6 bit word size</td>
<td>CONS_6BITS</td>
<td>0x80</td>
</tr>
<tr>
<td>5 bit word size</td>
<td>CONS_5BITS</td>
<td>0xC0</td>
</tr>
<tr>
<td>Word size mask</td>
<td>CONS_DBITS_MASK</td>
<td>0xC0</td>
</tr>
<tr>
<td>Word size shift</td>
<td>CONS_DBITS_SHIFT</td>
<td>6</td>
</tr>
</tbody>
</table>
EditMod Labels
2-configuration data
1-console port data structure
9-stop bit

Description
Number of stop bits to be used by the console device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
The configuration modules stop bit values are located in the header file rom.h, and are listed in Table 1-10.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop bit length of 1</td>
<td>CONS_1STOP</td>
<td>0x00</td>
</tr>
<tr>
<td>Stop bit length of 1.5</td>
<td>CONS_1P5STOP</td>
<td>0x10</td>
</tr>
<tr>
<td>Stop bit length of 2</td>
<td>CONS_2STOP</td>
<td>0x20</td>
</tr>
<tr>
<td>Stop bit mask</td>
<td>CONS_STOP_MASK</td>
<td>0x30</td>
</tr>
<tr>
<td>Stop bit shift</td>
<td>CONS_STOP_SHIFT</td>
<td>0x40</td>
</tr>
<tr>
<td>Stop data bit shift</td>
<td>CONS_DBITS_SHIFT</td>
<td>0x60</td>
</tr>
</tbody>
</table>
**EditMod Labels**

- 2-configuration data
- 1-console port data structure
- 10-flow control

**Description**

Flow control mode of the console device.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**

The configuration modules flow control values are located in the header file, `rom.h`, and are listed in Table 1-11.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>No handshaking</td>
<td>CONS_NOSHAKE</td>
<td>0x00</td>
</tr>
<tr>
<td>XOFF, any character on</td>
<td>CONS_SWSHAKE</td>
<td>0x01</td>
</tr>
<tr>
<td>Hardware handshaking</td>
<td>CONS_HWSHAKE</td>
<td>0x02</td>
</tr>
<tr>
<td>Strictly XON-XOFF</td>
<td>CONS_SWSTRICT</td>
<td>0x03</td>
</tr>
</tbody>
</table>

**Communication Device Fields**

The communication device fields are in the order they appear during an interactive EditMod session. Defined fields may appear in a different order in `config.des`. The fields can be changed using the EditMod utility or by modifying the description files. See cnfgdata Module Field Configuration Options for detailed instructions on changing these fields.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>comm_name</td>
<td>COMM_NAME</td>
</tr>
<tr>
<td>cons_vector</td>
<td>COMM_VECTOR</td>
</tr>
<tr>
<td>cons_priority</td>
<td>COMM_PRIORITY</td>
</tr>
<tr>
<td>cons_level</td>
<td>COMM_LEVEL</td>
</tr>
<tr>
<td>cons_timeout</td>
<td>COMM_TIMEOUT</td>
</tr>
<tr>
<td>cons_parity</td>
<td>COMM_PARITY</td>
</tr>
<tr>
<td>cons_baudrate</td>
<td>COMM_BAUDRATE</td>
</tr>
<tr>
<td>cons_wordsize</td>
<td>COMM_WORDSIZE</td>
</tr>
<tr>
<td>cons_stopbits</td>
<td>COMM_STOPBITS</td>
</tr>
<tr>
<td>cons_flow</td>
<td>COMM_FLOW</td>
</tr>
</tbody>
</table>
comm_name
COMM_NAME

EditMod Labels
2-configuration data
2-communication port data structure
1-communication port name

Description
Contains the communication device name string.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**cons_vector**

**EditMod Labels**

2-configuration data  
1-console port data structure  
2-interrupt vector number

**Description**

This is the vector number of the console device passed to the processor at interrupt time.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**

0 to 4294967295
EditMod Labels

2-configuration data
1-console port data structure
3-interrupt priority

Description

This is the software (polling) priority for the console device on the IRQ polling table.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values

The interrupt priority value range is 0-65534 (65535 is reserved). A non-zero priority determines the position of the device within the vector. Lower values are polled first.

Some considerations to keep in mind when selecting an interrupt priority:

- A priority of 0 indicates the device desires exclusive use of the vector.
- If the priority is 1, it is polled first and no other device can have a priority of 1 on the vector. For all other priority values, more than one device may share the same priority on a vector. In this case, first-in, first-out (FIFO) scheduling determines the order of precedence in the polling table for the devices.
- OS-9 does not allow a device to claim exclusive use of a vector if another device has already been installed on the vector. Additionally, it does not allow another device to use the vector once the vector has been claimed for exclusive use.
- This value is software dependent.

See Also

F_IRQ system call entry in the OS-9 Technical Manual.
EditMod Labels
- 2-configuration data
- 1-console port data structure
- 4-interrupt level

Description
This is the hardware priority of the console device interrupt. When a device interrupts the processor, the level of the interrupt is used to mask lower priority interrupts.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295. The number of supported interrupt levels is dependent on the processor being used (for example, 1-7 on 680x0 type CPUs).

See Also
The OS-9 Input/Output System section of the OS-9 Technical Manual.
cons_timeout
COMM_TIMEOUT

EditMod Labels
2-configuration data
1-console port data structure
5-polling timeout

Description
Polling time-out value for the console device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295
EditMod Labels
2-configuration data
1-console port data structure
6-parity

Description
Parity mode to be used by the console device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
The configuration modules parity values are located in the header file, rom.h, and are listed in Table 1-13.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>No parity</td>
<td>CONS_NOPARITY</td>
<td>0x00</td>
</tr>
<tr>
<td>Odd parity</td>
<td>CONS_ODDPARITY</td>
<td>0x01</td>
</tr>
<tr>
<td>Even parity</td>
<td>CONS_EVENPARITY</td>
<td>0x02</td>
</tr>
<tr>
<td>Mark parity</td>
<td>CONS_MARKPARITY</td>
<td>0x03</td>
</tr>
<tr>
<td>Space parity</td>
<td>CONS_SPACEPARITY</td>
<td>0x04</td>
</tr>
<tr>
<td>Parity mask</td>
<td>CONS_PARITY_MASK</td>
<td>0x0F</td>
</tr>
<tr>
<td>Parity shift</td>
<td>CONS_PARITY_SHIFT</td>
<td>0</td>
</tr>
</tbody>
</table>
**EditMod Labels**

2-configuration data
1-console port data structure
7-baud rate

**Description**

Baud rate to be used by the console device.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**

The configuration modules baud rate values are located in the header file, rom.h, and are listed in Table 1-14.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardwire baud rate</td>
<td>CONS_BAUDRATE_HARDWIRE</td>
<td>0x00</td>
</tr>
<tr>
<td>50 bits per second (bps)</td>
<td>CONS_BAUDRATE_50</td>
<td>0x01</td>
</tr>
<tr>
<td>75 bps</td>
<td>CONS_BAUDRATE_75</td>
<td>0x02</td>
</tr>
<tr>
<td>110 bps</td>
<td>CONS_BAUDRATE_110</td>
<td>0x03</td>
</tr>
<tr>
<td>134.5 bps</td>
<td>CONS_BAUDRATE_134P5</td>
<td>0x04</td>
</tr>
<tr>
<td>150 bps</td>
<td>CONS_BAUDRATE_150</td>
<td>0x05</td>
</tr>
<tr>
<td>300 bps</td>
<td>CONS_BAUDRATE_300</td>
<td>0x06</td>
</tr>
<tr>
<td>600 pbs</td>
<td>CONS_BAUDRATE_600</td>
<td>0x07</td>
</tr>
<tr>
<td>1200 bps</td>
<td>CONS_BAUDRATE_1200</td>
<td>0x08</td>
</tr>
<tr>
<td>1800 bps</td>
<td>CONS_BAUDRATE_1800</td>
<td>0x09</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------</td>
<td>---------</td>
</tr>
<tr>
<td>2000 bps</td>
<td>CONS_BAUDRATE_2000</td>
<td>0x0A</td>
</tr>
<tr>
<td>2400 bps</td>
<td>CONS_BAUDRATE_2400</td>
<td>0x0B</td>
</tr>
<tr>
<td>3600 bps</td>
<td>CONS_BAUDRATE_3600</td>
<td>0x0C</td>
</tr>
<tr>
<td>4800 bps</td>
<td>CONS_BAUDRATE_4800</td>
<td>0x0D</td>
</tr>
<tr>
<td>7200 bps</td>
<td>CONS_BAUDRATE_7200</td>
<td>0x0E</td>
</tr>
<tr>
<td>9600 bps</td>
<td>CONS_BAUDRATE_9600</td>
<td>0x0F</td>
</tr>
<tr>
<td>19,200 bps</td>
<td>CONS_BAUDRATE_19200</td>
<td>0x10</td>
</tr>
<tr>
<td>31,250 bps</td>
<td>CONS_BAUDRATE_31250</td>
<td>0x11</td>
</tr>
<tr>
<td>38,400 bps</td>
<td>CONS_BAUDRATE_38400</td>
<td>0x12</td>
</tr>
<tr>
<td>56,000 bps</td>
<td>CONS_BAUDRATE_56000</td>
<td>0x13</td>
</tr>
<tr>
<td>57,600 bps</td>
<td>CONS_BAUDRATE_57600</td>
<td>0x14</td>
</tr>
<tr>
<td>64,000 bps</td>
<td>CONS_BAUDRATE_64000</td>
<td>0x15</td>
</tr>
<tr>
<td>115,200 bps</td>
<td>CONS_BAUDRATE_115200</td>
<td>0x16</td>
</tr>
<tr>
<td>No echo</td>
<td>CONS_NOECHO</td>
<td>0x80</td>
</tr>
<tr>
<td>Baud rate mask</td>
<td>CONS_BAUDRATE_MASK</td>
<td>0x3F</td>
</tr>
</tbody>
</table>
Chapter 1: Low-Level System Configuration Module (cnfgdata)

**cons_wordsize**
**COMM_WORDSIZE**

**EditMod Labels**
- 2-configuration data
- 1-console port data structure
- 8-character size

**Description**
Bits-per-byte to be used by the console device.

**Port Generic Default Value**
0 (zero)

**Port Specific Override Value**
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**
The configuration module word size values are located in the header file, *rom.h*, and are listed in Table 1-15.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bit word size</td>
<td>CONS_8BITS</td>
<td>0x00</td>
</tr>
<tr>
<td>7 bit word size</td>
<td>CONS_7BITS</td>
<td>0x40</td>
</tr>
<tr>
<td>6 bit word size</td>
<td>CONS_6BITS</td>
<td>0x80</td>
</tr>
<tr>
<td>5 bit word size</td>
<td>CONS_5BITS</td>
<td>0xC0</td>
</tr>
<tr>
<td>Word size mask</td>
<td>CONS_DBITS_MASK</td>
<td>0xC0</td>
</tr>
<tr>
<td>Word size shift</td>
<td>CONS_DBITS_SHIFT</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 1-15. *cons_wordsize* Available Values
EditMod Labels

2-configuration data
1-console port data structure
9-stop bit

Description

Number of stop bits to be used by the console device.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values

The configuration modules stop bit values are located in the header file rom.h, and are listed in Table 1-16.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop bit length of 1</td>
<td>CONS_1STOP</td>
<td>0x00</td>
</tr>
<tr>
<td>Stop bit length of 1.5</td>
<td>CONS_1P5STOP</td>
<td>0x10</td>
</tr>
<tr>
<td>Stop bit length of 2</td>
<td>CONS_2STOP</td>
<td>0x20</td>
</tr>
<tr>
<td>Stop bit mask</td>
<td>CONS_STOP_MASK</td>
<td>0x30</td>
</tr>
<tr>
<td>Stop bit shift</td>
<td>CONS_STOP_SHIFT</td>
<td>0x40</td>
</tr>
<tr>
<td>Stop data bit shift</td>
<td>CONS_DBITS_SHIFT</td>
<td>0x60</td>
</tr>
</tbody>
</table>


**cons_flow**

**COMM_FLOW**

**EditMod Labels**

- 2-configuration data
- 1-console port data structure
- 10-flow control

**Description**

Flow control mode of the console device.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**

The configuration module flow control values are located in the header file, rom.h, and are listed in Table 1-17.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>No handshaking</td>
<td>CONS_NOSHAKE</td>
<td>0x00</td>
</tr>
<tr>
<td>XOFF, any character on</td>
<td>CONS_SWSHAKE</td>
<td>0x01</td>
</tr>
<tr>
<td>Hardware handshaking</td>
<td>CONS_HWSHAKE</td>
<td>0x02</td>
</tr>
<tr>
<td>Strictly XON-XOFF</td>
<td>CONS_SWSTRICT</td>
<td>0x03</td>
</tr>
</tbody>
</table>

**Debugger Fields**

The debugger fields are in the order they appear during an interactive EditMod session. Defined fields may appear in a different order in config.des. The fields can be changed using the EditMod utility or by modifying the description files. See cnfgdata Module Field Configuration Options for detailed instructions on how to change these fields.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug_name</td>
<td>DEBUGGER_NAME</td>
</tr>
<tr>
<td>debug_call_at_cold</td>
<td>DEBUGGER_COLD_FLAG</td>
</tr>
</tbody>
</table>
**debug_name**

**DEBUGGER_NAME**

**EditMod Labels**
2-configuration data
3-debugger data structure
1-debugger name

**Description**
Contains the name string of the debugger module used as the low-level debugger.

**Port Generic Default Value**
NULL

**Port Specific Override Value**
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
debug_call_at_cold
DEBUGGER_COLD_FLAG

EditMod Labels
2-configuration data
3-debugger data structure
2-cold start flag

Description
Cold start flag.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
The configuration modules debug_call_at_cold values are located in the header file, rom.h, and are listed in Table 1-19.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bypass calling debugger during boot sequence</td>
<td>DEBUG_BYPASS</td>
<td>0x0</td>
</tr>
<tr>
<td>Call debugger during boot sequence</td>
<td>DEBUG_CALL</td>
<td>0x1</td>
</tr>
</tbody>
</table>

Low-Level Protocol Manager Fields
The low-level protocol manager fields are in the order they appear during an interactive EditMod session. Defined fields may appear in a different order in config.des. The fields can be changed using the EditMod utility or by modifying the config.des description file. See cnfgdata Module Field Configuration Options for detailed instructions on changing these fields.
Table 1-20. Low-Level Protocol Manager Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxlppmprotos</td>
<td>LLPM_MAXPROTOS</td>
</tr>
<tr>
<td>maxrcvmbufs</td>
<td>LLPM_MAXRCVMBUFSS</td>
</tr>
<tr>
<td>maxlppmconns</td>
<td>LLPM_MAXCONNS</td>
</tr>
<tr>
<td>llpm_count</td>
<td>LLPM_COUNT</td>
</tr>
</tbody>
</table>
maxllpmprotos
LLPM_MAXPROTOS

EditMod Labels
2-configuration data
4-low level protocol manager data structure
1-maximum number of protocols

Description
Maximum number of protocol modules allowed on the protocol stack.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 65535
Chapter 1: Low-Level System Configuration Module (cnfgdata)

maxrcvmbufs
LLPM_MAXRCVMBUF

EditMod Labels
2-configuration data
4-low level protocol manager data structure
2-maximum number of receive mbufs

Description
Maximum number of memory buffers available for receiving packets. The size of each memory buffer varies depending on the driver used. (For example, llslip: 1024, ll21040: 1520).

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 65535
maxllpmconns
LLPM_MAXCONNS

EditMod Labels
2-configuration data
4-low level protocol manager data structure
3-maximum number of connections

Description
Maximum number of low-level protoman connections allowed.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 65535
Chapter 1: Low-Level System Configuration Module (cnfgdata)

EditMod Labels
2-configuration data
4-low level protocol manager data structure
4-number of data entries

Description
Number of low-level interface data entries.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295

Interface Data Fields
The interface data fields are in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des. The fields can be changed using the EditMod utility or by modifying the description files. See cnfgdata Module Field Configuration Options for detailed instructions on changing these fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address</td>
<td>Low-level IP address</td>
</tr>
<tr>
<td>subnet_mask</td>
<td>Low-level subnet mask</td>
</tr>
<tr>
<td>brdcst_address</td>
<td>Low-level broadcast address</td>
</tr>
<tr>
<td>gw_address</td>
<td>Low-level gateway address</td>
</tr>
<tr>
<td>mac_address</td>
<td>Low-level MAC address</td>
</tr>
<tr>
<td>hwtype</td>
<td>Low-level interface data driver type</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>if_flags</code></td>
<td>Interface flags</td>
</tr>
<tr>
<td><code>if_name</code></td>
<td>Low-level protocol manager name</td>
</tr>
<tr>
<td><code>port_address</code></td>
<td>Low-level protocol manager physical address</td>
</tr>
<tr>
<td><code>if_vector</code></td>
<td>Low-level protocol manager vector number</td>
</tr>
<tr>
<td><code>if_priority</code></td>
<td>Low-level protocol manager polling priority</td>
</tr>
<tr>
<td><code>if_level</code></td>
<td>Low-level protocol manager hardware priority</td>
</tr>
</tbody>
</table>
**EditMod Labels**

2-configuration data  
4-low level protocol manager data structure  
5-low level protocol interface data  
<n>-low level protocol interface data<n>  
1-internet address

**Description**

Low-level internet protocol (IP) address.

**Port Generic Default Value**

0.0.0.0

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**

Any dot(.) separated four item sequence of decimal numbers in the range of zero to 255.
subnet_mask
Low-level Subnet Mask

EditMod Labels
2-configuration data
4-low level protocol manager data structure
5-low level protocol interface data
<n>-low level protocol interface data[n]
2-subnet mask

Description
Low-level interface data subnet mask.

Port Generic Default Value
0.0.0.0

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
Any dot(.) separated four item sequence of decimal numbers in the range of zero to 255.
**brdcst_address**
Low-level Broadcast Address

**EditMod Labels**
- 2-configuration data
- 4-low level protocol manager data structure
- 5-low level protocol interface data
- <n>-low level protocol interface data[n]
- 3-broadcast address

**Description**
Low-level interface data broadcast address.

**Port Generic Default Value**
0.0.0.0

**Port Specific Override Value**
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**
Any dot(.) separated four item sequence of decimal numbers in the range of zero to 255.
gw_address
Low-level Gateway Address

EditMod Labels
2-configuration data
4-low level protocol manager data structure
5-low level protocol interface data
<n>-low level protocol interface data[n]
4-gateway address

Description
Low-level interface data gateway address.

Port Generic Default Value
0.0.0.0

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
Any dot(.) separated four item sequence of decimal numbers in the range of zero to 255.
mac_address
Low-level MAC address

EditMod Labels
2-configuration data
4-low level protocol manager data structure
5-low level protocol interface data
<n>-low level protocol interface data[n]
5-MAC (ethernet) address

Description
Low-level MAC (Ethernet address), machine address or hardware address.

Port Generic Default Value
0:0:0:0:0:0

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
Any colon(:) separated six item sequence of hexadecimal numbers in the range of zero to 255 (0xff). The 0x or $ prefix is not valid.
**hwtype**

Low-level Interface Data Driver Type

**EditMod Labels**

- 2-configuration data
- 4-low level protocol manager data structure
- 5-low level protocol interface data
- \(<n>\)-low level protocol interface data[\(<n>\)]
- 6-driver type

**Description**

Low-level interface data driver type.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des ([Figure 1-3](#)).

**Available Values**

The configuration modules `hwtype` values are located in the header file, `rom.h`, and are listed in Table 1-22.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>No driver type</td>
<td>LLPM_NOHW</td>
<td>0x0</td>
</tr>
<tr>
<td>SLIP driver type</td>
<td>LLPM_SLIP</td>
<td>0x1</td>
</tr>
<tr>
<td>Ethernet driver type</td>
<td>LLPM_ETHER</td>
<td>0x2</td>
</tr>
</tbody>
</table>

Table 1-22. `hwtype` Available Values
Chapter 1: Low-Level System Configuration Module (cnfgdata)

if_flags
Interface Flags

EditMod Labels
2-configuration data
4-low level protocol manager data structure
5-low level protocol interface data
<n>-low level protocol interface data<n>
12-interface-specific flag(s)

Description
Interface flags.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
The configuration modules if_flags values are located in the header file, rom.h, and are listed in Table 1-23.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applies only to SLIP array entries.</td>
<td>LLIF_CSLIP_ON</td>
<td>0x8000</td>
</tr>
<tr>
<td>Applies only to SLIP array entries.</td>
<td>LLIF_CSLIP_OFF</td>
<td>0x0000</td>
</tr>
</tbody>
</table>
**if_name**

Low-level Protocol Manager Name

**EditMod Labels**

2-configuration data  
4-low level protocol manager data structure  
5-low level protocol interface data  
<n>-low level protocol interface data[n]  
13-interface name

**Description**

Contains the llpm interface device name string.

**Port Generic Default Value**

NULL

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**

Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
port_address
Low-level Protocol Manager Physical Address

EditMod Labels
2-configuration data
4-low level protocol manager data structure
5-low level protocol interface data
<n>-low level protocol interface data[n]
14-interface port address

Description
This is the absolute physical address of the llpm interface device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295
if_vector
Low-level Protocol Manager Vector Number

EditMod Labels
2-configuration data
4-low level protocol manager data structure
5-low level protocol interface data
<n>-low level protocol interface data<n>
15-interrupt vector

Description
This is the vector number of the llpm interface device passed to the processor at interrupt time.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295

Value range is hardware/software dependent and determined at the OS level (OS-9 vs. OS-9 for 68K).
**if_priority**

**Low-level Protocol Manager Polling Priority**

**EditMod Labels**

- 2-configuration data
- 4-low level protocol manager data structure
- 5-low level protocol interface data
- <n>-low level protocol interface data[n]
- 16-interrupt priority

**Description**

This is the software (polling) priority for the llpm interface device on the IRQ polling table.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**

The interrupt priority value range is 0-65534 (65535 is reserved). A non-zero priority determines the position of the device within the vector. Lower values are polled first. Some considerations to keep in mind when selecting an interrupt priority:

- A priority of 0 indicates the device desires exclusive use of the vector.
- If the priority is 1, it is polled first and no other device can have a priority of 1 on the vector. For all other priority values, more than one device may share the same priority on a vector. In this case, first-in, first-out (FIFO) scheduling determines the order of precedence in the polling table for the devices.
- OS-9 does not allow a device to claim exclusive use of a vector if another device has already been installed on the vector. Additionally, it does not allow another device to use the vector once the vector has been claimed for exclusive use.
- This value is software dependent.

**See Also**

F_IRQ system call entry in the OS-9 Technical Manual.
if_level
Low-level Protocol Manager Hardware Priority

EditMod Labels
2-configuration data
4-low level protocol manager data structure
5-low level protocol interface data
<n>-low level protocol interface data[n]
17-interrupt level

Description
This is the hardware priority of the llpm interface device interrupt. When a device
interrupts the processor, the level of the interrupt is used to mask out lower priority
devices.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 65535. The number of supported interrupt levels is dependent on the processor
being used (for example, 1-7 on 680x0 type CPUs).

See Also
The OS-9 Input/Output System section of the OS-9 Technical Manual.

Configuration Boot Data Fields
The configuration boot data fields are in the order they appear during an interactive
EditMod session. Defined fields can appear in a different order in config.des. The
fields can be changed using the EditMod utility or by modifying the description files.
See cnfgdata Module Field Configuration Options for detailed instructions on how to
change these fields.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boot_count</code></td>
<td>BOOT_COUNT</td>
</tr>
<tr>
<td><code>boot_cmdsize</code></td>
<td>BOOT_CMDSIZE</td>
</tr>
</tbody>
</table>
EditMod Labels
2-configuration data
5-boot services data structure
1-number of boot system entries

Description
Number of boot system configuration entries.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295
EditMod Labels
2-configuration data
5-boot services data structure
3-maximum size of user input string

Description
This field defines the maximum size of user input string during boot menu selection.

Port Generic Default Value
32 characters

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295

Boot Data Fields
The boot data fields are in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des. The fields can be changed using the EditMod utility or by modifying the description files. See cnfgdata Module Field Configuration Options for detailed instructions on how to change these fields.

Table 1-25. Boot Data Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot_abname</td>
<td>Abbreviated booter name</td>
</tr>
<tr>
<td>boot_newab</td>
<td>New abbreviated booter name</td>
</tr>
<tr>
<td>boot_newname</td>
<td>Optional replacement full name</td>
</tr>
<tr>
<td>boot_automenu</td>
<td>Booter types for registration</td>
</tr>
<tr>
<td>boot_params</td>
<td>Optional parameter string</td>
</tr>
<tr>
<td>autoboot_delay</td>
<td>Autoboot delay value</td>
</tr>
</tbody>
</table>
**boot_abname**

Abbreviated Booter Name

**EditMod Labels**

- configuration data
- boot services data structure
- boot data
- boot data[<n>]
- abbreviated booter name

**Description**

Abbreviated booter name.

**Port Generic Default Value**

NULL

**Port Specific Override Value**

Refer to ROM/CNFNGDATA/config.des *(Figure 1-3).*

**Available Values**

Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
Boot New Abbreviated Booter Name

**EditMod Labels**
- configuration data
- boot services data structure
- boot data
- boot data[n]
- optional replacement abname

**Description**
New abbreviated booter name.

**Port Generic Default Value**
NULL

**Port Specific Override Value**
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**boot_newname**
Optional Replacement Full Name

**EditMod Labels**
- 2-configuration data
- 5-boot services data structure
- 2-boot data
- <n>-boot data[<n>]
- 3-optional replacement full name

**Description**
Optional replacement full name.

**Port Generic Default Value**
NULL

**Port Specific Override Value**
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
Chapter 1: Low-Level System Configuration Module (cnfgdata)

boott_automenu
Booter Types For Registration

EditMod Labels
2-configuration data
5-boot services data structure
2-boot data
<n>-boot data[n]
4-auto/menu flag

Description
Booter types for registration.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
The configuration modules boot_automenu values are located in the header file, rom.h, and are listed in Table 1-26.

<table>
<thead>
<tr>
<th>Description</th>
<th>config.des Macro</th>
<th>EditMod Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto booter</td>
<td>BT_AUTO</td>
<td>0x1</td>
</tr>
<tr>
<td>Menu booter</td>
<td>BT_MENU</td>
<td>0x2</td>
</tr>
</tbody>
</table>
**boot_params**  
Optional Parameter String

**EditMod Labels**
- 2-configuration data
- 5-boot services data structure
- 2-boot data
- \(<n>\)-boot data[\(<n>\)]
- 5-optional parameter string

**Description**
Optional parameter string.

**Port Generic Default Value**
NULL

**Port Specific Override Value**
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
autoboot_delay
Autoboot Delay Value

EditMod Labels
2-configuration data
5-boot services data structure
2-boot data
<n>-boot data[n]
6-autoboot delay in microseconds

Description
Handled in the boot.sys module, this is the delay value in microseconds prior to proceeding with an autoboot entry.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

Available Values
0 to 4294967295

Notification Services Field
The notification field can be changed using the EditMod utility or by modifying the description files. See cnfgdata Module Field Configuration Options for detailed instructions on changing this field.

Table 1-27. Notification Services Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_notifiers</td>
<td>MAX_NOTIFIERS</td>
</tr>
</tbody>
</table>
**max_notifiers**  
MAX_NOTIFIERS

**EditMod Labels**
- 2-configuration data
- 6-notification services data structure
- 1-maximum number of registered notifiers

**Description**
Used by the notification services module to indicate the maximum number of notification routines that can be registered.

**Port Generic Default Value**
0 (zero)

**Port Specific Override Value**
Refer to ROM/CNFGDATA/config.des (Figure 1-3).

**Available Values**
0 to 4294967295. While the only adverse effect of defining a larger max_notifiers value than necessary is the extra memory used for the unused records, here are some considerations to help determine an acceptable value:

- Notification services are required by any module that needs to know when the systems are in transition from polled mode to interrupt mode. Essentially this means the low-level serial and ethernet drivers (including iovcons).
- A module generally only installs one notification routine, but if a single module is used for two ports (like io16550 on Powerstacks and PCs), it installs two.

**See Also**
The Low-Level System Configuration section and the Porting OS-9 section of OS-9 Porting Guide.
OS-9 Configuration Module (init)

The init (initialization) module contains configuration data used by the kernel and other OS-9 system modules to control system bootup and execution. Values that can be configured in the init module include:

- Initial system data table sizes
- Memory layout and characteristics
- Names of the system ticker and other OS extensions
- Flag fields specifying various operational modes
- Process scheduling control, including first process to execute

The next section in this chapter provides a detailed example of the two reconfiguration options you can use to change configuration values for this module.

The rest of this chapter provides a detailed list of all of the init module fields, including field descriptions and available values.

This chapter includes the following topics:

- Init Module Field Configuration Options
- Init Module Field Reference
- Module Header Fields
- Module Body Fields
- Memlist Fields
- Cachelist Fields
Init Module Field Configuration Options

To change an init module configuration field, you can use either of the following methods:

1. Use the EditMod utility to directly modify existing init modules either as a stand-alone module or as part of a merged module group (such as a boot image).
2. Modify the description file for the init module and rebuild it using the makefile provided.

Direct Modification Advantages

The direct modification method has the following advantages:

- **Fast**: No source configuration file rebuilds are necessary.
- **Temporary**: The original module or merged-module group configuration can be easily restored through the appropriate rebuild.
- **Contained**: Changes are limited to the individual boot image modified (merged-module option).

Description File/Rebuild Advantages

The advantage of the description file/rebuild method is that the changes are permanent and reproducible. Modifications apply to all subsequent module rebuilds and to all merged-module groups built containing the updated module.

Both methods are documented in this section. These procedures are used with the field descriptions starting with the Module Header Fields. For direct modification, use the EditMod LABELS data to navigate the EditMod menus. The DESCRIPTION FILE MACRO data identifies the macro you need to define/modify in the configuration sources to rebuild the init module.

Direct Modification

Use the EditMod utility and the following procedures to directly modify fields in the existing init module. The module can stand-alone or it can be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The field references later in this chapter contain a description of each configurable field, its supported values, and the sequence of menu options required by EditMod to modify that field.

Refer to the Utilities Reference for a full description of EditMod’s capabilities.
Refer to your board guide for information about how to modify the module lists and remake the boot images, and for specific boot image names.

**Direct Modification Procedures**

To modify the stand-alone module, complete the following steps:

1. Change to the CMDS/BOOTOBJ/INITS directory (see Figure 2-1).
2. Use EditMod to edit the module:
   
   `$EditMod -e init`

To modify the module as part of a merged module group, complete the following steps:

1. Change to the BOOTS/SYSTEMS/PORTBOOT directory (see Figure 2-2).
2. Use EditMod to edit the module:
   
   `$EditMod -e init -f=<boot image name>`
3. Use the menu selections provided in the EditMod LABELS section of the field reference later in this chapter to locate the fields you want to edit.
4. Select a new value for the field from the AVAILABLE VALUES section of the field reference. Enter that value at the EditMod prompt to modify the field.

5. If you want to make additional modifications, use the p command (previous) to step backward through the EditMod menus. Repeat Steps 3 and 4 until you have made all desired modifications to the init module.

6. Select the w command (write) to save the changes.

7. Select the q command (quit) to exit EditMod.

Unless you modified the init module in your boot image, you should rebuild your boot image to include the new init module.

Example EditMod Session

This example modifies init as part of the boot image rom.

$ EditMod -e init -f-rom

1. module header
2. init module contents

Which? [?/1-2/p/t/a/w/q] 2

1. installation site code                  : 0x0
2. cpu class                               : 0x1bc7
3. installation string                     : "PS7111"
4. OS-9000 level/revision string           : "OS-9000 for the ARM"
5. initial module name                     : "shell"
6. parameter list                          : ""
7. system RBF device                       : ""
8. system SCF device                       : "/term"
9. customization module list               : "OS9P2 fpu ssm"
10. ticker module name                     : "tkarm"
11. real-time clock module name            : "rtc7110"
12. IO manager module name                 : "Ioman"
13. user accounting module name            : ""
14. memory list
15. number of process table entries         : 0x40
16. number of path table entries            : 0x40
17. number of system event table entries    : 0x20
18. number of ticks per second             : 0x64
19. number of clock ticks per time slice    : 0x2
20. initial system priority                 : 0x80
21. initial minimum executable priority     : 0x0
22. initial maximum natural process age     : 0x0
23. system call dispatch table entries      : 0x100
24. reserved for system specific flags      : 0x0
25. system time zone                       : 0
26. OS-9000 level                          : 1
27. OS-9000 major release number           : 2
28. OS-9000 minor release number           : 0
29. OS-9000 edition number                 : 0
30. compatibility flags                    : 0x2
31. process signal queue size              : 0x20
32. pre-IO customization module list        : "irq7110 irq7111"
33. cache list

Unless you modified the init module in your boot image, you should rebuild your boot image to include the new init module.
Description File Modification

You can use these procedures to modify the init description file sources and rebuild the init module for your port directory. The DESCRIPTION FILE MACROS section of the field reference specifies the name of the macro you modify/define in the description files to configure the field. The value used in the define is chosen from the AVAILABLE VALUES specified for the field.

Figure 2-3. Directory Location for Modifying the init Description File

Description File Modification Procedures

1. Change to the INIT directory. (see Figure 2-3).
2. Edit the file config.des and read the included comments for more information on using the specific description file provided in your software distribution. The config.des file contains a list of macro names defined to override the global default values for the configuration fields.
3. Refer to the DESCRIPTION FILE MACRO section in the field reference later in this chapter to determine the macro name you define to configure the target field.
4. Read the comments in config.des to determine where to place the define for this macro.
5. Select the value you want to use to configure the field. See the AVAILABLE VALUES section of the field reference data for values or macros that can be used for the definition. Define the macro by entering a definition in the appropriate description files as follows:
   
   #define <macro> <value>

6. Save the changes and rebuild the module, entering the following command in the INIT directory:
   
   os9make

7. Rebuild your boot image to include the new init module.
Init Module Field Reference

This section contains a list of the most commonly configured fields in the init module. Each field entry contains the following information:

- **<Field name>** - The call name for each field that can be reconfigured in the module.
- **EditMod LABELS** - EditMod menu selections for navigating to the proper field in an EditMod session.
- **DESCRIPTION FILE MACRO** - The macro name you modify/define in the description file.
- **DESCRIPTION** - A brief description of the field's purpose and use.
- **EXAMPLE** - An optional example of the description file entry showing how to change the value of this field.
- **PORT GENERIC DEFAULT VALUE** - The value set in the port generic description file for this field. This is the value the field is assigned when the module is built, unless the appropriate macro has been defined in the port specific description file to override this default value.
- **PORT SPECIFIC OVERRIDE VALUE** - The value set in the port specific description file for this field. If defined, this is the value the field is assigned when the module is built, overriding the port generic default value.
- **AVAILABLE VALUES** - Values to which the field can be set through EditMod or the description files. In many cases, this data is presented in a table that maps a description of the value to a numeric value appropriate for entry in EditMod, and to a pre-defined macro available for use in the description file.

The init module is sometimes referred to as the configuration module. It is located in memory in the sysboot file or in ROM. The init module is a non-executable module of type MT_SYSTEM. The init module contains system parameters used to configure OS-9 during start-up. The parameters set up the initial table sizes and system device names and the init module must always be available to determine system limits. For example, the amount of memory to allocate for internal tables, the name of the first program to run (usually either syso or shell), an initial directory, and other initialization settings are specified. You can examine the system limits defined in the init module at any time.

The init module must be present in the system in order for OS-9 to work.

For more information on the init module, see the OS-9 Technical Manual.
Module Header Fields

The following section contains the module header fields in the order they appear during an interactive `EditMod` session. Defined fields can appear in a different order in `config.des`.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>_m_group</td>
<td>MH_GROUP</td>
</tr>
<tr>
<td>_m_user</td>
<td>MH_USER</td>
</tr>
<tr>
<td>mod_name</td>
<td>MH_NAME</td>
</tr>
<tr>
<td>m_access</td>
<td>MH_ACCESS</td>
</tr>
<tr>
<td>m_tylan</td>
<td>MH_TYLAN</td>
</tr>
<tr>
<td>m_attrev</td>
<td>MH_ATTREV</td>
</tr>
<tr>
<td>m_edit</td>
<td>MH_EDITION</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header

1-module owner’s group number

Description

Group ID of the module’s owner. The group number allows people working in the same department or on the same project to share a common identification number.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

0 to 65535
Chapter 2: OS-9 Configuration Module (init)

_EditMod Labels_

1-module header
2-module owner’s user number

_Description_

User ID of the module’s owner. The user number identifies a specific user.

_Port Generic Default Value_

0 (zero)

_Port Specific Override Value_

Refer to INIT/config.des (Figure 2-3).

_Available Values_

0 to 65535
EditMod Labels

1-module header
3-module name

Description
Contains the module name string.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**m_access**

MH_ACCESS

**EditMod Labels**

1-module header

4-access permissions

**Description**

Defines the permissible module access by its owner or by other users.

**Port Generic Default Value**

Macro

```
MP_OWNER_READ | MP_OWNER_EXEC | MP_GROUP_READ |
MP_GROUP_EXEC | MP_WORLD_READ | MP_WORLD_EXEC
```

EditMod

0x555

**Port Specific Override Value**

Refer to INIT/config.des (Figure 2-3).

**Available Values**

Module access permission values are located in the header file, module.h, and are listed in Table 2-2.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read permission by owner</td>
<td>MP_OWNER_READ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Write permission by owner</td>
<td>MP_OWNER_WRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Execute permission by owner</td>
<td>MP_OWNER_EXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Owner permission mask</td>
<td>MP_OWNER_MASK</td>
<td>0x000f</td>
</tr>
<tr>
<td>Read permission by group</td>
<td>MP_GROUP_READ</td>
<td>0x0010</td>
</tr>
<tr>
<td>Write permission by group</td>
<td>MP_GROUP_WRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Execute permission by group</td>
<td>MP_GROUP_EXEC</td>
<td>0x0040</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Group permission mask</td>
<td>MP_GROUP_MASK</td>
<td>0x00f0</td>
</tr>
<tr>
<td>Read permission by world</td>
<td>MP_WORLD_READ</td>
<td>0x0100</td>
</tr>
<tr>
<td>Write permission by world</td>
<td>MP_WORLD_WRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Execute permission by world</td>
<td>MP_WORLD_EXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>World permission mask</td>
<td>MP_WORLD_MASK</td>
<td>0x0f00</td>
</tr>
<tr>
<td>All permissions for owner, group, and world</td>
<td>MP_WORLD_ACCESS</td>
<td>0x0777</td>
</tr>
<tr>
<td>System permission mask</td>
<td>MP_SYSTM_MASK</td>
<td>0xf000</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
5-type/language

Description
Contains the module’s type (first byte) and language (second byte). The language codes indicate if the module is executable and which language the run-time system requires for execution, if any.

Port Generic Default Value
Macro
\[(MT\_DATA<<8) + ML\_OBJECT\]

EditMod
0x401

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Module type values and language codes are located in the header file, module.h, and are listed in Table 2-3 and Table 2-4.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not used (wildcard value in system calls)</td>
<td>MT_ANY</td>
<td>0x0000</td>
</tr>
<tr>
<td>Program module</td>
<td>MT_PROGRAM</td>
<td>0x0001</td>
</tr>
<tr>
<td>Subroutine module</td>
<td>MT_SUBROUT</td>
<td>0x0002</td>
</tr>
<tr>
<td>Multi-module (reserved for future use)</td>
<td>MT_MULTI</td>
<td>0x0003</td>
</tr>
<tr>
<td>Data module</td>
<td>MT_DATA</td>
<td>0x0004</td>
</tr>
<tr>
<td>Configuration data block data module</td>
<td>MT_CDBDATA</td>
<td>0x0005</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0xb-0xa</td>
<td>0xb-0xa</td>
</tr>
</tbody>
</table>
### Table 2-3. `m_tylan` Available Module Type Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>User trap library</td>
<td>MT_TRAPLIB</td>
<td>0x000b</td>
</tr>
<tr>
<td>System module</td>
<td>MT_SYSTEM</td>
<td>0x000c</td>
</tr>
<tr>
<td>File manager module</td>
<td>MT_FILEMAN</td>
<td>0x000d</td>
</tr>
<tr>
<td>Physical device driver</td>
<td>MT_DEVDRVR</td>
<td>0x000e</td>
</tr>
<tr>
<td>Device descriptor module</td>
<td>MT_DEVDESC</td>
<td>0x000f</td>
</tr>
<tr>
<td>User definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module type mask</td>
<td>MT_MASK</td>
<td>0xff00</td>
</tr>
</tbody>
</table>

### Table 2-4. `m_tylan` Available Language Code Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unspecified language ( wildcard in system calls)</td>
<td>ML_ANY</td>
<td>0x0</td>
</tr>
<tr>
<td>Machine language</td>
<td>ML_OBJECT</td>
<td>0x1</td>
</tr>
<tr>
<td>Basic I-code (reserved for future use)</td>
<td>ML_ICODE</td>
<td>0x2</td>
</tr>
<tr>
<td>Pascal P-code (reserved for future use)</td>
<td>ML_PCODE</td>
<td>0x3</td>
</tr>
<tr>
<td>C I-code (reserved for future use)</td>
<td>ML_CCODE</td>
<td>0x4</td>
</tr>
<tr>
<td>Cobol I-code (reserved for future use)</td>
<td>ML_CBLCODE</td>
<td>0x5</td>
</tr>
<tr>
<td>Fortran</td>
<td>ML_FRTNCODE</td>
<td>0x6</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0x7-0xf</td>
<td>0x7-0xf</td>
</tr>
<tr>
<td>User-definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module language mask</td>
<td>MLMASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels
1-module header
6-revision/attributes

Description
Contains the module’s attributes (first byte) and revision (second byte).

Port Generic Default Value
Macro
MA_REENT<<8

EditMod
0x8000

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Module attribute and revision codes are located in the header file module.h, and are listed in Table 2-5.

If two modules with the same name are found in the memory search or are loaded into the current module directory, only the module with the highest revision level is kept. This enables easy substitution of modules for update or correction.

Table 2-5. m_attrev Available Attribute and Revision Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is re-entrant (sharable by multiple tasks).</td>
<td>MA_REENT</td>
<td>0x80 (shifted left to first byte: 0x8000)</td>
</tr>
<tr>
<td>(shifted left to first byte: MA_REENT&lt;&lt;8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The module is sticky. A sticky module is not removed from memory until its link count becomes -1 or memory is required for another use.</td>
<td>MA_GHOST</td>
<td>0x40 (shifted left to first byte: 0x4000)</td>
</tr>
<tr>
<td>(shifted left to first byte: MA_GHOST&lt;&lt;8)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The module is a system-state module.
(MA_SUPER<<8)
0x20
(shifted left to first byte:
0x2000)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-definable revision number</td>
<td>0x0-0xfe</td>
<td>0x0-0xfe</td>
</tr>
<tr>
<td>Module attribute mask</td>
<td>MA_MASK</td>
<td>0xff00</td>
</tr>
<tr>
<td>Module revision mask</td>
<td>MR_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
7-edition

Description
Indicates the software release level for maintenance. OS-9 does not use this field. Whenever a program is revised (even for a small change), increase this number. It is recommended that internal documentation within the source program be keyed to this system.

Port Generic Default Value
1

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 65535

Module Body Fields

The following section contains the module body fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_site</td>
<td>SITE</td>
</tr>
<tr>
<td>m_cputyp</td>
<td>MPUCHIP</td>
</tr>
<tr>
<td>install_name</td>
<td>INSTALNAME</td>
</tr>
<tr>
<td>os9rev_name</td>
<td>OS9K_REVSTR</td>
</tr>
<tr>
<td>sysgo_name</td>
<td>SYS_START</td>
</tr>
<tr>
<td>sparam_string</td>
<td>SYS_PARAMS</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>drive_name</td>
<td>SYS_DEVICE</td>
</tr>
<tr>
<td>console_name</td>
<td>CONS_NAME</td>
</tr>
<tr>
<td>extens_list</td>
<td>EXTENSIONS</td>
</tr>
<tr>
<td>ticker_name</td>
<td>TICK_NAME</td>
</tr>
<tr>
<td>rtc_name</td>
<td>RTC_NAME</td>
</tr>
<tr>
<td>ioman_name</td>
<td>IOMAN_NAME</td>
</tr>
<tr>
<td>acct_name</td>
<td>USRACCT_NAME</td>
</tr>
<tr>
<td>m_procs</td>
<td>PROCS</td>
</tr>
<tr>
<td>m_paths</td>
<td>PATHS</td>
</tr>
<tr>
<td>m_events</td>
<td>EVENTS</td>
</tr>
<tr>
<td>m_ticksec</td>
<td>TICK_SEC</td>
</tr>
<tr>
<td>m_slice</td>
<td>SLICE</td>
</tr>
<tr>
<td>m_syspri</td>
<td>SYS_PRIOR</td>
</tr>
<tr>
<td>m_minpty</td>
<td>MINPTY</td>
</tr>
<tr>
<td>m_maxage</td>
<td>MAXPTY</td>
</tr>
<tr>
<td>m_dsptbl</td>
<td>DSPTBLSZ</td>
</tr>
<tr>
<td>m_cpucompat</td>
<td>CPUCOMPAT</td>
</tr>
<tr>
<td>m_tmzone</td>
<td>SYS_TMZONE</td>
</tr>
<tr>
<td>m_level</td>
<td>OS_LEVEL</td>
</tr>
<tr>
<td>m_major</td>
<td>OS_VERSION</td>
</tr>
<tr>
<td>m_minor</td>
<td>OS_REVISION</td>
</tr>
<tr>
<td>m_edition</td>
<td>OS_EDITION</td>
</tr>
</tbody>
</table>
Table 2-6. Module Header Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_compat</td>
<td>COMPAT</td>
</tr>
<tr>
<td>m_maxsigs</td>
<td>MAXSIGS</td>
</tr>
<tr>
<td>preio_name</td>
<td>PREIOS_NAME</td>
</tr>
</tbody>
</table>
EditMod Labels

2-init module contents
1-installation site code

Description
This field contains the installation site code. This user-definable field can be used to identify the site of the system.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 4294967295
EditMod Labels
2-init module contents
2-cpu class

Description
This field contains the CPU family type. For example 403, 603, 80386, etc.

Port Generic Default Value
80386

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 4294967295
EditMod Labels
2-init module contents
3-installation string

Description
Installation name string.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
EditMod Labels

2-init module contents
4-OS-9000 level/revision string

Description
Contains the OS-9 level revision string.

Port Generic Default Value
“OS-9000”

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
EditMod Labels
2-init module contents
5-initial module name

Description
Contains the name string of the first executable module.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**sparam_string**

**SYS_PARAMS**

---

**EditMod Labels**

2-init module contents
6-parameter list

**Description**

Contains the parameter string (if any) to be passed to the first executable module.

**Port Generic Default Value**

NULL

**Port Specific Override Value**

Refer to INIT/config.des (Figure 2-3).

**Available Values**

Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
EditMod Labels
2-init module contents
7-system RBF device

Description
Contains the initial default directory name string, usually /d0 or /h0. The system initially does a chd and chx to this device prior to forking the initial device. If the system does not use disk, this offset must be zero.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
EditMod Labels

2-init module contents
8-system SCF device

Description
Contains the initial I/O pathlist string, usually /term. This pathlist is opened as the standard I/O path for the initial process. It is generally used to set up the initial I/O paths to and from a terminal. The value should be set to NULL if no console device is in use.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**EditMod Labels**

2-init module contents
9-customization module list

**Description**

Contains the name string of a list of customization modules, if any. A customization module complements or changes existing standard system calls used by OS-9. These modules are searched for at start-up and are usually found in the bootfile. If found, they are executed in system state.

Module names in the name string are separated by spaces. The default name string to be searched for is OS9P2. If there are no customization modules, this value should be set to NULL.

**Port Generic Default Value**

"OS9P2"

**Port Specific Override Value**

Refer to INIT/config.des (Figure 2-3).

**Available Values**

Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
EditMod Labels
2-init module contents
10-ticker module name

Description
Contains the name string of the module used to generate the system clock tick. The kernel attempts to call this module when the first \_os\_setime system call is made.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
EditMod Labels
2-init module contents
11-real-time clock module name

Description
Contains the real-time clock module name string. The kernel attempts to call this module when the time is set, in other words when `_os_setime` is called.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to `INIT/config.des` (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as `\n` and `\012`).
EditMod Labels

2-init module contents
12-IO manager module name

Description
Contains the name string of the module handling I/O system calls. This string is normally set to ioman.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
acct_name
USRACCT_NAME

EditMod Labels
2-init module contents
13-user accounting module name

Description
Contains the name string of the user accounting module.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**m_procs**
PROCS

**EditMod Labels**

2-init module contents
15-number of process table entries

**Description**

This is the number of entries in the process descriptor table. If this table becomes full, it is expanded automatically.

**Port Generic Default Value**

64 (0x40)

**Port Specific Override Value**

Refer to INIT/config.des (Figure 2-3).

**Available Values**

0 to 65535
EditMod Labels
2-init module contents
16-number of path table entries

Description
This is the initial number of open paths in the system. If this table becomes full, it is expanded automatically.

Port Generic Default Value
64 (0x40)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 65535
EditMod Labels

2-init module contents
17-number of system event table entries

Description

This is the initial number of entries allowed in the events table. If this table becomes full, it is expanded automatically.

Refer to the OS-9 Technical Manual for specific information on events.

Port Generic Default Value

32 (0x20)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

0 to 65535
EditMod Labels
2-init module contents
18-number of ticks per second

Description
This is the number of ticks into which a second of time is divided.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 65535
EditMod Labels

2-init module contents
19-number of clock ticks per time slice

Description
This is the number of clock ticks per time-slice.

Port Generic Default Value
2

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 65535
EditMod Labels
2-init module contents
20-initial system priority

Description
This is the system priority at which the first module (usually sysgo or shell) is executed. This is generally the base priority at which all processes start.

Port Generic Default Value
128 (0x80)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 65535
EditMod Labels
2-init module contents
21-initial minimum executable priority

Description
This is the initial system minimum executable priority. m_minpty is discussed later in this chapter and in the OS-9 Technical Manual.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 65535
m_maxage
MAXPTY

EditMod Labels

2-init module contents
22-initial maximum natural process age

Description

This is the initial system maximum natural age. m_maxage is discussed later in this chapter and in the OS-9 Technical Manual.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

0 to 65535
EditMod Labels

2-init module contents
23-system call dispatch table entries

Description

This field contains the number of entries in the system call dispatch table. There must be at least 256 entries in this table, and each entry requires eight bytes.

Port Generic Default Value

256 (0x100)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

0 to 65535
EditMod Labels

2-init module contents
24 - reserved for system specific flags

Description

This field is reserved for system-specific flags.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

0 to 65535
**m_tmzone**
SYS_TMZONE

**EditMod Labels**
- 2-init module contents
- 25-system time zone

**Description**
This is the system time zone in minutes offset from Greenwich Mean Time (GMT). Therefore, this field would be 360 for a system six time zones west of GMT and -360 for a system six time zones east of GMT.

**Port Generic Default Value**
0 (zero)

**Port Specific Override Value**
Refer to INIT/config.des (Figure 2-3).

**Available Values**
-32768 to 32767
EditMod Labels
2-init module contents
26-OS-9000 level

Description
The OS-9 level is the first byte of a four byte field that is divided into four parts: level, version, revision, and edition number. For example, level 2, version 2, revision 1, edition 0 is 2210.

Port Generic Default Value
1

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 255
2-init module contents
27-OS-9000 major release number

Description
The OS-9 level is the second byte of a four byte field that is divided into four parts: level, version, revision, and edition number. For example, level 2, version 2, revision 1, edition 0 is 2210.

Port Generic Default Value
2

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 255
EditMod Labels

2-init module contents
28-OS-9000 minor release level

Description

The OS-9 level is the third byte of a four byte field that is divided into four parts: level, version, revision, and edition number. For example, level 2, version 2, revision 1, edition 0 is 2210.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

0 to 255
EditMod Labels

2-init module contents
28-OS-9000 edition number

Description
The OS-9 level is the fourth byte of a four byte field that is divided into four parts: level, version, revision, and edition number. For example, level 2, version 2, revision 1, edition 0 is 2210.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 255
EditMod Labels
2-init module contents
30-compatibility flags

Description
This byte is used for revision compatibility.

Port Generic Default Value
Macro
B_WIPEMEM

EditMod
0x2

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Revision compatibility values are located in the header file init.h and are listed in Table 2-7.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set to ignore sticky bit in the module headers</td>
<td>B_GHOST</td>
<td>0x01</td>
</tr>
<tr>
<td>Set to patternize memory when allocated and returned</td>
<td>B_WIPEMEM</td>
<td>0x02</td>
</tr>
<tr>
<td>Set to inform the kernel not to automatically set the clock during coldstart</td>
<td>B_NOCLOCK</td>
<td>0x04</td>
</tr>
<tr>
<td>Set to not automatically expand system tables</td>
<td>B_EXPTBL</td>
<td>0x08</td>
</tr>
<tr>
<td>Set to have the kernel align user-state data modules on MMU boundaries when SSM is being used</td>
<td>B_UDATMOD</td>
<td>0x10</td>
</tr>
<tr>
<td>Set to disable the validation of the CRC for new modules</td>
<td>B_NOCRC</td>
<td>0x20</td>
</tr>
</tbody>
</table>
EditMod Labels
2-init module contents
31-process signal queue size

Description
This field specifies the default maximum number of signals queued up for a process.

Port Generic Default Value
32 (0x20)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 65535
EditMod Labels

2-init module contents
32-pre-I/O customization module list

Description
Contains the name string of a list of pre-I/O customization modules, if any. These
extension modules are initialized and called prior to the initialization of the I/O system
during bootstrap. For more information on customization modules, refer to the
description of m_extens.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as
\n and \012).

Memlist Fields

The memlist fields are in the order they appear during an interactive EditMod session.
Defined fields can appear in a different order in the description files. The fields can be
changed using the EditMod utility or by modifying the description files. See Init
Module Field Configuration Options for detailed instructions on changing these fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Memory type code</td>
</tr>
<tr>
<td>prior</td>
<td>Memory allocation priority</td>
</tr>
<tr>
<td>access</td>
<td>Access permissions</td>
</tr>
<tr>
<td>blksz</td>
<td>Search block size</td>
</tr>
<tr>
<td>lolim</td>
<td>Beginning block address</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------</td>
</tr>
<tr>
<td>hilim</td>
<td>Ending block address</td>
</tr>
<tr>
<td>desc</td>
<td>Memory list description</td>
</tr>
<tr>
<td>dma_addr</td>
<td>External bus address</td>
</tr>
</tbody>
</table>
EditMod Labels
2-init module contents
14-memory list
1-Add additional item to list
n-memory list [n-1]
1- memory type code (color)

Description
This is the memory type code.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Memory type values are defined in the header file, memory.h, and are listed in Table 2-9.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>System RAM memory</td>
<td>MEM_SYS</td>
<td>0x01</td>
</tr>
<tr>
<td>Shared memory (0x8000 - 0xffff)</td>
<td>MEM_SHARED</td>
<td>0x8000</td>
</tr>
</tbody>
</table>
Memory Allocation Priority

Description
This is the memory allocation priority. High priority RAM is allocated first (255 - 0). If the block priority is 0, the block can only be allocated by a request for the specific color (type) of the block.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 65535
EditMod Labels

2-init module contents
14-memory list
1-Add additional item to list
n-memory list [n-1]
3-access permissions

Description

This is the access permissions.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

Memory type access bit are defined in the header file, alloc.h, and in Table 2-10.

Table 2-10. Access Bit Definitions for Memory Type

<table>
<thead>
<tr>
<th>Description</th>
<th>config.des Macro</th>
<th>EditMod Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 0Indicates memory allocatable by user processes.</td>
<td>B_USERRAM</td>
<td>0x01</td>
</tr>
<tr>
<td>bit 1Indicates parity memory; the kernel initializes it during start-up.</td>
<td>B_PARITY</td>
<td>0x02</td>
</tr>
<tr>
<td>bit 2Indicates ROM; the kernel searches this for modules during start-up.</td>
<td>B_ROM</td>
<td>0x04</td>
</tr>
<tr>
<td>bit 3Non-volatile RAM; the kernel searches this for modules during start-up.</td>
<td>B_NVRAM</td>
<td>0x08</td>
</tr>
<tr>
<td>bit 4Shared memory.</td>
<td>B_SHARED</td>
<td>0x10</td>
</tr>
</tbody>
</table>

Only B_USERRAM memory can be initialized.
EditMod Labels

2-init module contents
14-memory list
1-Add additional item to list
n-memory list [n-1]
4-search block size

Description

This is the search block size. The kernel checks every search block size to see if RAM/ROM exists.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

0 to 65535
EditMod Labels

2-init module contents
14-memory list
1-Add additional item to list
n-memory list [n-1]
5-beginning address for this type

Description

This is the beginning address of the block as referenced by the CPU.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

0 to 4294967295
EditMod Labels

2-init module contents
14-memory list
1-Add additional item to list
n-memory list [n-1]
5-ending address + 1 for this type

Description

This is the ending address of the block as referenced by the CPU.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to INIT/config.des (Figure 2-3).

Available Values

0 to 4294967295
desc
Memory List Description

EditMod Labels
- 2-init module contents
- 14-memory list
- 1-Add additional item to list
- n-memory list [n-1]
- 6-memory list description

Description
This contains the memory list description name string.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as `\n` and `\012`).
**EditMod Labels**

2-init module contents
14-memory list
1-Add additional item to list
n-memory list [n-1]
7-translation address for dma’s

**Description**

External bus address of the beginning of the block. If 0, this field does not apply.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to INIT/config.des (Figure 2-3).

**Available Values**

0 to 4294967295

**Cachelist Fields**

The cachelist fields are in the order they appear during an interactive EditMod session. Defined fields may appear in a different order in the description files. The fields can be changed using the EditMod utility or by modifying the description files. See Init Module Field Configuration Options for detailed instructions on changing these fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blk_beg</td>
<td>Beginning address of memory region</td>
</tr>
<tr>
<td>blk_end</td>
<td>Ending address of memory region</td>
</tr>
</tbody>
</table>
**blk_beg**
Beginning Address of Memory Region

**EditMod Labels**
- 2-init module contents
- 33-cache list
- n-cache list[n-1]
- 1-beginning address of memory region

**Description**
This is the beginning address of the memory region.

**Port Generic Default Value**
0 (zero)

**Port Specific Override Value**
Refer to INIT/config.des (Figure 2-3).

**Available Values**
0 to 4294967295
blk_end
Ending Address of Memory Region

EditMod Labels
2-init module contents
33-cache list
n-cache list[n-1]
1-ending address + 1 of memory region

Description
This is the ending address of the memory region plus 1.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to INIT/config.des (Figure 2-3).

Available Values
0 to 4294967295
SCF Device Descriptors

SCF device descriptors contain configuration data specific to one serial device on an OS-9 system. Values that can be configured in the descriptor include:

- Device interrupt vector and priority
- Device I/O address
- Serial communication settings
- Special character mapping

The next section in this chapter provides a detailed example of the configuration options you can use to change configuration values in SCF descriptors.

The rest of this chapter provides a detailed list of all of the SCF device descriptor fields, including field descriptions and available values.

This chapter includes the following topics:

- SCF Field Configuration Options
- SCF Device Descriptor Field Reference
- Module Header Fields
- Device Descriptor Data Definition Fields
- SCF Description Block Fields
- SCF Logical Unit Static Storage Fields
- SCF Path Option Fields
SCF Field Configuration Options

To change an SCF device descriptor module configuration field, you can use either of the following methods:

1. Use the EditMod utility to directly modify existing SCF device descriptor modules either as a stand-alone module or as part of a merged module group (such as a boot image).
2. Modify the description file for the SCF device descriptor module and rebuild it using the makefile provided.

Direct Modification Advantages

The direct modification method has the following advantages:

- **Fast**  No source configuration file rebuilds are necessary.
- **Temporary**  The original module or merged-module group configuration can be easily restored through the appropriate rebuild.
- **Contained**  Changes are limited to the individual boot image modified (merged-module option).

Description File/Rebuild Advantages

The advantage of the description file/rebuild method is that the changes are permanent and reproducible. Modifications apply to all subsequent module rebuilds and to all merged-module groups built containing the updated module.

Both methods are documented in this section. These procedures are used with the field descriptions starting with the Module Header Fields. For direct modification, use the EditMod LABELS data to navigate the EditMod menus. The DESCRIPTION FILE MACRO data identifies the macro you need to define/modify in the configuration sources to rebuild the SCF device descriptor module.

Direct Modification

Use the EditMod utility and the following procedures to directly modify fields in the existing SCF device descriptor module. The module can stand-alone or it can be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The field references later in this chapter contain a description of each configurable field, its supported values, and the sequence of menu options required by EditMod to modify that field.

Refer to the Utilities Reference for a full description of EditMod's capabilities.
Chapter 3: SCF Device Descriptors

Direct Modification Procedures

To modify the stand-alone module, complete the following steps:
1. Go to the CMD$/BOOTOBJ$/DESC/<DEVICE> directory (see Figure 3-1).
2. Use EditMod to edit the module:
   $EditMod -e <descriptor>

To modify the module as part of a merged module group, complete the following steps:
1. Go to the BOOTS/SYSTEM$/PORTBOOT directory (see Figure 3-2).
2. Use EditMod to edit the module:
   $EditMod -e <descriptor> -f=<boot image name>

Refer to your board guide for information about how to modify the module lists and remake the boot images, and for specified boot image names.
3. Use the menu selections provided in the EditMod LABELS section of the field reference later in this chapter to locate the fields you want to edit.

4. Select a new value for the field from the AVAILABLE VALUES section of the field reference. Enter that value at the EditMod prompt to modify the field.

5. If you want to make additional modifications, use the p command (previous) to step backward through the EditMod menus. Repeat Steps 3 and 4 until you have made all desired modifications to the descriptor.

6. Select the w command (write) to save the changes.

7. Select the q command (quit) to exit EditMod.

Unless you modified the SCF device descriptors in your boot image, you should rebuild your boot image to include the new descriptor.

Example EditMod Session

This example modifies an SCF device descriptor as part of the boot image rom:

$ EditMod -e term -f=rom

1. module header
2. device descriptor data definitions
3. SCF description block
4. SCF logical unit static storage
5. SCF path options

$Which? [?/1-2/p/t/a/w/q] 4

Which? [?/1-5/p/t/a/w/q] 4

1. irq vector number : 0x4c
2. irq interrupt level : 0x0
3. irq polling priority : 5
4. polled input flag : 1
5. polled output flag : 1
6. driver accessible copy of logical unit number: 0x1
7. interrupt mask word : 0x80
8. send XOFF when buffer is this full : 246
9. size of input buffer : 256
10. input buffer
11. size of output buffer : 256
12. output buffer
13. lines left until end of page : 24
14. keyboard interrupt character : \x03
15. keyboard quit character : \x05
16. keyboard pause character : \x17
17. x-on character : \x11
18. x-off character : \x13
19. baud rate : 0xf
20. parity : 0
21. stop bits : 0
22. word size : 8
23. RTS state : 0

$Which? [?/1-6/p/t/a/w/q] 3
**Description File Modification**

You can use these procedures to modify the appropriate description file and rebuild the SCF device descriptors for your port directory. The DESCRIPTION FILE MACROS section of the field reference specifies the name of the macro you modify/define in the description files to configure the field. The value used in the define is chosen from the AVAILABLE VALUES specified for the field.

**Figure 3-3. Directory Location for Modifying the SCF Description File**

**Description File Modification Procedures**

1. Change to the SCF/<DEVICE> directory (see Figure 3-3).
2. Edit the file config.des and read the included comments for more information on using the specific description files provided in your software distribution. The config.des file contains a list of macro names that can be defined to override the global default values for the configuration fields.
3. Refer to the DESCRIPTION FILE MACRO section in the field reference later in this chapter to determine the macro name you define to configure the target field.
4. Read the comments in config.des to determine where to place the define for this macro.
5. Select the value you want to use to configure the field. See the AVAILABLE VALUES section of the field reference data for values or macros that can be used for the definition. Define the macro by entering a definition in the appropriate description files as follows:
   
   ```
   #define <macro> <value>
   ```
6. Save the changes and rebuild the SCF device descriptors, entering the following command in the SCF/<DEVICE>/DESC directory:
7. Rebuild your boot image to include the new descriptor.

SCF Device Descriptor Field Reference

This section contains a list of the most commonly configured fields in the SCF device descriptors. Each field entry contains the following information:

- `<Field name>` - The call name for each field that can be reconfigured in the module.
- `EditMod LABELS` - `EditMod` menu selections for navigating to the proper field in an `EditMod` session.
- `DESCRIPTION FILE MACRO` - The macro name you modify/define in the description file.
- `DESCRIPTION` - A brief description of the field's purpose and use.
- `EXAMPLE` - An optional example of the description file entry showing how to change the value of this field.
- `PORT GENERIC DEFAULT VALUE` - The value set in the port generic description file for this field. This is the value the field is assigned when the module is built, unless the appropriate macro has been defined in the port specific description file to override this default value.
- `PORT SPECIFIC OVERRIDE VALUE` - The value set in the port specific description file for this field. If defined, this is the value the field is assigned when the module is built, overriding the port generic default value.
- `AVAILABLE VALUES` - Values to which the field can be set through `EditMod` or the description files. In many cases, this data is presented in a table that maps a description of the value to a numeric value appropriate for entry in `EditMod`, and to a pre-defined macro available for use in the description file.

Module Header Fields

The following section contains the module header fields in the order they appear during an interactive `EditMod` session. Defined fields can appear in a different order in `config.des`.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>_m_group</td>
<td>MH_GROUP</td>
</tr>
<tr>
<td>_m_user</td>
<td>MH_USER</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------</td>
</tr>
<tr>
<td>mod_name</td>
<td>MH_NAME</td>
</tr>
<tr>
<td>m_access</td>
<td>MH_ACCESS</td>
</tr>
<tr>
<td>m_tylan</td>
<td>MH_TYLAN</td>
</tr>
<tr>
<td>m_attrev</td>
<td>MH_ATTREV</td>
</tr>
<tr>
<td>m_edit</td>
<td>MH_EDITION</td>
</tr>
</tbody>
</table>
EditMod Labels
1-module header
1-module owner’s group number

Description
Group ID of the module’s owner. The group number allows people working in the same department or on the same project to share a common identification number.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 65535
EditMod Labels
1-module header
2-module owner’s user number

Description
User ID of the module’s owner. The user number identifies a specific user.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 65535
EditMod Labels
1-module header
3-module name

Description
Contains the module name string.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels

1-module header
4-access permissions

Description

Defines the permissible module access by its owner or by other users.

Port Generic Default Value

Macro

\[
\text{MP\_OWNER\_READ} \mid \text{MP\_OWNER\_EXEC} \mid \text{MP\_GROUP\_READ} \mid \\
\text{MP\_GROUP\_EXEC} \mid \text{MP\_WORLD\_READ} \mid \text{MP\_WORLD\_EXEC}
\]

EditMod

0x555

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

Module access permission values are located in the header file, module.h, and are listed in Table 3-2.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read permission by owner</td>
<td>MP_OWNER_READ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Write permission by owner</td>
<td>MP_OWNER_WRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Execute permission by owner</td>
<td>MP_OWNER_EXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Owner permission mask</td>
<td>MP_OWNER_MASK</td>
<td>0x000f</td>
</tr>
<tr>
<td>Read permission by group</td>
<td>MP_GROUP_READ</td>
<td>0x0010</td>
</tr>
<tr>
<td>Write permission by group</td>
<td>MP_GROUP_WRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Execute permission by group</td>
<td>MP_GROUP_EXEC</td>
<td>0x0040</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Group permission mask</td>
<td>MP_GROUP_MASK</td>
<td>0x00F0</td>
</tr>
<tr>
<td>Read permission by world</td>
<td>MP_WORLD_READ</td>
<td>0x0100</td>
</tr>
<tr>
<td>Write permission by world</td>
<td>MP_WORLD_WRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Execute permission by world</td>
<td>MP_WORLD_EXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>World permission mask</td>
<td>MP_WORLD_MASK</td>
<td>0x0F00</td>
</tr>
<tr>
<td>All permissions for owner, group, and world</td>
<td>MP_WORLD_ACCESS</td>
<td>0x0777</td>
</tr>
<tr>
<td>System permission mask</td>
<td>MP_SYSTM_MASK</td>
<td>0xF000</td>
</tr>
</tbody>
</table>
EditMod Labels

1. module header
5. type/language

Description
Contains the module’s type (first byte) and language (second byte). The language codes indicate if the module is executable and which language the run-time system requires for execution, if any.

Port Generic Default Value
Macro

\[(MT\_DATA<<8) + ML\_OBJECT\]

EditMod

0x401

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Module type values and language codes are located in the header file, module.h, and are listed in Table 3-3 and Table 3-4.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not used (wildcard value in system calls)</td>
<td>MT_ANY</td>
<td>0x0000</td>
</tr>
<tr>
<td>Program module</td>
<td>MT_PROGRAM</td>
<td>0x0001</td>
</tr>
<tr>
<td>Subroutine module</td>
<td>MT_SUBROUT</td>
<td>0x0002</td>
</tr>
<tr>
<td>Multi-module (reserved for future use)</td>
<td>MT_MULTI</td>
<td>0x0003</td>
</tr>
<tr>
<td>Data module</td>
<td>MT_DATA</td>
<td>0x0004</td>
</tr>
<tr>
<td>Configuration data block data module</td>
<td>MT_CDBDATA</td>
<td>0x0005</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0xb-0xa</td>
<td>0xb-0xa</td>
</tr>
</tbody>
</table>
### Table 3-3. m_tylan Available Module Type Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>User trap library</td>
<td>MT_TRAPLIB</td>
<td>0x000b</td>
</tr>
<tr>
<td>System module</td>
<td>MT_SYSTEM</td>
<td>0x000c</td>
</tr>
<tr>
<td>File manager module</td>
<td>MT_FILEMAN</td>
<td>0x000d</td>
</tr>
<tr>
<td>Physical device driver</td>
<td>MT_DEVDRVR</td>
<td>0x000e</td>
</tr>
<tr>
<td>Device descriptor module</td>
<td>MT_DEVDESC</td>
<td>0x000f</td>
</tr>
<tr>
<td>User definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module type mask</td>
<td>MT_MASK</td>
<td>0xf00</td>
</tr>
</tbody>
</table>

### Table 3-4. m_tylan Available Language Code Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unspecified language (wildcard in system calls)</td>
<td>ML_ANY</td>
<td>0x0</td>
</tr>
<tr>
<td>Machine language</td>
<td>ML_OBJECT</td>
<td>0x1</td>
</tr>
<tr>
<td>Basic I-code (reserved for future use)</td>
<td>ML_ICODE</td>
<td>0x2</td>
</tr>
<tr>
<td>Pascal P-code (reserved for future use)</td>
<td>ML_PCODE</td>
<td>0x3</td>
</tr>
<tr>
<td>C I-code (reserved for future use)</td>
<td>ML_CCODE</td>
<td>0x4</td>
</tr>
<tr>
<td>Cobol I-code (reserved for future use)</td>
<td>ML_CBLCODE</td>
<td>0x5</td>
</tr>
<tr>
<td>Fortran</td>
<td>ML_FRTNCODE</td>
<td>0x6</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0x7-0xf</td>
<td>0x7-0xf</td>
</tr>
<tr>
<td>User-definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module language mask</td>
<td>ML_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
Chapter 3: SCF Device Descriptors

EditMod Labels

1-module header

6-revision/attributes

Description

Contains the module's attributes (first byte) and revision (second byte).

Port Generic Default Value

Macro

MA_REENT<<8

EditMod

0x8000

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

Module attribute and revision codes are located in the header file module.h, and are listed in Table 3-5.

If two modules with the same name are found in the memory search or are loaded into the current module directory, only the module with the highest revision level is kept. This enables easy substitution of modules for update or correction.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is re-entrant (shareable by multiple tasks).</td>
<td>MA_REENT</td>
<td>0x80 (shifted left to first byte: 0x8000)</td>
</tr>
<tr>
<td>(shifted left to first byte: MA_REENT&lt;&lt;8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The module is sticky. A sticky module is not removed from memory until its link count becomes -1 or memory is required for another use.</td>
<td>MA_GHOST</td>
<td>0x40 (shifted left to first byte: 0x4000)</td>
</tr>
<tr>
<td>(shifted left to first byte: MA_GHOST&lt;&lt;8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>----------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>The module is a system-state module.</td>
<td>MA_SUPER</td>
<td>0x20 (shifted left to first byte: 0x2000)</td>
</tr>
<tr>
<td></td>
<td>(shifted left to first byte: MA_SUPER&lt;&lt;8)</td>
<td></td>
</tr>
<tr>
<td>User-definable revision number</td>
<td>0x0-0xfe</td>
<td>0x0 - 0xfe</td>
</tr>
<tr>
<td>Module attribute mask</td>
<td>MA_MASK</td>
<td>0xff00</td>
</tr>
<tr>
<td>Module revision mask</td>
<td>MR_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
7-edition

Description

Indicates the software release level for maintenance. OS-9 does not use this field. Whenever a program is revised (even for a small change), increase this number. We recommend internal documentation within the source program be keyed to this system.

Port Generic Default Value

1

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

0 to 65535

Device Descriptor Data Definition Fields

The following section contains the device descriptor data definition fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd_port</td>
<td>PORTADDR</td>
</tr>
<tr>
<td>dd_lun</td>
<td>LUN</td>
</tr>
<tr>
<td>dd_pd_size</td>
<td>PD_SIZE</td>
</tr>
<tr>
<td>dd_type</td>
<td>DD_TYPE</td>
</tr>
<tr>
<td>dd_mode</td>
<td>DD_MODE</td>
</tr>
<tr>
<td>fmgr_name</td>
<td>FMGR_NAME</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------</td>
</tr>
<tr>
<td>drvr_name</td>
<td>DRVR_NAME</td>
</tr>
<tr>
<td>dd_class</td>
<td>DD_CLASS</td>
</tr>
</tbody>
</table>
EditMod Labels
2-device descriptor data definitions
1-device port address

Description
Absolute physical address of the hardware controller. This is the address of the device on the bus. This is the lowest address the device has mapped. Port address is hardware dependent.

Macro Example
#define PORTADDR 0xfffe4000

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
2-device descriptor data definitions
2-logical unit number

Description
Distinguishes the different devices driven from a unique controller. Each unique number represents a different logical unit static storage area.

Macro Example
#define LUN 2

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 65535
dd_pd_size
PD_SIZE

EditMod Labels
2-device descriptor data definitions
3-path descriptor size

Description
Size of the path descriptor. IOMAN uses this value when it allocates a path descriptor.

Port Generic Default Value
0x234

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 65535
EditMod Labels
2-device descriptor data definitions
4-device type

Description
Identifies the I/O class of the device.

Port Generic Default Value
Macro
  DT_SCF

EditMod
  0x0

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Device type values are defined in the header file io.h, and are listed in Table 3-7.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Character File Type</td>
<td>DT_SCF</td>
<td>0x0</td>
</tr>
<tr>
<td>Random Block File Type</td>
<td>DT_RBF</td>
<td>0x1</td>
</tr>
<tr>
<td>Pipe File Type</td>
<td>DT_PIPE</td>
<td>0x2</td>
</tr>
<tr>
<td>Sequential Block File Type</td>
<td>DT_SBF</td>
<td>0x3</td>
</tr>
<tr>
<td>Network File Type</td>
<td>DT_NFM</td>
<td>0x4</td>
</tr>
<tr>
<td>Compact Disc File Type</td>
<td>DT_CDFM</td>
<td>0x5</td>
</tr>
<tr>
<td>User Communication Manager</td>
<td>DT_UCM</td>
<td>0x6</td>
</tr>
<tr>
<td>Socket Communication Manager</td>
<td>DT_SOCK</td>
<td>0x7</td>
</tr>
</tbody>
</table>
### Table 3-7. *dd_type* Available Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo-Keyboard Manager</td>
<td>DT_PTTY</td>
<td>0x8</td>
</tr>
<tr>
<td>Graphics File Manager</td>
<td>DT_GFM</td>
<td>0x9</td>
</tr>
<tr>
<td>Inet File Manager</td>
<td>DT_INET</td>
<td>0x10</td>
</tr>
<tr>
<td>Multi-media File Manager</td>
<td>DT_MFM</td>
<td>0x11</td>
</tr>
<tr>
<td>Generic Device File Manager</td>
<td>DT_DVM</td>
<td>0x12</td>
</tr>
<tr>
<td>Null File Manager</td>
<td>DT_NULL</td>
<td>0x13</td>
</tr>
<tr>
<td>DVD File Manager</td>
<td>DT_DVDFM</td>
<td>0x14</td>
</tr>
<tr>
<td>Module Directory File System Manager</td>
<td>DT_MODFM</td>
<td>0x15</td>
</tr>
<tr>
<td>PC-DOS File Manager</td>
<td>DT_PCF</td>
<td>0xa</td>
</tr>
<tr>
<td>Non-volatile RAM File Manager</td>
<td>DT_NRF</td>
<td>0xb</td>
</tr>
<tr>
<td>ISDN File Manager</td>
<td>DT_ISDN</td>
<td>0xc</td>
</tr>
<tr>
<td>MPFM File Manager</td>
<td>DT_MPFM</td>
<td>0xd</td>
</tr>
<tr>
<td>Real-Time Network File Manager</td>
<td>DT_RTNFM</td>
<td>0xe</td>
</tr>
<tr>
<td>Serial Protocol File Manager</td>
<td>DT_SPF</td>
<td>0xf</td>
</tr>
<tr>
<td>Reserved for Microware Use Only</td>
<td>17-127</td>
<td>0xa1-0x7f</td>
</tr>
</tbody>
</table>
EditMod Labels

2-device descriptor data definitions
5-device mode capabilities

Description

Used to check the validity of a caller’s access mode byte in I_CREATE or I_OPEN system calls. If a bit is set, the device can perform the corresponding function. The S_ISIZE bit is usually set, because it is handled by the file manager or ignored. If the S_ISHARE bit is set, the device is non-sharable. A printer is an example of a non-sharable device.

Port Generic Default Value

Macro

\begin{verbatim}
S_ISIZE | S_IREAD | S_IWRITE
\end{verbatim}

EditMod

0x2003

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

The file access modes are defined in the header file, modes.h, and located in Table 3-8. The file access permission values are defined in the header file modes.h and in Table 3-9.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|}
\hline
Description & Macro & EditMod \\
\hline
Truncate on open & S_ITRUNC & 0x0100 \\
Ensure contiguous file & S_ICONTIG & 0x0400 \\
Error if file exists on create & S_IEXCL & 0x0400 \\
Create file & S_ICREAT & 0x0800 \\
\hline
\end{tabular}
\caption{dd_mode Available Values for File Access Modes}
\end{table}
### Table 3-8. \texttt{dd_mode} Available Values for File Access Modes (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append to file</td>
<td>S_IAPPEND</td>
<td>0x1000</td>
</tr>
<tr>
<td>Non-sharable</td>
<td>S_ISHARE</td>
<td>0x4000</td>
</tr>
</tbody>
</table>

### Table 3-9. \texttt{dd_mode} Available Values for File Access Permissions

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask for permission bits</td>
<td>S_IPRM</td>
<td>0xffff</td>
</tr>
<tr>
<td>Owner read</td>
<td>S_IREAD</td>
<td>0x0001</td>
</tr>
<tr>
<td>Owner write</td>
<td>S_IWRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Owner execute</td>
<td>S_IEXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Search permission</td>
<td>S_ISEARCH</td>
<td>0x0004</td>
</tr>
<tr>
<td>Group read</td>
<td>S_IGREAD</td>
<td>0x0010</td>
</tr>
<tr>
<td>Group write</td>
<td>S_IGWRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Group execute</td>
<td>S_IGEXEC</td>
<td>0x0040</td>
</tr>
<tr>
<td>Group search</td>
<td>S_IGSEARCH</td>
<td>0x0040</td>
</tr>
<tr>
<td>Public read</td>
<td>S IOREAD</td>
<td>0x0100</td>
</tr>
<tr>
<td>Public write</td>
<td>S_IOWRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Public execute</td>
<td>S_IOEXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>Public search</td>
<td>S_IOSEARCH</td>
<td>0x0400</td>
</tr>
</tbody>
</table>
EditMod Labels
2-device descriptor data definitions
6-file manager name

Description
Contains the name string of the file manager module to use.

Port Generic Default Value
“scf”

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
Chapter 3: SCF Device Descriptors

EditMod Labels
2-device descriptor data definitions
7-driver name

Description
Contains the name string of the device driver module to use.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**EditMod Labels**

2-device descriptor data definitions
8-device class (sequential or random)

**Description**

Used to identify the class of the device, whether it is random or sequential access.

**Port Generic Default Value**

Macro

\[ \text{DC_SEQ} \]

**EditMod**

0x1

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

Device class available values are defined in the header file, io.h, and in Table 3-10.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential access device</td>
<td>DC_SEQ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Random access device</td>
<td>DC_RND</td>
<td>0x0002</td>
</tr>
</tbody>
</table>

**SCF Description Block Fields**

The following section contains the SCF description block fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>outdev_name</td>
<td>OUTDEVNAME</td>
</tr>
</tbody>
</table>
**EditMod Labels**

3-SCF description block
1-output device name

**Description**

**Macro Example**

**Port Generic Default Value**

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

**SCF Logical Unit Static Storage Fields**

The following section contains the SCF logical unit static storage fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

### Table 3-12. Device Descriptor Data Definition Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>hardware_vector</td>
<td>VECTOR</td>
</tr>
<tr>
<td>v_irqlevel</td>
<td>IRQLEVEL</td>
</tr>
<tr>
<td>v_priority</td>
<td>PRIORITY</td>
</tr>
<tr>
<td>v_pollin</td>
<td>INPUT_TYPE</td>
</tr>
<tr>
<td>v_pollout</td>
<td>OUTPUT_TYPE</td>
</tr>
<tr>
<td>v_lun</td>
<td>LUN</td>
</tr>
</tbody>
</table>
### Table 3-12. Device Descriptor Data Definition Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>v_irqmask</td>
<td>IRQ_MASK</td>
</tr>
<tr>
<td>v_maxbuff</td>
<td>MAXBUFF</td>
</tr>
<tr>
<td>v_insize</td>
<td>INSIZE</td>
</tr>
<tr>
<td>v_outsize</td>
<td>OUTSIZE</td>
</tr>
<tr>
<td>v_line</td>
<td>PAGE_SIZE</td>
</tr>
<tr>
<td>v_intr</td>
<td>KYBDINTR</td>
</tr>
<tr>
<td>v_quit</td>
<td>KYBDQUIT</td>
</tr>
<tr>
<td>v_psch</td>
<td>KYBDPAUSE</td>
</tr>
<tr>
<td>v_xon</td>
<td>XON</td>
</tr>
<tr>
<td>v_xoff</td>
<td>XOFF</td>
</tr>
<tr>
<td>v_baud</td>
<td>BAUDRATE</td>
</tr>
<tr>
<td>v_parity</td>
<td>LUPARITY</td>
</tr>
<tr>
<td>v_stopbits</td>
<td>STOPBITS</td>
</tr>
<tr>
<td>v_wordsize</td>
<td>WORDSIZE</td>
</tr>
<tr>
<td>v_rtsstate</td>
<td>RTSSTATE</td>
</tr>
<tr>
<td>v_devspec</td>
<td></td>
</tr>
</tbody>
</table>
EditMod Labels

4-SCF logical unit static storage
1-irq vector number

Description

This is the vector passed to the processor at interrupt time. Vector is hardware/software dependent. You can program some devices to produce different vectors. See your board guide for vector mappings for specific processors.

Macro Example

#define VECTOR 80

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

0 to 255
EditMod Labels

4-SCF logical unit static storage
2-irq interrupt level

Description

This is the hardware priority of the console device interrupt. When a device interrupts a processor, the level of the interrupt is used to mask lower priority interrupts.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

0 to 65535

The number of supported interrupt levels is dependent on the processor being used (for example, 1-7 on 680x0 type CPUs).
EditMod Labels
4-SCF logical unit static storage
3-irq polling priority

Description
This is the software (polling) priority for the console device on the IRQ polling table.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The interrupt priority value range is 0-65534 (65535 is reserved). A non-zero priority determines the position of the device within the vector. Lower values are polled first.

Some considerations to keep in mind when selecting an interrupt priority:
• A priority of 0 indicates the device desires exclusive use of the vector.
• If the priority is 1, it is polled first and no other device can have a priority of 1 on the vector. For all other priority values, more than one device can share the same priority on a vector. In this case, first-in, first-out (FIFO) scheduling determines the order of precedence in the polling table for the devices.
• OS-9 does not allow a device to claim exclusive use of a vector if another device has already been installed on the vector. Additionally, it does not allow another device to use the vector once the vector has been claimed for exclusive use.
• This value is software dependent.

See Also
F_IRQ system call entry in the OS-9 Technical Manual.
**v_pollin**

**INPUT_TYPE**

**EditMod Labels**

- 4-SCF logical unit static storage
- 4-polled input flag

**Description**

This specifies whether input on the device is interrupt driven or polled. If the device is operated in polled mode, SCF calls the driver's read routine for every character. This value is device dependent.

**Macro Example**

```
#define INPUT_TYPE IRQDRIVEN
```

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

Device input values are defined in the header file, scf.h, and in Table 3-13.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrupt driven</td>
<td>IRQDRIVEN</td>
<td>0x0000</td>
</tr>
<tr>
<td>Polled</td>
<td>POLLED</td>
<td>0x0001</td>
</tr>
</tbody>
</table>

Table 3-13. v_pollin Available Values
v_pollout
OUTPUT_TYPE

EditMod Labels
4-SCF logical unit static storage
5-polled output flag

Description
This specifies whether output on the device is interrupt driven or polled. If the device is operated in polled mode, SCF calls the driver’s write routine to transmit every character. This value is device dependent.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Device input values are defined in the header file, scf.h, and in Table 3-14.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrupt driven</td>
<td>IRQDRIVEN</td>
<td>0x0000</td>
</tr>
<tr>
<td>Polled</td>
<td>POLLED</td>
<td>0x0001</td>
</tr>
</tbody>
</table>

Table 3-14. v_pollout Available Values
EditMod Labels
4-SCF logical unit static storage
6-driver accessible copy of logical unit number

Description
Since more than one device may have the same port address, the logical unit number distinguishes the devices having the same port address.

Macro Example
#define LUN  2

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 65535
EditMod Labels

4-SCF logical unit static storage
7-interrupt mask word

Description
This is the interrupt mask for the SCF device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 65535
EditMod Labels
4-SCF logical unit static storage
8-send XOFF when buffer is this full

Description
This specifies the device to send on XOFF when the buffer is full, in bytes.

Port Generic Default Value
246

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 65535
EditMod Labels
4-SCF logical unit static storage
9-size of input buffer

Description
This specifies the size of the input buffer for the logical unit.

Port Generic Default Value
256

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 65535
**EditMod Labels**

4-SCF logical unit static storage

11-size of output buffer

**Description**

This specifies the size of the output buffer for the logical unit.

**Port Generic Default Value**

256

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

0 to 65535
**v_line**

**PAGE_SIZE**

**EditMod Labels**
4-SCF logical unit static storage

13-lines left until end of page

**Description**
This specifies the number of lines per screen (or page).

**Port Generic Default Value**
24

**Port Specific Override Value**
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**
0 to 65535
EditMod Labels
4-SCF logical unit static storage
14-keyboard interrupt character

Description
This specifies the control key to use for the keyboard interrupt function.

Port Generic Default Value
Macro
CTRL_C

EditMod
0x03

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The ASCII control and special characters are defined in the header file, scf.h, and in Table 3-15.

Table 3-15. ASCII Control Character Available Values

<table>
<thead>
<tr>
<th>SCF/OS-9 Compatible Standard Codes</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_NULL</td>
<td>C_NULL</td>
<td>0x00</td>
</tr>
<tr>
<td>C_REPEAT</td>
<td>CTRL_A</td>
<td>0x01</td>
</tr>
<tr>
<td>C_INTR</td>
<td>CTRL_B</td>
<td>0x02</td>
</tr>
<tr>
<td>C_REPRINT</td>
<td>CTRL_C</td>
<td>0x03</td>
</tr>
<tr>
<td>C_QUIT</td>
<td>CTRL_D</td>
<td>0x04</td>
</tr>
<tr>
<td>C_BELL</td>
<td>CTRL_E</td>
<td>0x05</td>
</tr>
<tr>
<td></td>
<td>CTRL_F</td>
<td>0x06</td>
</tr>
<tr>
<td></td>
<td>CTRL_G</td>
<td>0x07</td>
</tr>
</tbody>
</table>
Table 3-15. ASCII Control Character Available Values (Continued)

<table>
<thead>
<tr>
<th>SCF/OS-9 Compatible Standard Codes</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_BACKSPACE</td>
<td>CTRL_H</td>
<td>0x08</td>
</tr>
<tr>
<td>C_TAB</td>
<td>CTRL_I</td>
<td>0x09</td>
</tr>
<tr>
<td>C_LINEFEED</td>
<td>CTRL_J</td>
<td>0x0A</td>
</tr>
<tr>
<td></td>
<td>CTRL_K</td>
<td>0x0B</td>
</tr>
<tr>
<td>C_FORMFEED</td>
<td>CTRL_L</td>
<td>0x0C</td>
</tr>
<tr>
<td>C_CR</td>
<td>CTRL_M</td>
<td>0x0D</td>
</tr>
<tr>
<td></td>
<td>CTRL_N</td>
<td>0x0E</td>
</tr>
<tr>
<td></td>
<td>CTRL_O</td>
<td>0x0F</td>
</tr>
<tr>
<td></td>
<td>CTRL_P</td>
<td>0x10</td>
</tr>
<tr>
<td>C_XON</td>
<td>CTRL_Q</td>
<td>0x11</td>
</tr>
<tr>
<td></td>
<td>CTRL_R</td>
<td>0x12</td>
</tr>
<tr>
<td>C_XOFF</td>
<td>CTRL_S</td>
<td>0x13</td>
</tr>
<tr>
<td></td>
<td>CTRL_T</td>
<td>0x14</td>
</tr>
<tr>
<td></td>
<td>CTRL_U</td>
<td>0x15</td>
</tr>
<tr>
<td></td>
<td>CTRL_V</td>
<td>0x16</td>
</tr>
<tr>
<td>C_PAUSE</td>
<td>CTRL_W</td>
<td>0x17</td>
</tr>
<tr>
<td>C_DELLINE</td>
<td>CTRL_X</td>
<td>0x18</td>
</tr>
<tr>
<td></td>
<td>CTRL_Y</td>
<td>0x19</td>
</tr>
<tr>
<td></td>
<td>CTRL_Z</td>
<td>0x1A</td>
</tr>
<tr>
<td></td>
<td>CTRL_SPACE</td>
<td>0x20</td>
</tr>
<tr>
<td></td>
<td>CTRL_COMMA</td>
<td>0x2C</td>
</tr>
<tr>
<td></td>
<td>CTRL_PERIOD</td>
<td>0x2E</td>
</tr>
</tbody>
</table>
### Table 3-15. ASCII Control Character Available Values (Continued)

<table>
<thead>
<tr>
<th>SCF/OS-9 Compatible Standard Codes</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRL_SLASH</td>
<td>0x2F</td>
<td></td>
</tr>
<tr>
<td>C_EOF</td>
<td>0x1B</td>
<td></td>
</tr>
</tbody>
</table>
EditMod Labels
4-SCF logical unit static storage
15-keyboard quit character

Description
This specifies the control key to use for the keyboard quit function.

Port Generic Default Value
Macro
CTRL_E

EditMod
0x05

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
See Table 3-15.
EditMod Labels
4-SCF logical unit static storage
16-keyboard pause character

Description
This specifies the control key to use for the keyboard pause function.

Port Generic Default Value
Macro
CTRL_W

EditMod
0x17

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
See Table 3-15.
**EditMod Labels**

4-SCF logical unit static storage
17-x-on character

**Description**

This specifies the control key to use for the X-ON protocol function.

**Port Generic Default Value**

Macro

```c
CTRL_Q
```

EditMod

0x11

**Port Specific Override Value**

Refer to `SCF/<DEVICE>/DESC/config.des` ([Figure 3-3](#)).

**Available Values**

See Table 3-15.
EditMod Labels
4-SCF logical unit static storage
18-x-off character

Description
This specifies the control key to use for the X-OFF protocol function.

Port Generic Default Value
Macro
CTRL_S

EditMod
0x13

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
See Table 3-15.
**EditMod Labels**

4-SCF logical unit static storage
19-baud rate

**Description**

This specifies the baud rate of the device.

**Port Generic Default Value**

Macro

9600

EditMod

0xf

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The SCF device descriptor baud rate values are located in the header file, scf.h, and are listed in Table 3-16.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardwire baud rate</td>
<td>HARDWIRE</td>
<td>0x00</td>
</tr>
<tr>
<td>50 bits per second (bps)</td>
<td>BAUD50</td>
<td>0x01</td>
</tr>
<tr>
<td>75 bps</td>
<td>BAUD75</td>
<td>0x02</td>
</tr>
<tr>
<td>110 bps</td>
<td>BAUD110</td>
<td>0x03</td>
</tr>
<tr>
<td>134.5 bps</td>
<td>BAUD134P5</td>
<td>0x04</td>
</tr>
<tr>
<td>150 bps</td>
<td>BAUD150</td>
<td>0x05</td>
</tr>
<tr>
<td>300 bps</td>
<td>BAUD300</td>
<td>0x06</td>
</tr>
<tr>
<td>600 bps</td>
<td>BAUD600</td>
<td>0x07</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>1200 bps</td>
<td>BAUD1200</td>
<td>0x08</td>
</tr>
<tr>
<td>1800 bps</td>
<td>BAUD1800</td>
<td>0x09</td>
</tr>
<tr>
<td>2000 bps</td>
<td>BAUD2000</td>
<td>0x0A</td>
</tr>
<tr>
<td>2400 bps</td>
<td>BAUD2400</td>
<td>0x0B</td>
</tr>
<tr>
<td>3600 bps</td>
<td>BAUD3600</td>
<td>0x0C</td>
</tr>
<tr>
<td>4800 bps</td>
<td>BAUD4800</td>
<td>0x0D</td>
</tr>
<tr>
<td>7200 bps</td>
<td>BAUD7200</td>
<td>0x0E</td>
</tr>
<tr>
<td>9600 bps</td>
<td>BAUD9600</td>
<td>0x0F</td>
</tr>
<tr>
<td>19,200 bps</td>
<td>BAUD19200</td>
<td>0x10</td>
</tr>
<tr>
<td>31,250 bps</td>
<td>BAUD31250</td>
<td>0x11</td>
</tr>
<tr>
<td>38,400 bps</td>
<td>BAUD38400</td>
<td>0x12</td>
</tr>
<tr>
<td>56,000 bps</td>
<td>BAUD56000</td>
<td>0x13</td>
</tr>
<tr>
<td>57,600 bps</td>
<td>BAUD57600</td>
<td>0x14</td>
</tr>
<tr>
<td>64,000 bps</td>
<td>BAUD64000</td>
<td>0x15</td>
</tr>
<tr>
<td>115,200 bps</td>
<td>BAUD115200</td>
<td>0x16</td>
</tr>
</tbody>
</table>
**EditMod Labels**

4-SCF logical unit static storage
20-parity

**Description**

This specifies the parity mode of the device.

**Port Generic Default Value**

Macro

NOPARITY

EditMod

0

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

Parity modes are defined in the header file, scf.h, and Table 3-17.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>No parity</td>
<td>NOPARITY</td>
<td>0x00</td>
</tr>
<tr>
<td>Odd parity</td>
<td>ODDPARITY</td>
<td>0x01</td>
</tr>
<tr>
<td>Even parity</td>
<td>EVENPARITY</td>
<td>0x02</td>
</tr>
<tr>
<td>Mark parity</td>
<td>MARKPARITY</td>
<td>0x03</td>
</tr>
<tr>
<td>Space parity</td>
<td>SPACEPARITY</td>
<td>0x04</td>
</tr>
</tbody>
</table>

**Table 3-17. v_parity Available Values**
EditMod Labels

4-SCF logical unit static storage
21-stop bits

Description
This specifies the number of stop bits to be used for transmission.

Port Generic Default Value
Macro
ONESTOP

EditMod
0

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
See Table 3-18.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop bit length of 1</td>
<td>ONESTOP</td>
<td>0x0</td>
</tr>
<tr>
<td>Stop bit length of 1.5</td>
<td>ONE_5STOP</td>
<td>0x1</td>
</tr>
<tr>
<td>Stop bit length of 2</td>
<td>TWO_STOP</td>
<td>0x2</td>
</tr>
</tbody>
</table>
**EditMod Labels**

4-SCF logical unit static storage
22-word size

**Description**

This specifies the number of bits per character to be used for transmission.

**Port Generic Default Value**

Macro

```c
WORDSIZE8
```

**Port Specific Override Value**

Refer to `SCF/<DEVICE>/DESC/config.des` (Figure 3-3).

**Available Values**

Word size values are located in the header file, `scf.h`, and are listed in Table 3-19.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 bits per character</td>
<td>WORDSIZE5</td>
<td>0x5</td>
</tr>
<tr>
<td>6 bits per character</td>
<td>WORDSIZE6</td>
<td>0x6</td>
</tr>
<tr>
<td>7 bits per character</td>
<td>WORDSIZE7</td>
<td>0x7</td>
</tr>
<tr>
<td>8 bits per character</td>
<td>WORDSIZE8</td>
<td>0x8</td>
</tr>
</tbody>
</table>
EditMod Labels
4-SCF logical unit static storage
23-RTS state

Description
This determines the state of the Request to Send (RTS) line for hardware handshaking.

Port Generic Default Value
Macro
RTSDISABLED

EditMod
0

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The Request to Send (RTS) state values are defined in the header file, scf.h, and in Table 3-20.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>rts disabled</td>
<td>RTSDISABLED</td>
<td>0x0</td>
</tr>
<tr>
<td>rts enabled</td>
<td>RTSENABLED</td>
<td>0x1</td>
</tr>
</tbody>
</table>

Table 3-20. v_rtsstate Available Values
EditMod Labels
4-SCF logical unit static storage
24-<device specific storage label> (optional)

Description
Optional device specific information structure. Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3) to determine if structure exists, and if so, the available fields.

SCF Path Option Fields
The following section contains the SCF path option fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

Table 3-21. Device Descriptor Data Definition Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>pd_inmap0type</td>
<td>TYPE0x7f</td>
</tr>
<tr>
<td>pd_inmap0func_code</td>
<td>FUNC0x7f</td>
</tr>
<tr>
<td>pd_inmap0size</td>
<td>SIZE0x7f</td>
</tr>
<tr>
<td>pd_inmap0string</td>
<td>STRING0x7f</td>
</tr>
<tr>
<td>pd_inmap1type</td>
<td>TYPE0x01</td>
</tr>
<tr>
<td>pd_inmap1func_code</td>
<td>FUNC0x01</td>
</tr>
<tr>
<td>pd_inmap1size</td>
<td>SIZE0x01</td>
</tr>
<tr>
<td>pd_inmap1string</td>
<td>STRING0x01</td>
</tr>
<tr>
<td>pd_inmap2type</td>
<td>TYPE0x02</td>
</tr>
<tr>
<td>pd_inmap2func_code</td>
<td>FUNC0x02</td>
</tr>
<tr>
<td>pd_inmap2size</td>
<td>SIZE0x02</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>pd_inmap2string</td>
<td>STRING0x02</td>
</tr>
<tr>
<td>pd_inmap3type</td>
<td>TYPE0x03</td>
</tr>
<tr>
<td>pd_inmap3func_code</td>
<td>FUNC0x03</td>
</tr>
<tr>
<td>pd_inmap3size</td>
<td>SIZE0x03</td>
</tr>
<tr>
<td>pd_inmap3string</td>
<td>STRING0x03</td>
</tr>
<tr>
<td>pd_inmap4type</td>
<td>TYPE0x04</td>
</tr>
<tr>
<td>pd_inmap4func_code</td>
<td>FUNC0x04</td>
</tr>
<tr>
<td>pd_inmap4size</td>
<td>SIZE0x04</td>
</tr>
<tr>
<td>pd_inmap4string</td>
<td>STRING0x04</td>
</tr>
<tr>
<td>pd_inmap5type</td>
<td>TYPE0x05</td>
</tr>
<tr>
<td>pd_inmap5func_code</td>
<td>FUNC0x05</td>
</tr>
<tr>
<td>pd_inmap5size</td>
<td>SIZE0x05</td>
</tr>
<tr>
<td>pd_inmap5string</td>
<td>STRING0x05</td>
</tr>
<tr>
<td>pd_inmap6type</td>
<td>TYPE0x06</td>
</tr>
<tr>
<td>pd_inmap6func_code</td>
<td>FUNC0x06</td>
</tr>
<tr>
<td>pd_inmap6size</td>
<td>SIZE0x06</td>
</tr>
<tr>
<td>pd_inmap6string</td>
<td>STRING0x06</td>
</tr>
<tr>
<td>pd_inmap7type</td>
<td>TYPE0x07</td>
</tr>
<tr>
<td>pd_inmap7func_code</td>
<td>FUNC0x07</td>
</tr>
<tr>
<td>pd_inmap7size</td>
<td>SIZE0x07</td>
</tr>
<tr>
<td>pd_inmap7string</td>
<td>STRING0x07</td>
</tr>
<tr>
<td>pd_inmap8type</td>
<td>TYPE0x08</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>pd_inmap8func_code</td>
<td>FUNC0x08</td>
</tr>
<tr>
<td>pd_inmap8size</td>
<td>SIZE0x08</td>
</tr>
<tr>
<td>pd_inmap8string</td>
<td>STRING0x08</td>
</tr>
<tr>
<td>pd_inmap9type</td>
<td>TYPE0x09</td>
</tr>
<tr>
<td>pd_inmap9func_code</td>
<td>FUNC0x09</td>
</tr>
<tr>
<td>pd_inmap9size</td>
<td>SIZE0x09</td>
</tr>
<tr>
<td>pd_inmap9string</td>
<td>STRING0x09</td>
</tr>
<tr>
<td>pd_inmap10type</td>
<td>TYPE0x0a</td>
</tr>
<tr>
<td>pd_inmap10func_code</td>
<td>FUNC0x0a</td>
</tr>
<tr>
<td>pd_inmap10size</td>
<td>SIZE0x0a</td>
</tr>
<tr>
<td>pd_inmap10string</td>
<td>STRING0x0a</td>
</tr>
<tr>
<td>pd_inmap11type</td>
<td>TYPE0x0b</td>
</tr>
<tr>
<td>pd_inmap11func_code</td>
<td>FUNC0x0b</td>
</tr>
<tr>
<td>pd_inmap11size</td>
<td>SIZE0x0b</td>
</tr>
<tr>
<td>pd_inmap11string</td>
<td>STRING0x0b</td>
</tr>
<tr>
<td>pd_inmap12type</td>
<td>TYPE0x0c</td>
</tr>
<tr>
<td>pd_inmap12func_code</td>
<td>FUNC0x0c</td>
</tr>
<tr>
<td>pd_inmap12size</td>
<td>SIZE0x0c</td>
</tr>
<tr>
<td>pd_inmap12string</td>
<td>STRING0x0c</td>
</tr>
<tr>
<td>pd_inmap13type</td>
<td>TYPE0x0d</td>
</tr>
<tr>
<td>pd_inmap13func_code</td>
<td>FUNC0x0d</td>
</tr>
<tr>
<td>pd_inmap13size</td>
<td>SIZE0x0d</td>
</tr>
</tbody>
</table>
### Table 3-21. Device Descriptor Data Definition Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>pd_inmap13string</td>
<td>STRING0x0d</td>
</tr>
<tr>
<td>pd_inmap14type</td>
<td>TYPE0x0e</td>
</tr>
<tr>
<td>pd_inmap14func_code</td>
<td>FUNC0x0e</td>
</tr>
<tr>
<td>pd_inmap14size</td>
<td>SIZE0x0e</td>
</tr>
<tr>
<td>pd_inmap14string</td>
<td>STRING0x0e</td>
</tr>
<tr>
<td>pd_inmap15type</td>
<td>TYPE0x0f</td>
</tr>
<tr>
<td>pd_inmap15func_code</td>
<td>FUNC0x0f</td>
</tr>
<tr>
<td>pd_inmap15size</td>
<td>SIZE0x0f</td>
</tr>
<tr>
<td>pd_inmap15string</td>
<td>STRING0x0f</td>
</tr>
<tr>
<td>pd_inmap16type</td>
<td>TYPE0x10</td>
</tr>
<tr>
<td>pd_inmap16func_code</td>
<td>FUNC0x10</td>
</tr>
<tr>
<td>pd_inmap16size</td>
<td>SIZE0x10</td>
</tr>
<tr>
<td>pd_inmap16string</td>
<td>STRING0x10</td>
</tr>
<tr>
<td>pd_inmap17type</td>
<td>TYPE0x11</td>
</tr>
<tr>
<td>pd_inmap17func_code</td>
<td>FUNC0x11</td>
</tr>
<tr>
<td>pd_inmap17size</td>
<td>SIZE0x11</td>
</tr>
<tr>
<td>pd_inmap17string</td>
<td>STRING0x11</td>
</tr>
<tr>
<td>pd_inmap18type</td>
<td>TYPE0x12</td>
</tr>
<tr>
<td>pd_inmap18func_code</td>
<td>FUNC0x12</td>
</tr>
<tr>
<td>pd_inmap18size</td>
<td>SIZE0x12</td>
</tr>
<tr>
<td>pd_inmap18string</td>
<td>STRING0x12</td>
</tr>
<tr>
<td>pd_inmap19type</td>
<td>TYPE0x13</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>pd_inmap19func_code</td>
<td>FUNC0x13</td>
</tr>
<tr>
<td>pd_inmap19size</td>
<td>SIZE0x13</td>
</tr>
<tr>
<td>pd_inmap19string</td>
<td>STRING0x13</td>
</tr>
<tr>
<td>pd_inmap20type</td>
<td>TYPE0x14</td>
</tr>
<tr>
<td>pd_inmap20func_code</td>
<td>FUNC0x14</td>
</tr>
<tr>
<td>pd_inmap20size</td>
<td>SIZE0x14</td>
</tr>
<tr>
<td>pd_inmap20string</td>
<td>STRING0x14</td>
</tr>
<tr>
<td>pd_inmap21type</td>
<td>TYPE0x15</td>
</tr>
<tr>
<td>pd_inmap21func_code</td>
<td>FUNC0x15</td>
</tr>
<tr>
<td>pd_inmap21size</td>
<td>SIZE0x15</td>
</tr>
<tr>
<td>pd_inmap21string</td>
<td>STRING0x15</td>
</tr>
<tr>
<td>pd_inmap22type</td>
<td>TYPE0x16</td>
</tr>
<tr>
<td>pd_inmap22func_code</td>
<td>FUNC0x16</td>
</tr>
<tr>
<td>pd_inmap22size</td>
<td>SIZE0x16</td>
</tr>
<tr>
<td>pd_inmap22string</td>
<td>STRING0x16</td>
</tr>
<tr>
<td>pd_inmap23type</td>
<td>TYPE0x17</td>
</tr>
<tr>
<td>pd_inmap23func_code</td>
<td>FUNC0x17</td>
</tr>
<tr>
<td>pd_inmap23size</td>
<td>SIZE0x17</td>
</tr>
<tr>
<td>pd_inmap23string</td>
<td>STRING0x17</td>
</tr>
<tr>
<td>pd_inmap24type</td>
<td>TYPE0x18</td>
</tr>
<tr>
<td>pd_inmap24func_code</td>
<td>FUNC0x18</td>
</tr>
<tr>
<td>pd_inmap24size</td>
<td>SIZE0x18</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>pd_inmap24string</td>
<td>STRING0x18</td>
</tr>
<tr>
<td>pd_inmap25type</td>
<td>TYPE0x19</td>
</tr>
<tr>
<td>pd_inmap25func_code</td>
<td>FUNC0x19</td>
</tr>
<tr>
<td>pd_inmap25size</td>
<td>SIZE0x19</td>
</tr>
<tr>
<td>pd_inmap25string</td>
<td>STRING0x19</td>
</tr>
<tr>
<td>pd_inmap26type</td>
<td>TYPE0x1a</td>
</tr>
<tr>
<td>pd_inmap26func_code</td>
<td>FUNC0x1a</td>
</tr>
<tr>
<td>pd_inmap26size</td>
<td>SIZE0x1a</td>
</tr>
<tr>
<td>pd_inmap26string</td>
<td>STRING0x1a</td>
</tr>
<tr>
<td>pd_inmap27type</td>
<td>TYPE0x1b</td>
</tr>
<tr>
<td>pd_inmap27func_code</td>
<td>FUNC0x1b</td>
</tr>
<tr>
<td>pd_inmap27size</td>
<td>SIZE0x1b</td>
</tr>
<tr>
<td>pd_inmap27string</td>
<td>STRING0x1b</td>
</tr>
<tr>
<td>pd_inmap28type</td>
<td>TYPE0x1c</td>
</tr>
<tr>
<td>pd_inmap28func_code</td>
<td>FUNC0x1c</td>
</tr>
<tr>
<td>pd_inmap28size</td>
<td>SIZE0x1c</td>
</tr>
<tr>
<td>pd_inmap28string</td>
<td>STRING0x1c</td>
</tr>
<tr>
<td>pd_inmap29type</td>
<td>TYPE0x1d</td>
</tr>
<tr>
<td>pd_inmap29func_code</td>
<td>FUNC0x1d</td>
</tr>
<tr>
<td>pd_inmap29size</td>
<td>SIZE0x1d</td>
</tr>
<tr>
<td>pd_inmap29string</td>
<td>STRING0x1d</td>
</tr>
<tr>
<td>pd_inmap30type</td>
<td>TYPE0x1e</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td><code>pd_inmap30func_code</code></td>
<td><code>FUNC0x1e</code></td>
</tr>
<tr>
<td><code>pd_inmap30size</code></td>
<td><code>SIZE0x1e</code></td>
</tr>
<tr>
<td><code>pd_inmap30string</code></td>
<td><code>STRING0x1e</code></td>
</tr>
<tr>
<td><code>pd_inmap31type</code></td>
<td><code>TYPE0x1f</code></td>
</tr>
<tr>
<td><code>pd_inmap31func_code</code></td>
<td><code>FUNC0x1f</code></td>
</tr>
<tr>
<td><code>pd_inmap31size</code></td>
<td><code>SIZE0x1f</code></td>
</tr>
<tr>
<td><code>pd_inmap31string</code></td>
<td><code>STRING0x1f</code></td>
</tr>
<tr>
<td><code>pd_eorch</code></td>
<td><code>EORCH</code></td>
</tr>
<tr>
<td><code>pd_eofch</code></td>
<td><code>EOFCH</code></td>
</tr>
<tr>
<td><code>pd_tabch</code></td>
<td><code>TABCH</code></td>
</tr>
<tr>
<td><code>pd_bellch</code></td>
<td><code>BELLCH</code></td>
</tr>
<tr>
<td><code>pd_bspch</code></td>
<td><code>BSPCH</code></td>
</tr>
<tr>
<td><code>pd_case</code></td>
<td><code>UPC_LOCK</code></td>
</tr>
<tr>
<td><code>pd_backsp</code></td>
<td><code>BSB</code></td>
</tr>
<tr>
<td><code>pd_delete</code></td>
<td><code>LINEDEL</code></td>
</tr>
<tr>
<td><code>pd_echo</code></td>
<td><code>AUTOECHO</code></td>
</tr>
<tr>
<td><code>pd_alf</code></td>
<td><code>AUTOLF</code></td>
</tr>
<tr>
<td><code>pd_pause</code></td>
<td><code>PAGEPAUSE</code></td>
</tr>
<tr>
<td><code>pd_insm</code></td>
<td><code>INSERTMODE</code></td>
</tr>
<tr>
<td><code>pd_nulls</code></td>
<td><code>EOLNULLS</code></td>
</tr>
<tr>
<td><code>pd_page</code></td>
<td><code>PAGESIZE</code></td>
</tr>
<tr>
<td><code>pd_tabsiz</code></td>
<td><code>TABSIZE</code></td>
</tr>
</tbody>
</table>
EditMod Labels
5-SCF path options
1-‘\x7f’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
EDFUNCTION

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.

Table 3-22. ASCII Control Character Available Values

<table>
<thead>
<tr>
<th>Control Character is...</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>removed from the data stream.</td>
<td>IGNORE</td>
<td>0x0</td>
</tr>
<tr>
<td>passed on without editing.</td>
<td>PASSTHRU</td>
<td>0x1</td>
</tr>
<tr>
<td>removed from the data stream.</td>
<td>EDFUNCTION</td>
<td>0x2</td>
</tr>
</tbody>
</table>
EditMod Labels
5-SCF path options
2-‘\x7f’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
DELCHRU

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.

Table 3-23. ASCII Control Character Available Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move cursor to the left</td>
<td>MOVLEFT</td>
<td>0x00</td>
</tr>
<tr>
<td>Move cursor to the right</td>
<td>MOVRIGHT</td>
<td>0x01</td>
</tr>
<tr>
<td>Move cursor to the beginning of the</td>
<td>MOVBEGR</td>
<td>0x02</td>
</tr>
<tr>
<td>line</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move cursor to the end of the line</td>
<td>MOVEND</td>
<td>0x03</td>
</tr>
<tr>
<td>Reprint the current line to cursor</td>
<td>REPRINT</td>
<td>0x04</td>
</tr>
<tr>
<td>position</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Truncate the line at the cursor</td>
<td>TRUNCATE</td>
<td>0x05</td>
</tr>
<tr>
<td>position</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete character to the left</td>
<td>DELCHRL</td>
<td>0x06</td>
</tr>
</tbody>
</table>
### Table 3-23. ASCII Control Character Available Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete character under the cursor</td>
<td>DELCHRU</td>
<td>0x07</td>
</tr>
<tr>
<td>Delete word to the left</td>
<td>DELWRDL</td>
<td>0x08</td>
</tr>
<tr>
<td>Delete word to the right</td>
<td>DELWRDR</td>
<td>0x09</td>
</tr>
<tr>
<td>Delete the entire line</td>
<td>DELINE</td>
<td>0x0A</td>
</tr>
<tr>
<td>Undefined (reserved)</td>
<td>UNDEF1</td>
<td>0x0B</td>
</tr>
<tr>
<td>Input mode toggle (type over vs. insert)</td>
<td>MODETOGL</td>
<td>0x0C</td>
</tr>
<tr>
<td>Undefined (reserved)</td>
<td>UNDEF2</td>
<td>0x0D</td>
</tr>
<tr>
<td>End of record (read only)</td>
<td>ENDOREC</td>
<td>0x0E</td>
</tr>
<tr>
<td>End of file</td>
<td>ENDOFILE</td>
<td>0x0F</td>
</tr>
</tbody>
</table>
EditMod Labels
5-SCF path options
3-‘\x7f’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
**EditMod Labels**

5-SCF path options
4-‘\x7f’ string for key

**Description**
Character string to be echoed to the terminal.

**Port Generic Default Value**
NULL

**Port Specific Override Value**
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
pd_inmap1type
TYPE0x01

EditMod Labels
5-SCF path options
5-‘\x01’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
EDFUNCTION

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
6-‘\x01’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
MOVEND

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap1size
SIZE0x01

EditMod Labels
5-SCF path options
7-’\x01’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
8-‘\x01’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**pd_inmap2type**

**TYPE0x02**

**EditMod Labels**

5-SCF path options
9-‘\x02’ character mapping type

**Description**

Input mapping type for specified character.

**Port Generic Default Value**

Macro

`EDFUNCTION`

EditMod

`0x2`

**Port Specific Override Value**

Refer to `SCF/<DEVICE>/DESC/config.des` (**Figure 3-3**).

**Available Values**

The input mapping type codes are defined in the header file `scf.h`, and in **Table 3-22**.
**EditMod Labels**

5-SCF path options
10-‘\x02’ editing function code

**Description**

SCF editing function mapping code for specified character.

**Port Generic Default Value**

Macro

MOVLEFT

**EditMod**

0x07

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap2size
SIZE0x02

EditMod Labels
5-SCF path options
11-‘\x02’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels

5-SCF path options
12-‘\x02’ string for key

Description

Character string to be echoed to the terminal.

Port Generic Default Value

NULL

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
13-'\x03' character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
IGNORE

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
14-‘\x03’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels
5-SCF path options
15-‘\03’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this
field is specified as 0 (zero), an editing function built into SCF is executed to perform
the editing function. If this field is non-zero, the string pointed to by string 0x00 is
echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
16-'\x03' string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**pd_inmap4type**

**TYPE0x04**

**EditMod Labels**

5-SCF path options

17-‘\x04’ character mapping type

**Description**

Input mapping type for specified character.

**Port Generic Default Value**

Macro

\[ EDFUNCTION \]

**EditMod**

0x2

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
Chapter 3: SCF Device Descriptors

**pd_inmap4func_code**

**FUNCTION0x04**

**EditMod Labels**

5-SCF path options
18-‘\x04’ editing function code

**Description**

SCF editing function mapping code for specified character.

**Port Generic Default Value**

Macro

```
DELCHRU
```

**EditMod**

0x07

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap4size
SIZE 0x04

EditMod Labels
5-SCF path options
19-'\x04' size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
20-‘\x04’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap5type
TYPE 0x05

**EditMod Labels**

5-SCF path options
21-‘\x05’ character mapping type

**Description**

Input mapping type for specified character.

**Port Generic Default Value**

Macro

IGNORE

EditMod

0x2

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
pd_inmap5func_code
FUNC0x05

EditMod Labels
5-SCF path options
22-‘\x05’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap5size
SIZE0x05

EditMod Labels
5-SCF path options
23-‘\x05’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
**pd_inmap5string**
STRING0x05

**EditMod Labels**
5-SCF path options
24-‘\05’ string for key

**Description**
Character string to be echoed to the terminal.

**Port Generic Default Value**
NULL

**Port Specific Override Value**
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap6type
TYPE0x06

EditMod Labels
5-SCF path options
25-‘\x06’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
EDFUNCTION

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
26-‘\06’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
MOVRIGHT

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels

5-SCF path options
27-‘\x06’ size of associated string

Description

This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

0 to 4294967295
EditMod Labels
5-SCF path options
28-‘\x06’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
**pd_inmap7type**

**TYPE0x07**

**EditMod Labels**

5-SCF path options
29-‘\x07’ character mapping type

**Description**

Input mapping type for specified character.

**Port Generic Default Value**

Macro

```
PASSTHRU
```

EditMod

0x2

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
**EditMod Labels**

5-SCF path options
30-‘\x07’ editing function code

**Description**

SCF editing function mapping code for specified character.

**Port Generic Default Value**

Macro

0

EditMod

0x07

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap7size
SIZE0x07

EditMod Labels
5-SCF path options
31-‘\x07’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
Chapter 3: SCF Device Descriptors

pd_inmap7string
STRING0x07

EditMod Labels
5-SCF path options
32-‘\07’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
33-‘\08’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
    EDFUNCTION

EditMod
0x2

Port Specific Override Value
Refer to scf/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5 - SCF path options
34 - ‘\08’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
   DELCHRL

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
**pd_inmap8size**

**SIZE 0x08**

**EditMod Labels**

5-SCF path options
35-‘\x08’ size of associated string

**Description**

This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

0 to 4294967295
pd_inmap8string
STRING0x08

EditMod Labels
5-SCF path options
36-‘\08’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
37-‘\09’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
EDFUNCTION

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
38-‘\x09’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
MODETOGL

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap9size  
SIZE0x09

EditMod Labels

5-SCF path options
39-‘\x09’ size of associated string

Description

This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

0 to 4294967295
EditMod Labels

5-SCF path options
40-‘\09’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
41-‘\0a’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
  PASSTHRU

EditMod
  0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
42-‘\0a’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
**pd_inmap10size**

**SIZE0x0a**

**EditMod Labels**

5-SCF path options

43-‘\x0a’ size of associated string

**Description**

This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

0 to 4294967295
EditMod Labels
5-SCF path options
44-‘\0a’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
**pd_inmap11type**

**TYPE0x0b**

**EditMod Labels**

5-SCF path options
45-'\0b' character mapping type

**Description**

Input mapping type for specified character.

**Port Generic Default Value**

Macro

```
EDFUNCTION
```

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The input mapping type codes are defined in the header file `scf.h`, and in Table 3-22.
EditMod Labels

5-SCF path options
46-‘\0b’ editing function code

Description

SCF editing function mapping code for specified character.

Port Generic Default Value

Macro

TRUNCATE

EditMod

0x07

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
**pd_inmap11size**

SIZE 0x0b

**EditMod Labels**

5-SCF path options
47-`\0b` size of associated string

**Description**

This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

0 to 4294967295
EditMod Labels
5-SCF path options
48-‘\x0b’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap12type

TYPE 0x0c

EditMod Labels

5-SCF path options
49-‘\x0c’ character mapping type

Description

Input mapping type for specified character.

Port Generic Default Value

Macro

EDFUNCTION

EditMod

0x2

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
**EditMod Labels**

5 - SCF path options
50 - ‘\0c’ editing function code

**Description**

SCF editing function mapping code for specified character.

**Port Generic Default Value**

Macro

```
DELWRDL
```

**EditMod**

0x07

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels
5-SCF path options
51-‘\0c’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
52-‘\0c’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
53-‘\0d’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
    EDFUNCTION

EditMod
    0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
Chapter 3: SCF Device Descriptors

**pd_inmap13func_code**

**FUNC**: \(0x0d\)

### EditMod Labels
- **5**: SCF path options
- **54**: ‘\(0d\)’ editing function code

### Description
SCF editing function mapping code for specified character.

### Port Generic Default Value
Macro
```
ENDOREC
```

### Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

### Available Values
The SCF editing function mapping type codes are defined in the header file `scf.h`, and in Table 3-23.
**pd_inmap13size**

**SIZE0x0d**

**EditMod Labels**
- 5-SCF path options
- 55-`\0d` size of associated string

**Description**
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

**Port Generic Default Value**
0 (zero)

**Port Specific Override Value**
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**
0 to 4294967295
EditMod Labels
5-SCF path options
56-‘\0d’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap14type
TYPE0x0e

EditMod Labels
5-SCF path options
57-‘\0e’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
PASSTHRU

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
58-‘\x0e‘ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap14size
SIZE0x0e

EditMod Labels
5-SCF path options
59-‘\xe’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels

5-SCF path options
60-‘\0e’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value

NULL

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
61-‘\0f’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
PASSTHRU

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
62-'\0f' editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap15size
SIZE0x0f

EditMod Labels
5-SCF path options
63-‘\x0f’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
64-‘\0f’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap16type
TYPE0x10

**EditMod Labels**
5-SCF path options
65-‘\10’ character mapping type

**Description**
Input mapping type for specified character.

**Port Generic Default Value**
Macro

```
EDFUNCTION
```

EditMod

0x2

**Port Specific Override Value**
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
Chapter 3: SCF Device Descriptors

pd_inmap16func_code
FUNC0x10

EditMod Labels
5-SCF path options
6-‘\10’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
REPRINT

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
**pd_inmap16size**
SIZE0x10

**EditMod Labels**
5-SCF path options
67-‘\x10’ size of associated string

**Description**
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

**Port Generic Default Value**
0 (zero)

**Port Specific Override Value**
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**
0 to 4294967295
pd_inmap16string
STRING0x10

EditMod Labels
5-SCF path options
68-‘\x10’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap17type
TYPE0x11

EditMod Labels
5-SCF path options
69-‘\11’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
   IGNORE

EditMod
   0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels

5-SCF path options
70-‘\x11’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap17size
SIZE0x11

EditMod Labels
5-SCF path options
71-‘\x11’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
72-‘\x11’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap18type
TYPE0x12

EditMod Labels
5-SCF path options
73-‘\x12’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
EDFUNCTION

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
pd_inmap18func_code
FUNC0x12

EditMod Labels
5-SCF path options
74-‘\x12’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
   DELWRDR

EditMod
   0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels

5-SCF path options
75-‘\x12’ size of associated string

Description

This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

0 to 4294967295
EditMod Labels
5-SCF path options
76-‘\x12’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
77-‘\13’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
  IGNORE

EditMod
  0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5 - SCF path options
78 - ‘\x13’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels
5-SCF path options
79-‘\x13’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
pd_inmap19string
STRING0x13

EditMod Labels
5-SCF path options
80-‘\x13’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap20type
TYPE0x14

EditMod Labels
5-SCF path options
81-‘\x14’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
PASSTHRU

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
82-‘\x14’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels
5-SCF path options
83-‘\x14’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels

5-SCF path options
84-’\x14’ string for key

Description

Character string to be echoed to the terminal.

Port Generic Default Value

NULL

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap21type
TYPE0x15

EditMod Labels
5-SCF path options
85-‘\x15’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
PASSTHRU

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
**pd_inmap21func_code**

**FUNC0x15**

### EditMod Labels

5-SCF path options
86-‘\x15’ editing function code

### Description

SCF editing function mapping code for specified character.

### Port Generic Default Value

Macro
0

EditMod
0x07

### Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

### Available Values

The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap21size
SIZE0x15

EditMod Labels
5-SCF path options
87-‘\x15’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
88-‘\x15’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap22type
TYPE 0x16

EditMod Labels
5-SCF path options
89-‘\x16’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
PASSTHRU

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
90-‘\x16’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels
5-SCF path options
91-‘\x16’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels

5-SCF path options
92-’\x16’ string for key

Description

Character string to be echoed to the terminal.

Port Generic Default Value

NULL

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap23type
TYPE0x17

EditMod Labels
5-SCF path options
93-‘\x17’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
IGNORE

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
94-‘\x17’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap23size
SIZE0x17

EditMod Labels
5-SCF path options
95-‘\x17’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
96-‘\x17’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
**EditMod Labels**
5-SCF path options
97-‘\x18’ character mapping type

**Description**
Input mapping type for specified character.

**Port Generic Default Value**
Macro
```
EDFUNCTION
```

**EditMod**
0x2

**Port Specific Override Value**
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
98-‘\18’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
    DELINE

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels

5-SCF path options
99-'\x18' size of associated string

Description

This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

0 to 4294967295
EditMod Labels

5-SCF path options
100-'\x18' string for key

Description

Character string to be echoed to the terminal.

Port Generic Default Value

NULL

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
**pd_inmap25type**

TYPE 0x19

---

**EditMod Labels**

5-SCF path options

101-’\x19’ character mapping type

**Description**

Input mapping type for specified character.

**Port Generic Default Value**

Macro

```
PASSTHRU
```

**EditMod**

0x2

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
- 5 - SCF path options
- 102 - \x19 editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
DELCHRU

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap25size
SIZE0x19

EditMod Labels
5-SCF path options
103-’\x19’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
104-’\x19’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
105-’\x1a’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
EDFUNCTION

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5 - SCF path options
106 - ‘\x1a’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
MOVEBEG

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels

5-SCF path options
107-’\x1a’ size of associated string

Description

This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

0 to 4294967295
**EditMod Labels**

5-SCF path options
108-‘\x1a’ string for key

**Description**

Character string to be echoed to the terminal.

**Port Generic Default Value**

NULL

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
**pd_inmap27type**

TYPE 0x1b

**EditMod Labels**

5-SCF path options
109-’\x1b‘ character mapping type

**Description**

Input mapping type for specified character.

**Port Generic Default Value**

Macro

EDFUNCTION

EditMod

0x2

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
pd_inmap27func_code
FUNC0x1b

EditMod Labels
5-SCF path options
110-‘\x1b’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
ENDOFILE

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
**pd_inmap27size**
SIZE0x1b

**EditMod Labels**

5-SCF path options
111-’\x1b’ size of associated string

**Description**

This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

0 to 4294967295
**pd_inmap27string**

**STRING0x1b**

**EditMod Labels**

5-SCF path options
112-'\x1b' string for key

**Description**

Character string to be echoed to the terminal.

**Port Generic Default Value**

NULL

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
pd_inmap28type
  TYPE0x1c

EditMod Labels
5-SCF path options
113-’\x1c’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
  PASSTHRU

EditMod
  0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5-SCF path options
114-'\x1c' editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap28size
SIZE0x1c

EditMod Labels
5-SCF path options
115-’\x1c’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
116-’\x1c’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
**EditMod Labels**

5-SCF path options

117-‘\x1d’ character mapping type

**Description**

Input mapping type for specified character.

**Port Generic Default Value**

Macro

PASSTHRU

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
EditMod Labels
5 - SCF path options
118 - ‘\x1d’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
EditMod Labels
5-SCF path options
119-’\x1d’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
120-’\x1d’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
121-’\xe’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
PASSTHRU

EditMod
0x2

Port Specific Override Value
Refer to \texttt{SCF/<DEVICE>/DESC/config.des} (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file \texttt{scf.h}, and in Table 3-22.
**pd_inmap30func_code**

**FUNC0x1e**

### EditMod Labels

5-SCF path options
122-‘\x1e’ editing function code

### Description

SCF editing function mapping code for specified character.

### Port Generic Default Value

Macro

0

EditMod

0x07

### Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

### Available Values

The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
**pd_inmap30size**  
SIZE 0x1e

**EditMod Labels**
- 5-SCF path options
- 123-'\x1e' size of associated string

**Description**
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

**Port Generic Default Value**
0 (zero)

**Port Specific Override Value**
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**
0 to 4294967295
pd_inmap30string
STRING0x1e

EditMod Labels
5-SCF path options
124-'\x1e' string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
125-‘\x1f’ character mapping type

Description
Input mapping type for specified character.

Port Generic Default Value
Macro
PASSTHRU

EditMod
0x2

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The input mapping type codes are defined in the header file scf.h, and in Table 3-22.
Chapter 3: SCF Device Descriptors

pd_inmap31func_code
func0x1f

EditMod Labels
5-SCF path options
126-’\x1f’ editing function code

Description
SCF editing function mapping code for specified character.

Port Generic Default Value
Macro
0

EditMod
0x07

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF editing function mapping type codes are defined in the header file scf.h, and in Table 3-23.
pd_inmap31size
SIZE0x1f

EditMod Labels
5-SCF path options
127-‘\x1f’ size of associated string

Description
This field specifies the size of the editing function string to echo to the terminal. If this field is specified as 0 (zero), an editing function built into SCF is executed to perform the editing function. If this field is non-zero, the string pointed to by string 0x00 is echoed to the terminal.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
0 to 4294967295
EditMod Labels
5-SCF path options
128-’\1f’ string for key

Description
Character string to be echoed to the terminal.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
5-SCF path options
129-end of record character (read only)

Description
This specifies the end of record character.

Port Generic Default Value
Macro
EORCH (defined as C_CR in scfdesc.h)

EditMod
'\n'

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The ASCII control and special characters are defined in the header file, scf.h, and in Table 3-24.

<table>
<thead>
<tr>
<th>SCF/OS-9 Compatible Standard Codes</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_NULL</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td>C_REPEAT</td>
<td>CTRL_A</td>
<td>0x01</td>
</tr>
<tr>
<td></td>
<td>CTRL_B</td>
<td>0x02</td>
</tr>
<tr>
<td>C_INTR</td>
<td>CTRL_C</td>
<td>0x03</td>
</tr>
<tr>
<td>C_REPRINT</td>
<td>CTRL_D</td>
<td>0x04</td>
</tr>
<tr>
<td>C_QUIT</td>
<td>CTRL_E</td>
<td>0x05</td>
</tr>
<tr>
<td></td>
<td>CTRL_F</td>
<td>0x06</td>
</tr>
<tr>
<td>C_BELL</td>
<td>CTRL_G</td>
<td>0x07</td>
</tr>
<tr>
<td>SCF/OS-9 Compatible Standard Codes</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>C_BACKSPACE</td>
<td>CTRL_H</td>
<td>0x08</td>
</tr>
<tr>
<td>C_TAB</td>
<td>CTRL_I</td>
<td>0x09</td>
</tr>
<tr>
<td>C_LINEFEED</td>
<td>CTRL_J</td>
<td>0x0A</td>
</tr>
<tr>
<td></td>
<td>CTRL_K</td>
<td>0x0B</td>
</tr>
<tr>
<td>C_FORMFEED</td>
<td>CTRL_L</td>
<td>0x0C</td>
</tr>
<tr>
<td>C_CR</td>
<td>CTRL_M</td>
<td>0x0D</td>
</tr>
<tr>
<td></td>
<td>CTRL_N</td>
<td>0x0E</td>
</tr>
<tr>
<td></td>
<td>CTRL_O</td>
<td>0x0F</td>
</tr>
<tr>
<td></td>
<td>CTRL_P</td>
<td>0x10</td>
</tr>
<tr>
<td>C_XON</td>
<td>CTRL_Q</td>
<td>0x11</td>
</tr>
<tr>
<td></td>
<td>CTRL_R</td>
<td>0x12</td>
</tr>
<tr>
<td>C_XOFF</td>
<td>CTRL_S</td>
<td>0x13</td>
</tr>
<tr>
<td></td>
<td>CTRL_T</td>
<td>0x14</td>
</tr>
<tr>
<td></td>
<td>CTRL_U</td>
<td>0x15</td>
</tr>
<tr>
<td></td>
<td>CTRL_V</td>
<td>0x16</td>
</tr>
<tr>
<td>C_PAUSE</td>
<td>CTRL_W</td>
<td>0x17</td>
</tr>
<tr>
<td>C_DELLINE</td>
<td>CTRL_X</td>
<td>0x18</td>
</tr>
<tr>
<td></td>
<td>CTRL_Y</td>
<td>0x19</td>
</tr>
<tr>
<td></td>
<td>CTRL_Z</td>
<td>0x1A</td>
</tr>
<tr>
<td></td>
<td>CTRL_SPACE</td>
<td>0x20</td>
</tr>
<tr>
<td></td>
<td>CTRL_COMMA</td>
<td>0x2C</td>
</tr>
<tr>
<td></td>
<td>CTRL_PERIOD</td>
<td>0x2E</td>
</tr>
<tr>
<td>SCF/OS-9 Compatible Standard Codes</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>CTRL_SLASH</td>
<td>0x2F</td>
</tr>
<tr>
<td></td>
<td>C_EOF</td>
<td>0x1B</td>
</tr>
</tbody>
</table>
EditMod Labels

5-SCF path options
130-end of file character

Description
This specifies the end of file character.

Port Generic Default Value
Macro
EOFCH (defined as C_EOF in scfdesc.h)

EditMod
0x1B

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The ASCII control and special characters are defined in the header file, scf.h, and in Table 3-24.
EditMod Labels
5-SCF path options
131-tab character (0 = none)

Description
This defines the tab character.

Port Generic Default Value
Macro
    TABCH (defined as C_TAB in scfdesc.h)

EditMod
0x09

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The ASCII control and special characters are defined in the header file, scf.h, and in Table 3-24.
EditMod Labels

5-SCF path options
132-bell (line overflow)

Description

This defines the bell character.

Port Generic Default Value

Macro

BELLCH (defined as C_BELL in scfdesc.h)

EditMod

0x07

Port Specific Override Value

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values

The ASCII control and special characters are defined in the header file, scf.h, and in Table 3-24.
EditMod Labels
5-SCF path options
133-backspace echo character

Description
This defines the backspace echo character.

Port Generic Default Value
Macro
BSPCH (defined as C_BACKSPACE in scfdesc.h)

EditMod
'b'

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The ASCII control and special characters are defined in the header file, scf.h, and in Table 3-24.
EditMod Labels
5-SCF path options
134-case lock

Description
This specifies the state of the upper case lock character.

Port Generic Default Value
The default is upper and lower case.
Macro
\[ \text{UPC\_LOCK} \] (defined as \text{PLOFF} in \text{scfdesc.h})

EditMod
0x0

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF character logic states are defined in the header file, \text{scf.h}, and in Table 3-25.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive logic off - Upper and lower case</td>
<td>PLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Positive logic on</td>
<td>PLON</td>
<td>0x01</td>
</tr>
<tr>
<td>Negative logic off</td>
<td>NLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Negative logic on</td>
<td>NLON</td>
<td>0x01</td>
</tr>
</tbody>
</table>
**pd_backsp**

**BSB**

**EditMod Labels**

- 5-SCF path options
- 135-backspace

**Description**

This specifies the state of the backspace character.

**Port Generic Default Value**

The default is destructive backspace.

**Macro**

BSB (defined as PLON in scfdesc.h)

**EditMod**

0x01

**Port Specific Override Value**

Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

**Available Values**

The SCF character logic states are defined in the header file, scf.h, and in Table 3-26.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive logic off</td>
<td>PLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Positive logic on - Destructive backspace</td>
<td>PLON</td>
<td>0x01</td>
</tr>
<tr>
<td>Negative logic off</td>
<td>NLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Negative logic on</td>
<td>NLON</td>
<td>0x01</td>
</tr>
</tbody>
</table>
EditMod Labels
5-SCF path options
136-delete line

Description
This specifies the state of the delete line character.

Port Generic Default Value
The default is destructive line delete.
Macro
INLINEDEL (defined as PLON in scfdesc.h)

EditMod
0x01

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF character logic states are defined in the header file, scf.h, and in Table 3-27.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive logic off - Nondestructive line delete</td>
<td>PLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Positive logic on - Destructive line delete</td>
<td>PLON</td>
<td>0x01</td>
</tr>
<tr>
<td>Negative logic off</td>
<td>NLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Negative logic on</td>
<td>NLON</td>
<td>0x01</td>
</tr>
</tbody>
</table>
**EditMod Labels**

5-SCF path options
137-echo

**Description**

This specifies whether the character echo is on or off.

**Port Generic Default Value**

The default is echo on.

Macro

`AUTOECHO` *(defined as `PLON` in `scfdesc.h`)*

**EditMod**

0x01

**Port Specific Override Value**

Refer to `SCF/<DEVICE>/DESC/config.des` *(Figure 3-3)*.

**Available Values**

The SCF character logic states are defined in the header file, `scf.h`, and in Table 3-28.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive logic off - Echo off</td>
<td>PLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Positive logic on - Echo on</td>
<td>PLON</td>
<td>0x01</td>
</tr>
<tr>
<td>Negative logic off</td>
<td>NLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Negative logic on</td>
<td>NLON</td>
<td>0x01</td>
</tr>
</tbody>
</table>
Chapter 3: SCF Device Descriptors

EditMod Labels
5-SCF path options
138-auto-linefeed

Description
This specifies whether the auto line feed is on or off.

Port Generic Default Value
The default is auto linefeed on.
Macro
AUTOLF (defined as PLON in scfdesc.h)

EditMod
0x01

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF character logic states are defined in the header file, scf.h, and in Table 3-29.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive logic off - Auto linefeed off</td>
<td>PLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Positive logic on - Auto linefeed on</td>
<td>PLON</td>
<td>0x01</td>
</tr>
<tr>
<td>Negative logic off</td>
<td>NLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Negative logic on</td>
<td>NLON</td>
<td>0x01</td>
</tr>
</tbody>
</table>
EditMod Labels
5-SCF path options
139-end-of-page pause

Description
This specifies whether the page pause is on or off.

Port Generic Default Value
The default is page pause on.
Macro
PAGEPAUSE (defined as PLON in scfdesc.h)

EditMod
0x01

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF character logic states are defined in the header file, scf.h, and in Table 3-30.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive logic off - Auto linefeed off</td>
<td>POFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Positive logic on - Auto linefeed on</td>
<td>PLON</td>
<td>0x01</td>
</tr>
<tr>
<td>Negative logic off</td>
<td>NLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Negative logic on</td>
<td>NLON</td>
<td>0x01</td>
</tr>
</tbody>
</table>
EditMod Labels
5-SCF path options
140-insert mode

Description
This specifies whether the insert mode is on or off.

Port Generic Default Value
The default is insert mode off.
Macro
\[
\text{INSERTMODE} \quad \text{(defined as PLOFF in scfdesc.h)}
\]

EditMod
0x00

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
The SCF character logic states are defined in the header file, scf.h, and in Table 3-31.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive logic off - Insert mode off</td>
<td>PLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Positive logic on - Insert mode on</td>
<td>PLON</td>
<td>0x01</td>
</tr>
<tr>
<td>Negative logic off</td>
<td>NLOFF</td>
<td>0x00</td>
</tr>
<tr>
<td>Negative logic on</td>
<td>NLON</td>
<td>0x01</td>
</tr>
</tbody>
</table>
EditMod Labels
5-SCF path options
141-end of line null count

Description
This specifies the number of end of line nulls.

Port Generic Default Value
0 (no end of line nulls)

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
-128 to 127
EditMod Labels
5-SCF path options
142-lines per page

Description
This specifies the number of lines per page.

Port Generic Default Value
24

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
-128 to 127
EditMod Labels
5-SCF path options
143-tab field size

Description
This specifies the number of spaces a tab skips.

Port Generic Default Value
4

Port Specific Override Value
Refer to SCF/<DEVICE>/DESC/config.des (Figure 3-3).

Available Values
-128 to 127
SBF device descriptors contain configuration data specific to one OS-9 format disk device on an OS-9 system. Values which can be configured in the descriptor include:

- Device interrupt vector and priority
- Device I/O address
- Device geometry
- Logical sector size

The next section in this chapter provides a detailed example of the configuration options you can use to change configuration values for SBF (sequential block file).

The rest of this chapter provides a detailed list of all of the SBF device descriptor fields.

This chapter includes the following topics:

- SBF Field Configuration Options
- SBF Device Descriptor Field Reference
- Module Header Fields
- Device Descriptor Data Definition Fields
- SBF Path Options Fields
- SBF Logical Unit Status Fields
SBF Field Configuration Options

To change an SBF device descriptor module configuration field, you can use either of the following methods:

1. Use the EditMod utility to directly modify existing SBF device descriptor modules either as a stand-alone module or as part of a merged module group (such as a boot image).
2. Modify the description file for the SBF device descriptor module and rebuild it using the makefile provided.

Direct Modification Advantages

The direct modification method has the following advantages:

- **Fast** No source configuration file rebuilds are necessary.
- **Temporary** The original module or merged-module group configuration can be easily restored through the appropriate rebuild.
- **Contained** Changes are limited to the individual boot image modified (merged-module option).

Description File/Rebuild Advantages

The advantage of the description file/rebuild method is that the changes are permanent and reproducible. Modifications apply to all subsequent module rebuilds and to all merged-module groups built containing the updated module.

Both methods are documented in this section. These procedures are used with the field descriptions starting with the Module Header Fields. For direct modification, use the EditMod LABELS data to navigate the EditMod menus. The DESCRIPTION FILE MACRO data identifies the macro you need to define/modify in the configuration sources to rebuild the SBF device descriptor module.

Direct Modification

Use the EditMod utility and the following procedures to directly modify fields in the existing SBF device descriptor module. The module can stand-alone or it can be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The field references later in this chapter contain a description of each configurable field, its supported values, and the sequence of menu options required by EditMod to modify that field.

Refer to the Utilities Reference for a full description of EditMod’s capabilities.
Refer to your board guide for information about how to modify the module lists and remake the boot images, and for specific boot image names.

**Direct Modification Procedures**

To modify the stand-alone module, complete the following steps:

1. Change to the `CMDS/BOOTOBJ/DESC/<DEVICE>` directory (see Figure 4-1).
2. Use `EditMod` to edit the module:
   ```
   $EditMod -e <descriptor>
   ```

To modify the module as part of a merged module group, complete the following steps:

1. Change to the `BOOTS/SYSTEMS/PORTBOOT` directory (see Figure 4-2).
2. Use `EditMod` to edit the module:
   ```
   $EditMod -e <descriptor> -f=<boot image name>
   ```
3. Use the menu selections provided in the EditMod LABELS section of the field reference later in this chapter to locate the fields you want to edit.

4. Select a new value for the field from the AVAILABLE VALUES section of the field reference. Enter that value at the EditMod prompt to modify the field.

5. If you want to make additional modifications, use the p command (previous) to step backward through the EditMod menus. Repeat Steps 3 and 4 until you have made all desired modifications to the descriptor.

6. Select the w command (write) to save the changes.

7. Select the q command (quit) to exit EditMod.

Unless you modified the SBF device descriptors in your boot image, you should rebuild your boot image to include the new descriptor.

**Example EditMod Session**

This example modifies an SBF device descriptor as part of the boot image rom:

```
$ EditMod -e mt0 -f=rom
```

1. module header
2. device descriptor data definitions
3. SBF path options structure
4. SBF logical unit status

Which? [?/1-4/p/t/a/w/q] 4

1. irq vector : 0x4b
2. irq level : 0x4
3. irq priority : 0xa
4. drive flag : 0x0

Which? [?/1-4/p/t/a/w/q] 3

irq priority : 0xa

New value: 1

1. irq vector : 0x4b
2. irq level : 0x4
3. irq priority : 0xa
4. drive flag : 0x0

Which? [?/1-19/p/t/a/w/q] w

Which? [?/1-19/p/t/a/w/q] q
Description File Configuration

You can use these procedures to modify the appropriate description file and rebuild the SBF device descriptors for your port directory. The DESCRIPTION FILE MACROS section of the field reference specifies the name of the macro you modify/define in the description files to configure the field. The value used in the define is chosen from the AVAILABLE VALUES specified for the field.

![Figure 4-3. Directory Location for Modifying SBF Description Files](image)

Description File Configuration Procedures

1. Change to the SBF/<DEVICE> directory (see Figure 4-3).
2. Edit the file config.des and read the included comments for more information on how to use the specific description files provided in your software distribution. The config.des file contains a list of macro names that can be defined to override the global default values for the configuration fields.
3. Refer to the DESCRIPTION FILE MACRO section in the field reference later in this chapter to determine the macro name you define to configure the target field.
4. Read the comments in config.des to determine where to place the define for this macro.
5. Select the value you want to use to configure the field. See the AVAILABLE VALUES section of the field reference data for values or macros that can be used for the definition. Define the macro by entering a definition in the appropriate description files as follows:

```c
#define <macro> <value>
```
6. Save the changes and rebuild the SBF device descriptors, entering the following command in the SBF/<DEVICE>/DESC directory:

```bash
os9make
```
7. Rebuild your boot image to include the new descriptor.

SBF Device Descriptor Field Reference

This section contains a list of the most commonly configured fields in the SBF device descriptors. Each field entry contains the following information:

- **<Field name>** - The call name for each field that can be reconfigured in the module.
• **EditMod LABELS**: EditMod menu selections for navigating to the proper field in and EditMod session.

• **DESCRIPTION FILE MACRO**: The macro name you modify/define in the description file.

• **DESCRIPTION**: A brief description of the field’s purpose and use.

• **EXAMPLE**: An optional example of the description file entry showing how to change the value of this field.

• **PORT GENERIC DEFAULT VALUE**: The value set in the port generic description file for this field. This is the value the field is assigned when the module is built, unless the appropriate macro has been defined in the port specific description file to override this default value.

• **PORT SPECIFIC OVERRIDE VALUE**: The value set in the port specific description file for this field. If defined, this is the value the field is assigned when the module is built, overriding the port generic default value.

• **AVAILABLE VALUES**: Values to which the field can be set through EditMod or the description files. In many cases, this data is presented in a table that maps a description of the value to a numeric value appropriate for entry in EditMod, and to a pre-defined macro available for use in the description file.

### Module Header Fields

The following section contains the module header fields in the order they appear during an interactive EditMod session. Defined fields may appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>_m_group</td>
<td>MH_GROUP</td>
</tr>
<tr>
<td>_m_user</td>
<td>MH_USER</td>
</tr>
<tr>
<td>mod_name</td>
<td>MH_NAME</td>
</tr>
<tr>
<td>m_access</td>
<td>MH_ACCESS</td>
</tr>
<tr>
<td>m_tylan</td>
<td>MH_TYLAN</td>
</tr>
<tr>
<td>m_attrev</td>
<td>MH_ATTREV</td>
</tr>
<tr>
<td>m_edit</td>
<td>MH_EDITION</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
1-module owner’s group number

Description

Group ID of the module’s owner. The group number allows people working in the same department or on the same project to share a common identification number.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values

0 to 65535
EditMod Labels
1-module header
2-module owner’s user number

Description
User ID of the module’s owner. The user number identifies a specific user.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
0 to 65535
EditMod Labels
1-module header
3-module name

Description
Contains the module name string.

Port Generic Default Value
String value (None)

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
EditMod Labels

4-access permissions

Description
Defines the permissible module access by its owner or by other users.

Port Generic Default Value
Macro

\[
\text{mp}_\text{access} = \text{MP}_{\text{OWNER}} \text{READ} \mid \text{MP}_{\text{OWNER}} \text{EXEC} \mid \text{MP}_{\text{GROUP}} \text{READ} \mid \text{MP}_{\text{GROUP}} \text{EXEC} \mid \text{MP}_{\text{WORLD}} \text{READ} \mid \text{MP}_{\text{WORLD}} \text{EXEC}
\]

EditMod
0x555

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
Module access permission values are located in the header file, module.h, and are listed in Table 4-2.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read permission by owner</td>
<td>MPOWNERREAD</td>
<td>0x0001</td>
</tr>
<tr>
<td>Write permission by owner</td>
<td>MPUOWNERWRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Execute permission by owner</td>
<td>MPUOWNEREXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Owner permission mask</td>
<td>MPUOWNERMASK</td>
<td>0x000f</td>
</tr>
<tr>
<td>Read permission by group</td>
<td>MPGROUPEXEC</td>
<td>0x0010</td>
</tr>
<tr>
<td>Write permission by group</td>
<td>MPGRPOUPEXEC</td>
<td>0x0020</td>
</tr>
<tr>
<td>Execute permission by group</td>
<td>MPGRPOUPEXEC</td>
<td>0x0040</td>
</tr>
</tbody>
</table>

Table 4-2. m_access Available Values
### Table 4-2. Available Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group permission mask</td>
<td>MP_GROUP_MASK</td>
<td>0x00f0</td>
</tr>
<tr>
<td>Read permission by world</td>
<td>MP_WORLD_READ</td>
<td>0x0100</td>
</tr>
<tr>
<td>Write permission by world</td>
<td>MP_WORLD_WRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Execute permission by world</td>
<td>MP_WORLD_EXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>World permission mask</td>
<td>MP_WORLD_MASK</td>
<td>0x0f00</td>
</tr>
<tr>
<td>All permissions for owner, group, and world</td>
<td>MP_WORLD_ACCESS</td>
<td>0x0777</td>
</tr>
<tr>
<td>System permission mask</td>
<td>MP_SYSTM_MASK</td>
<td>0xf000</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
5-type/language

Description

Contains the module’s type (first byte) and language (second byte). The language codes indicate if the module is executable and which language the run-time system requires for execution, if any.

Port Generic Default Value

Macro

\[(MT\_DATA<<8) + ML\_OBJECT\]

EditMod

0x401

Port Specific Override Value

Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values

Module type values and language codes are located in the header file, module.h, and are listed in Table 4-3 and Table 4-4.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not used (wildcard value in system calls)</td>
<td>MT_ANY</td>
<td>0x0000</td>
</tr>
<tr>
<td>Program module</td>
<td>MT_PROGRAM</td>
<td>0x0001</td>
</tr>
<tr>
<td>Subroutine module</td>
<td>MT_SUBROUT</td>
<td>0x0002</td>
</tr>
<tr>
<td>Multi-module (reserved for future use)</td>
<td>MT_MULTI</td>
<td>0x0003</td>
</tr>
<tr>
<td>Data module</td>
<td>MT_DATA</td>
<td>0x0004</td>
</tr>
<tr>
<td>Configuration data block data module</td>
<td>MT_CDBDATA</td>
<td>0x0005</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0xb-0xa</td>
<td>0xb-0xa</td>
</tr>
</tbody>
</table>
### Table 4-3. `m_tylan` Available Module Type Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>User trap library</td>
<td>MT_TRAPLIB</td>
<td>0x000b</td>
</tr>
<tr>
<td>System module</td>
<td>MT_SYSTEM</td>
<td>0x000c</td>
</tr>
<tr>
<td>File manager module</td>
<td>MT_FILEMAN</td>
<td>0x000d</td>
</tr>
<tr>
<td>Physical device driver</td>
<td>MT_DEVDRVR</td>
<td>0x000e</td>
</tr>
<tr>
<td>Device descriptor module</td>
<td>MT_DEVDESC</td>
<td>0x000f</td>
</tr>
<tr>
<td>User definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module type mask</td>
<td>MT_MASK</td>
<td>0xff00</td>
</tr>
</tbody>
</table>

### Table 4-4. `m_tylan` Available Language Code Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unspecified language (wildcard in system calls)</td>
<td>ML_ANY</td>
<td>0x0</td>
</tr>
<tr>
<td>Machine language</td>
<td>ML_OBJECT</td>
<td>0x1</td>
</tr>
<tr>
<td>Basic I-code (reserved for future use)</td>
<td>ML_ICODE</td>
<td>0x2</td>
</tr>
<tr>
<td>Pascal P-code (reserved for future use)</td>
<td>MLPCODE</td>
<td>0x3</td>
</tr>
<tr>
<td>C I-code (reserved for future use)</td>
<td>ML_CCODE</td>
<td>0x4</td>
</tr>
<tr>
<td>Cobol I-code (reserved for future use)</td>
<td>ML_CBLCODE</td>
<td>0x5</td>
</tr>
<tr>
<td>Fortran</td>
<td>ML_FRTNCODE</td>
<td>0x6</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0x7-0xf</td>
<td>0x7-0xf</td>
</tr>
<tr>
<td>User-definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module language mask</td>
<td>ML_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
6-revision/attributes

Description
Contains the module’s attributes (first byte) and revision (second byte).

Port Generic Default Value
Macro
MA_REENT<<8

EditMod
0x8000

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
Module attribute and revision codes are located in the header file module.h., and are listed in Table 4-5.

If two modules with the same name are found in the memory search or are loaded into the current module directory, only the module with the highest revision level is kept. This enables easy substitution of modules for update or correction.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is re-entrant (shareable by multiple tasks).</td>
<td>MA_REENT</td>
<td>0x80 (shifted left to first byte: 0x8000)</td>
</tr>
<tr>
<td></td>
<td>(shifted left to first byte: MA_REENT&lt;&lt;8)</td>
<td></td>
</tr>
<tr>
<td>The module is sticky. A sticky module is not removed from memory until its link count becomes -1 or memory is required for another use.</td>
<td>MA_GHOST</td>
<td>0x40 (shifted left to first byte: 0x4000)</td>
</tr>
<tr>
<td></td>
<td>(shifted left to first byte: MA_GHOST&lt;&lt;8)</td>
<td></td>
</tr>
</tbody>
</table>
The module is a system-state module. MA_SUPER (shifted left to first byte: MA_SUPER<<8) 0x20 (shifted left to first byte: 0x2000)

User-definable revision number 0x0-0xfe 0x0-0xfe

Module attribute mask MA_MASK 0xff00

Module revision mask MR_MASK 0x00ff
EditMod Labels

1-module header
7-edition

Description

Indicates the software release level for maintenance. OS-9 does not use this field. Whenever a program is revised (even for a small change), increase this number. We recommend internal documentation within the source program be keyed to this system.

Port Generic Default Value

1

Port Specific Override Value

Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values

0 to 65535

Device Descriptor Data Definition Fields

The following section contains the device descriptor data definition fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd_port</td>
<td>PORTADDR</td>
</tr>
<tr>
<td>dd_lun</td>
<td>LUN</td>
</tr>
<tr>
<td>dd_pd_size</td>
<td>PD_SIZE</td>
</tr>
<tr>
<td>dd_type</td>
<td>DD_TYPE</td>
</tr>
<tr>
<td>dd_mode</td>
<td>DD_MODE</td>
</tr>
<tr>
<td>dd_port</td>
<td>MFGR_NAME</td>
</tr>
</tbody>
</table>
### Table 4-6. Device Descriptor Data Definition Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>drvr_name</td>
<td>DRVR_NAME</td>
</tr>
<tr>
<td>dd_class</td>
<td>DD_CLASS</td>
</tr>
</tbody>
</table>
**EditMod Labels**
- 2-device descriptor data definitions
- 1-device port address

**Description**
Absolute physical address of the hardware controller. This is the address of the device on the bus. This is the lowest address the device has mapped. Port address is hardware dependent.

**Macro Example**
```c
#define PORTADDR    0xfffe4000
```

**Port Generic Default Value**
0 (zero)

**Port Specific Override Value**
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

**Available Values**
0 to 4294967295
Chapter 4: SBF Device Descriptors

dd_lun
LUN

EditMod Labels
2-device descriptor data definitions
2-logical unit number

Description
Distinguishes the different devices driven from a unique controller. Each unique number represents a different logical unit static storage area.

Macro Example
#define LUN 2

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
0 to 65535
dd_pd_size
PD_SIZE

EditMod Labels
2-device descriptor data definitions
3-path descriptor size

Description
Size of the path descriptor. IOMAN uses this value when it allocates a path descriptor.

Port Generic Default Value
124

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
0 to 65535
**EditMod Labels**

2 - device descriptor data definitions
4 - device type

**Description**

Identifies the I/O class of the device.

**Port Generic Default Value**

Macro

```c
DT_SBF
```

**EditMod**

0x3

**Port Specific Override Value**

Refer to `SBF/<DEVICE>/DESC/config.des` (Figure 4-3).

**Available Values**

Device type values are defined in the header file `io.h`, and are listed in Table 4-7.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Character File Type</td>
<td>DT_SCF</td>
<td>0x0</td>
</tr>
<tr>
<td>Random Block File Type</td>
<td>DT_RBF</td>
<td>0x1</td>
</tr>
<tr>
<td>Pipe File Type</td>
<td>DT_PIPE</td>
<td>0x2</td>
</tr>
<tr>
<td>Sequential Block File Type</td>
<td>DT_SBF</td>
<td>0x3</td>
</tr>
<tr>
<td>Network File Type</td>
<td>DT_NFM</td>
<td>0x4</td>
</tr>
<tr>
<td>Compact Disc File Type</td>
<td>DT_CDFM</td>
<td>0x5</td>
</tr>
<tr>
<td>User Communication Manager</td>
<td>DT_UCM</td>
<td>0x6</td>
</tr>
<tr>
<td>Socket Communication Manager</td>
<td>DT_SOCKET</td>
<td>0x7</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>Pseudo-Keyboard Manager</td>
<td>DT_PTTY</td>
<td>0x8</td>
</tr>
<tr>
<td>Graphics File Manager</td>
<td>DT_GFM</td>
<td>0x9</td>
</tr>
<tr>
<td>PC-DOS File Manager</td>
<td>DT_PCF</td>
<td>0xa</td>
</tr>
<tr>
<td>Non-volatile RAM File Manager</td>
<td>DT_NRF</td>
<td>0xb</td>
</tr>
<tr>
<td>ISDN File Manager</td>
<td>DT_ISDN</td>
<td>0xc</td>
</tr>
<tr>
<td>MPFM File Manager</td>
<td>DT_MPFM</td>
<td>0xd</td>
</tr>
<tr>
<td>Real-Time Network File Manager</td>
<td>DT_RTNFM</td>
<td>0xe</td>
</tr>
<tr>
<td>Serial Protocol File Manager</td>
<td>DT_SPF</td>
<td>0xf</td>
</tr>
<tr>
<td>Inet File Manager</td>
<td>DT_INET</td>
<td>0xa0</td>
</tr>
<tr>
<td>Reserved for Microware Use Only</td>
<td>0xa1-0x7f</td>
<td></td>
</tr>
</tbody>
</table>

Table 4-7. \texttt{dd\_type} Available Values (Continued)
Chapter 4: SBF Device Descriptors

dd_mode
DD_MODE

EditMod Labels

2-device descriptor data definitions
5-device mode capabilities

Description

Used to check the validity of a caller’s access mode byte in \texttt{I\_CREATE} or \texttt{I\_OPEN} system calls. If a bit is set, the device can perform the corresponding function. The \texttt{S\_ISIZE} bit is usually set, because it is handled by the file manager or ignored. If the \texttt{S\_ISHARE} bit is set, the device is non-sharable. A printer is an example of a non-sharable device.

Port Generic Default Value

Macro
\[\texttt{S\_IPRM}\]

EditMod
\[0xFFFF\]

Port Specific Override Value

Refer to \texttt{SBF/\texttt{<DEVICE>/DESC/config.des}} (Figure 4-3).

Available Values

The file access modes are defined in the header file, \texttt{modes.h}, and located in Table 4-8. The file access permission values are defined in the header file \texttt{modes.h} and in Table 4-9.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|}
\hline
Description & Macro & EditMod \\
\hline
Truncate on open & \texttt{S\_ITRUNC} & 0x0100 \\
Ensure contiguous file & \texttt{S\_ICONTIG} & 0x0400 \\
Error if file exists on create & \texttt{S\_IEXCL} & 0x0400 \\
Create file & \texttt{S\_ICREAT} & 0x0800 \\
\hline
\end{tabular}
\caption{dd_mode Available Values for File Access Modes}
\end{table}
### Table 4-8. *dd_mode* Available Values for File Access Modes (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append to file</td>
<td>S_IAPPEND</td>
<td>0x1000</td>
</tr>
<tr>
<td>Non-sharable</td>
<td>S_ISHARE</td>
<td>0x4000</td>
</tr>
</tbody>
</table>

### Table 4-9. *dd_mode* Available Values for File Access Permissions

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask for permission bits</td>
<td>S_IPRM</td>
<td>0xffff</td>
</tr>
<tr>
<td>Owner read</td>
<td>S_IREAD</td>
<td>0x0001</td>
</tr>
<tr>
<td>Owner write</td>
<td>S_IWRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Owner execute</td>
<td>S_IEXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Search permission</td>
<td>S_ISEARCH</td>
<td>0x0004</td>
</tr>
<tr>
<td>Group read</td>
<td>S_IGREAD</td>
<td>0x0010</td>
</tr>
<tr>
<td>Group write</td>
<td>S_IGWRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Group execute</td>
<td>S_IGEXEC</td>
<td>0x0040</td>
</tr>
<tr>
<td>Group search</td>
<td>S_IGSEARCH</td>
<td>0x0040</td>
</tr>
<tr>
<td>Public read</td>
<td>S_IOREAD</td>
<td>0x0100</td>
</tr>
<tr>
<td>Public write</td>
<td>S_IOWRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Public execute</td>
<td>S_IOEXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>Public search</td>
<td>S_IOSEARCH</td>
<td>0x0400</td>
</tr>
</tbody>
</table>
Chapter 4: SBF Device Descriptors

**fmgr_name**
*FMGR_NAME*

**EditMod Labels**

2-device descriptor data definitions
6-file manager name

**Description**

Contains the name string of the file manager module to use.

**Port Generic Default Value**

“sbf”

**Port Specific Override Value**

Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

**Available Values**

Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
**EditMod Labels**
2-device descriptor data definitions
7-driver name

**Description**
Contains the name string of the device driver module to use.

**Port Generic Default Value**
NULL

**Port Specific Override Value**
Refer to SBF/<DEVICE>/DESC/config.des (*Figure 4-3*).

**Available Values**
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**dd_class**

**DD_CLASS**

**EditMod Labels**

1-module header

2-device descriptor data definitions

3-device class (sequential or random)

**Description**

Used to identify the class of the device, whether it is random or sequential access.

**Port Generic Default Value**

Macro

```
DC_SEQ
```

EditMod

0x1

**Port Specific Override Value**

Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

**Available Values**

Device class available values are defined in the header file, io.h, and in Table 4-10.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential access device</td>
<td>DC_SEQ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Random access device</td>
<td>DC_RND</td>
<td>0x0002</td>
</tr>
</tbody>
</table>

**SBF Path Options Fields**

The following section contains the SBF path options fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>pd_blksz</td>
<td>BLKSIZE</td>
</tr>
<tr>
<td>pd_flags</td>
<td>FLAGS</td>
</tr>
<tr>
<td>pd_dmamode</td>
<td>DMAMODE</td>
</tr>
<tr>
<td>pd_sci_id</td>
<td>SCSIID</td>
</tr>
<tr>
<td>pd_scsilun</td>
<td>SCSILUN</td>
</tr>
</tbody>
</table>
EditMod Labels
3-SBF path options structure
1-size of blocks allocated

Description
Logical block size in bytes.

Port Generic Default Value
512

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
0 to 4294967295
**pd_flags**

FLAGS

**EditMod Labels**

3-SBF path options structure
2-SBF/driver compatibility flags

**Description**

SBF driver compatibility flags.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

**Available Values**

Compatibility flag values are defined in the header file sbf.h, and in Table 4-12.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewind tape on close</td>
<td>DEV_REWIND_FLG</td>
<td>0x0001</td>
</tr>
<tr>
<td>Erase to end after writing</td>
<td>DEV_ERASE_FLG</td>
<td>0x0002</td>
</tr>
<tr>
<td>Take drive off-line on close</td>
<td>DEV_OFFLINE_FLG</td>
<td>0x0004</td>
</tr>
<tr>
<td>Device can skip backwards</td>
<td>DEV_SKIPBACK_FLG</td>
<td>0x0008</td>
</tr>
</tbody>
</table>
EditMod Labels
3-SBF path options structure
3-DMA type/usage

Description
DMA mode to be used by the driver.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
0 to 65535
**EditMod Labels**

3 - SBF path options structure
4 - SCSI controller ID

**Description**

SCSI ID of the device’s controller.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

**Available Values**

0 to 255
EditMod Labels
3-SBF path options structure
5-SCSI controller drive LUN

Description
Logical Unit Number of the tape device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
0 to 255

SBF Logical Unit Status Fields
The following section contains the SBF logical unit status fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>sbf_vector</td>
<td>VECTOR</td>
</tr>
<tr>
<td>sbf_irqlevel</td>
<td>IRQLEVEL</td>
</tr>
<tr>
<td>sbf_priority</td>
<td>PRIORITY</td>
</tr>
<tr>
<td>sbf_dflag</td>
<td>DRIVE_FLAG</td>
</tr>
</tbody>
</table>
**EditMod Labels**

4-SBF logical unit status
1-irq vector

**Description**

This is the vector number of the device interrupt.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

**Available Values**

0 to 255
### sbf_irqlevel

<table>
<thead>
<tr>
<th>EditMod Labels</th>
<th>Description</th>
<th>Port Generic Default Value</th>
<th>Port Specific Override Value</th>
<th>Available Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-SBF logical unit status</td>
<td>This is the hardware priority of the device interrupt.</td>
<td>0 (zero)</td>
<td>Refer to SBF/&lt;DEVICE&gt;/DESC/config.des (Figure 4-3).</td>
<td>0 to 255</td>
</tr>
</tbody>
</table>
EditMod Labels
4-SBF logical unit status
3-irq priority

Description
This is the software (polling) priority of the device interrupt.

Port Generic Default Value
5

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
0 to 255
EditMod Labels
4-SBF logical unit status
4-drive flag

Description
Current state of SBF device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to SBF/<DEVICE>/DESC/config.des (Figure 4-3).

Available Values
Drive flag values are defined in the header file sbf.h, and in Table 4-14.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read is in progress on device</td>
<td>DFLG_READFLAG</td>
<td>0x0001</td>
</tr>
<tr>
<td>Write is in progress on device</td>
<td>DFLG_WRITEFLAG</td>
<td>0x0002</td>
</tr>
<tr>
<td>Driver is using the device</td>
<td>DFLG_DRIVEBUSY</td>
<td>0x0004</td>
</tr>
<tr>
<td>Drive is at EOF</td>
<td>DFLG_EOFFLAG</td>
<td>0x0008</td>
</tr>
</tbody>
</table>
RBF Device Descriptors

RBF device descriptors contain configuration data specific to one OS-9 format disk device on an OS-9 system. Values that can be configured in the descriptor include:

- Device interrupt vector and priority
- Device I/O address
- Device geometry
- Logical sector size

The next section in this chapter provides a detailed example of the configuration options you can use to change configuration values for RBF (random block file) devices.

The rest of this chapter provides a detailed list of all of the RBF device descriptor fields.

This chapter includes the following topics:

RBF Field Configuration Options
RBF Device Descriptor Field Reference
Module Header Fields
Device Descriptor Data Definition Fields
RBF Path Option Fields
RBF Logical Unit Static Storage Fields
RBF Logical Unit Options
RBF Field Configuration Options

To change an RBF device descriptor module configuration field, you can use either of the following methods:

1. Use the EditMod utility to directly modify existing RBF device descriptor modules either as a stand-alone module or as part of a merged module group (such as a boot image).

2. Modify the description file for the RBF device descriptor module and rebuild it using the makefile provided.

Direct Modification Advantages

The direct modification method has the following advantages:

- **Fast**: No source configuration file rebuilds are necessary.
- **Temporary**: The original module or merged-module group configuration can be easily restored through the appropriate rebuild.
- **Contained**: Changes are limited to the individual boot image modified (merged-module option).

Description File/Rebuild Advantages

The advantage of the description file/rebuild method is that the changes are permanent and reproducible. Modifications apply to all subsequent module rebuilds and to all merged-module groups built containing the updated module.

Both methods are documented in this section. These procedures are used with the field descriptions starting with the Module Header Fields. For direct modification, use the EditMod LABELS data to navigate through the EditMod menus. The DESCRIPTION FILE MACRO data identifies the macro you need to define/modify in the configuration sources to rebuild the RBF device descriptor module.

Direct Modification

Use the EditMod utility and the following procedures to directly modify fields in the existing RBF device descriptor module. The module can stand-alone or it can be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The field references later in this chapter contain a description of each configurable field, its supported values, and the sequence of menu options required by EditMod to modify that field.

Refer to the Utilities Reference for a full description of EditMod’s capabilities.
Refer to your board guide for information about how to modify the module lists and remake the boot images, and for specified boot image names.

**Direct Modification Procedures**

To modify the stand-alone module, complete the following steps:

1. Change to the `CMDS/BOOTOJBS/DESC/<DEVICE>` directory (see Figure 5-1).
2. Use `EditMod` to edit the module:
   ```bash
   $EditMod -e <descriptor>
   ```

To modify the module as part of a merged module group, complete the following steps:

1. Change to the `BOOTS/SYSTEMS/PORTBOOT` directory (see Figure 5-2).
2. Use `EditMod` to edit the module:
   ```bash
   $EditMod -e <descriptor> -f=<boot image name>
   ```
3. Use the menu selections provided in the `EditMod` LABELS section of the field reference later in this chapter to locate the fields you want to edit.
4. Select a new value for the field from the AVAILABLE VALUES section of the field reference. Enter that value at the EditMod prompt to modify the field.

5. If you want to make additional modifications, use the p command (previous) to step backward through the EditMod menus. Repeat Steps 3 and 4 until you have made all desired modifications to the descriptor.

6. Select the w command (write) to save the changes.

7. Select the q command (quit) to exit EditMod.

Unless you modified the RBF device descriptors in your boot image, you should rebuild your boot image to include the new descriptor.

Example EditMod Session
This example modifies an RBF device descriptor as part of the boot image rom:

```
$ EditMod -e r0 -f=rom
```

1. module header
2. device descriptor data definitions
3. RBF path options
4. RBF logical unit static storage

Which? [?/1-4/p/t/a/w/q] 4

1. interrupt vector : 0x0
2. interrupt level : 0
3. interrupt priority : 5
4. RBF logical unit options

Which? [?/1-4/p/t/a/w/q] 3

interrupt priority : 5
New value: 1

1. interrupt vector : 0x0
2. interrupt level : 0
3. interrupt priority : 1
4. RBF logical unit options

Which? [?/1-19/p/t/a/w/q] w

Which? [?/1-19/p/t/a/w/q] q
Description File Configuration

You can use these procedures to modify the appropriate description file and rebuild the RBF device descriptors for your port directory. The DESCRIPTION FILE MACROS section of the field reference specifies the name of the macro you modify/define in the description files to configure the field. The value used in the define is chosen from the AVAILABLE VALUES specified for the field.

Figure 5-3. Directory Location for Modifying RBF Description Files

Description File Configuration Procedures

1. Change to the RBF/<DEVICE> directory (see Figure 5-3).
2. Edit the file config.des and read the included comments for more information on how to use the specific description files provided in your software distribution. The config.des file contains a list of macro names that can be defined to override the global default values for the configuration fields.
3. Refer to the DESCRIPTION FILE MACRO section in the field reference later in this chapter to determine the macro name you define to configure the target field.
4. Read the comments in config.des to determine where to place the define for this macro.
5. Select the value you want to use to configure the field. See the AVAILABLE VALUES section of the field reference data for values or macros that can be used for the definition. Define the macro by entering a definition in the appropriate description files as follows:
   
   #define <macro> <value>

6. Save the changes and rebuild the RBF device descriptors, entering the following command in the RBF/<DEVICE>/DESC directory:

   os9make

7. Rebuild your boot image to include the new descriptor.

RBF Device Descriptor Field Reference

This section contains a list of the most commonly configured fields in the RBF device descriptors. Each field entry contains the following information:

- <Field name> - The call name for each field that can be reconfigured in the module.
• **EditMod LABELS** - EditMod menu selections for navigating to the proper field in an EditMod session.

• **DESCRIPTION FILE MACRO** - The macro name you modify/define in the description file.

• **DESCRIPTION** - A brief description of the field’s purpose and use.

• **EXAMPLE** - An optional example of the description file entry showing how to change the value of this field.

• **PORT GENERIC DEFAULT VALUE** - The value set in the port generic description file for this field. This is the value the field is assigned when the module is built, unless the appropriate macro has been defined in the port specific description file to override this default value.

• **PORT SPECIFIC OVERRIDE VALUE** - The value set in the port specific description file for this field. If defined, this is the value the field is assigned when the module is built, overriding the port generic default value.

• **AVAILABLE VALUES** - Values to which the field can be set through EditMod or the description files. In many cases, this data is presented in a table that maps a description of the value to a numeric value appropriate for entry in EditMod, and to a pre-defined macro available for use in the description file.

### Module Header Fields

The following section contains the module header fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>_m_group</td>
<td>MH_GROUP</td>
</tr>
<tr>
<td>_m_user</td>
<td>MH_USER</td>
</tr>
<tr>
<td>mod_name</td>
<td>MH_NAME</td>
</tr>
<tr>
<td>m_access</td>
<td>MH_ACCESS</td>
</tr>
<tr>
<td>m_tylan</td>
<td>MH_TYLAN</td>
</tr>
<tr>
<td>m_attrev</td>
<td>MH_ATTREV</td>
</tr>
<tr>
<td>m_edit</td>
<td>MH_EDITION</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
1-module owner’s group number

Description
Group ID of the module’s owner. The group number allows people working in the same department or on the same project to share a common identification number.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
0 to 65535
EditMod Labels
1-module header
2-module owner’s user number

Description
User ID of the module’s owner. The user number identifies a specific user.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
0 to 65535
EditMod Labels
1-module header
3-module name

Description
Contains the module name string.

Port Generic Default Value
String value (None)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels

- module header
- access permissions

**Description**

Defines the permissible module access by its owner or by other users.

**Port Generic Default Value**

Macro

```
MP_OWNER_READ | MP_OWNER_EXEC | MP_GROUP_READ |
MP_GROUP_EXEC | MP_WORLD_READ | MP_WORLD_EXEC
```

**EditMod**

0x555

**Port Specific Override Value**

Refer to `RBF/<DEVICE>/DESC/config.des (Figure 5-3)`.

**Available Values**

Module access permission values are located in the header file, `module.h`, and are listed in Table 5-2.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read permission by owner</td>
<td>MP_OWNER_READ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Write permission by owner</td>
<td>MP_OWNER_WRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Execute permission by owner</td>
<td>MP_OWNER_EXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Owner permission mask</td>
<td>MP_OWNER_MASK</td>
<td>0x000f</td>
</tr>
<tr>
<td>Read permission by group</td>
<td>MP_GROUP_READ</td>
<td>0x0010</td>
</tr>
<tr>
<td>Write permission by group</td>
<td>MP_GROUP_WRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Execute permission by group</td>
<td>MP_GROUP_EXEC</td>
<td>0x0040</td>
</tr>
</tbody>
</table>
### Table 5-2: Available Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group permission mask</td>
<td>MP_GROUP_MASK</td>
<td>0x00f0</td>
</tr>
<tr>
<td>Read permission by world</td>
<td>MP_WORLD_READ</td>
<td>0x0100</td>
</tr>
<tr>
<td>Write permission by world</td>
<td>MP_WORLD_WRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Execute permission by world</td>
<td>MP_WORLD_EXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>World permission mask</td>
<td>MP_WORLD_MASK</td>
<td>0x0f00</td>
</tr>
<tr>
<td>All permissions for owner, group, and world</td>
<td>MP_WORLD_ACCESS</td>
<td>0x0777</td>
</tr>
<tr>
<td>System permission mask</td>
<td>MP_SYSTM_MASK</td>
<td>0xf000</td>
</tr>
</tbody>
</table>
EditMod Labels

- Module header
- Type/language

Description

Contains the module’s type (first byte) and language (second byte). The language codes indicate if the module is executable and which language the run-time system requires for execution, if any.

Port Generic Default Value

Macro

\[(\text{MT\_DATA} \ll 8) + \text{ML\_OBJECT}\]

EditMod

0x401

Port Specific Override Value

Refer to \texttt{RBF/<DEVICE>/DESC/config.des} (Figure 5-3).

Available Values

Module type values and language codes are located in the header file, \texttt{module.h}, and are listed in Table 5-3 and Table 5-4.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not used (wildcard value in system calls)</td>
<td>MT_ANY</td>
<td>0x0000</td>
</tr>
<tr>
<td>Program module</td>
<td>MT_PROGRAM</td>
<td>0x0001</td>
</tr>
<tr>
<td>Subroutine module</td>
<td>MT_SUBROUT</td>
<td>0x0002</td>
</tr>
<tr>
<td>Multi-module (reserved for future use)</td>
<td>MT_MULTI</td>
<td>0x0003</td>
</tr>
<tr>
<td>Data module</td>
<td>MT_DATA</td>
<td>0x0004</td>
</tr>
<tr>
<td>Configuration data block data module</td>
<td>MT_CDBDATA</td>
<td>0x0005</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0xb-0xa</td>
<td>0xb-0xa</td>
</tr>
</tbody>
</table>
### Table 5-3. m_tylan Available Module Type Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>User trap library</td>
<td>MT_TRPLIB</td>
<td>0x000b</td>
</tr>
<tr>
<td>System module</td>
<td>MT_SYSTEM</td>
<td>0x000c</td>
</tr>
<tr>
<td>File manager module</td>
<td>MT_FILEMAN</td>
<td>0x000d</td>
</tr>
<tr>
<td>Physical device driver</td>
<td>MT_DEVDRVR</td>
<td>0x000e</td>
</tr>
<tr>
<td>Device descriptor module</td>
<td>MT_DEVDESC</td>
<td>0x000f</td>
</tr>
<tr>
<td>User definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module type mask</td>
<td>MT_MASK</td>
<td>0xff00</td>
</tr>
</tbody>
</table>

### Table 5-4. m_tylan Available Language Code Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unspecified language (wildcard in system calls)</td>
<td>ML_ANY</td>
<td>0x0</td>
</tr>
<tr>
<td>Machine language</td>
<td>ML_OBJECT</td>
<td>0x1</td>
</tr>
<tr>
<td>Basic I-code (reserved for future use)</td>
<td>ML_ICODE</td>
<td>0x2</td>
</tr>
<tr>
<td>Pascal P-code (reserved for future use)</td>
<td>ML_PCODE</td>
<td>0x3</td>
</tr>
<tr>
<td>C I-code (reserved for future use)</td>
<td>ML_CCODE</td>
<td>0x4</td>
</tr>
<tr>
<td>Cobol I-code (reserved for future use)</td>
<td>ML_CBLCODE</td>
<td>0x5</td>
</tr>
<tr>
<td>Fortran</td>
<td>ML_FRTNCODE</td>
<td>0x6</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0x7-0xf</td>
<td>0x7-0xf</td>
</tr>
<tr>
<td>User-definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module language mask</td>
<td>ML_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
6-revision/attributes

Description
Contains the module’s attributes (first byte) and revision (second byte).

Port Generic Default Value
Macro
MA_REENT<<8

EditMod
0x8000

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
Module attribute and revision codes are located in the header file module.h., and are listed in Table 5-5.

If two modules with the same name are found in the memory search or are loaded into the current module directory, only the module with the highest revision level is kept. This enables easy substitution of modules for update or correction.

Table 5-5. m_attrev Available Attribute and Revision Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is re-entrant (sharable by multiple tasks).</td>
<td>MA_REENT (shifted left to first byte: MA_REENT&lt;&lt;8)</td>
<td>0x80 (shifted left to first byte: 0x8000)</td>
</tr>
<tr>
<td>The module is sticky. A sticky module is not removed from memory until its link count becomes -1 or memory is required for another use.</td>
<td>MA_GHOST (shifted left to first byte: MA_GHOST&lt;&lt;8)</td>
<td>0x40 (shifted left to first byte: 0x4000)</td>
</tr>
</tbody>
</table>
Table 5-5. Available Attribute and Revision Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is a system-state module.</td>
<td>MA_SUPER</td>
<td>0x20 (shifted left to first byte: 0x2000)</td>
</tr>
<tr>
<td></td>
<td>(shifted left to first byte: MA_SUPER&lt;&lt;8)</td>
<td></td>
</tr>
<tr>
<td>User-definable revision number</td>
<td>0x0-0xfe</td>
<td>0x0-0xfe</td>
</tr>
<tr>
<td>Module attribute mask</td>
<td>MA_MASK</td>
<td>0xff00</td>
</tr>
<tr>
<td>Module revision mask</td>
<td>MR_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
7-edition

Description

Indicates the software release level for maintenance. OS-9 does not use this field. Whenever a program is revised (even for a small change), increase this number. We recommend internal documentation within the source program be keyed to this system.

Port Generic Default Value

1

Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values

0 to 65535

Device Descriptor Data Definition Fields

The following section contains the device descriptor data definition fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd_port</td>
<td>PORTADDR</td>
</tr>
<tr>
<td>dd_lun</td>
<td>LUN</td>
</tr>
<tr>
<td>dd_pd_size</td>
<td>PD_SIZE</td>
</tr>
<tr>
<td>dd_type</td>
<td>DD_TYPE</td>
</tr>
<tr>
<td>dd_mode</td>
<td>DD_MODE</td>
</tr>
<tr>
<td>dd_port</td>
<td>MFGR_NAME</td>
</tr>
</tbody>
</table>
### Table 5-6. Device Descriptor Data Definition Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>drvr_name</code></td>
<td>DRVR_NAME</td>
</tr>
<tr>
<td><code>dd_class</code></td>
<td>DD_CLASS</td>
</tr>
</tbody>
</table>
EditMod Labels
2-device descriptor data definitions
1-device port address

Description
Absolute physical address of the hardware controller. This is the address of the device on the bus. This is the lowest address the device has mapped. Port address is hardware dependent.

Macro Example
#define PORTADDR    0xfffe4000

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
0 to 4294967295
**dd_lun**

**LUN**

**EditMod Labels**

2-device descriptor data definitions
2-logical unit number

**Description**

Distinguishes between the different devices driven from a unique controller. Each unique number represents a different logical unit static storage area.

**Macro Example**

```c
#define LUN 2
```

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

**Available Values**

0 to 65535
dd_pd_size
PD_SIZE

EditMod Labels
2-device descriptor data definitions
3-path descriptor size

Description
Size of the path descriptor. IOMAN uses this value when it allocates a path descriptor.

Port Generic Default Value
360

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
0 to 65535
**dd_type**

**DD_TYPE**

**EditMod Labels**

2-device descriptor data definitions

4-device type

**Description**

Identifies the I/O class of the device.

**Port Generic Default Value**

Macro

```
DT_RBF
```

EditMod

0x1

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

**Available Values**

Device type values are defined in the header file io.h, and are listed in Table 5-7.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Character File Type</td>
<td>DT_SCF</td>
<td>0x0</td>
</tr>
<tr>
<td>Random Block File Type</td>
<td>DT_RBF</td>
<td>0x1</td>
</tr>
<tr>
<td>Pipe File Type</td>
<td>DT_PIPE</td>
<td>0x2</td>
</tr>
<tr>
<td>Sequential Block File Type</td>
<td>DT_SBF</td>
<td>0x3</td>
</tr>
<tr>
<td>Network File Type</td>
<td>DT_NFM</td>
<td>0x4</td>
</tr>
<tr>
<td>Compact Disc File Type</td>
<td>DT_CDFM</td>
<td>0x5</td>
</tr>
<tr>
<td>User Communication Manager</td>
<td>DT_UCM</td>
<td>0x6</td>
</tr>
<tr>
<td>Socket Communication Manager</td>
<td>DT_SOCK</td>
<td>0x7</td>
</tr>
</tbody>
</table>
Table 5-7. *dd_type* Available Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo-Keyboard Manager</td>
<td>DT_PTTY</td>
<td>0x8</td>
</tr>
<tr>
<td>Graphics File Manager</td>
<td>DT_GFM</td>
<td>0x9</td>
</tr>
<tr>
<td>PC-DOS File Manager</td>
<td>DT_PCF</td>
<td>0xa</td>
</tr>
<tr>
<td>Non-volatile RAM File Manager</td>
<td>DT_NRF</td>
<td>0xb</td>
</tr>
<tr>
<td>ISDN File Manager</td>
<td>DT_ISDN</td>
<td>0xc</td>
</tr>
<tr>
<td>MPFM File Manager</td>
<td>DT_MPFM</td>
<td>0xd</td>
</tr>
<tr>
<td>Real-Time Network File Manager</td>
<td>DT_RTNFM</td>
<td>0xe</td>
</tr>
<tr>
<td>Serial Protocol File Manager</td>
<td>DT_SPF</td>
<td>0xf</td>
</tr>
<tr>
<td>Inet File Manager</td>
<td>DT_INET</td>
<td>0xa0</td>
</tr>
<tr>
<td>Reserved for Microware Use Only</td>
<td>17-127</td>
<td>0xa1-0x7f</td>
</tr>
</tbody>
</table>
**dd_mode**

**DD_MODE**

**EditMod Labels**

2-device descriptor data definitions  
5-device mode capabilities  

**Description**

Used to check the validity of a caller’s access mode byte in `I_CREATE` or `I_OPEN` system calls. If a bit is set, the device can perform the corresponding function. The `S_ISIZE` bit is usually set, because it is handled by the file manager or ignored. If the `S_ISHARE` bit is set, the device is non-sharable. A printer is an example of a non-sharable device.

**Port Generic Default Value**

Macro  

`S_IPRM`

EditMod  

`0xFFFF`

**Port Specific Override Value**

Refer to `RBF/<DEVICE>/DESC/config.des` (Figure 5-3).

**Available Values**

The file access modes are defined in the header file, `modes.h`, and located in Table 5-8. The file access permission values are defined in the header file `modes.h` and in Table 5-9.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate on open</td>
<td><code>S_ITRUNC</code></td>
<td>0x0100</td>
</tr>
<tr>
<td>Ensure contiguous file</td>
<td><code>S_ICONTIG</code></td>
<td>0x0400</td>
</tr>
<tr>
<td>Error if file exists on create</td>
<td><code>S_IEXCL</code></td>
<td>0x0400</td>
</tr>
<tr>
<td>Create file</td>
<td><code>S_ICREAT</code></td>
<td>0x0800</td>
</tr>
</tbody>
</table>
### Table 5-8. \texttt{dd\_mode} Available Values for File Access Modes (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append to file</td>
<td>S_IAPPEND</td>
<td>0x1000</td>
</tr>
<tr>
<td>Non-sharable</td>
<td>S_IShare</td>
<td>0x4000</td>
</tr>
</tbody>
</table>

### Table 5-9. \texttt{dd\_mode} Available Values for File Access Permissions

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask for permission bits</td>
<td>S_IPRM</td>
<td>0xffff</td>
</tr>
<tr>
<td>Owner read</td>
<td>S_IREAD</td>
<td>0x0001</td>
</tr>
<tr>
<td>Owner write</td>
<td>S_IWRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Owner execute</td>
<td>S_IEXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Search permission</td>
<td>S_ISEARCH</td>
<td>0x0004</td>
</tr>
<tr>
<td>Group read</td>
<td>S_IGREAD</td>
<td>0x0010</td>
</tr>
<tr>
<td>Group write</td>
<td>S_IGWRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Group execute</td>
<td>S_IGEXEC</td>
<td>0x0040</td>
</tr>
<tr>
<td>Group search</td>
<td>S_IGSEARCH</td>
<td>0x0040</td>
</tr>
<tr>
<td>Public read</td>
<td>S_IOREAD</td>
<td>0x0100</td>
</tr>
<tr>
<td>Public write</td>
<td>S_IOWRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Public execute</td>
<td>S_IOEXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>Public search</td>
<td>S_IOSEARCH</td>
<td>0x0400</td>
</tr>
</tbody>
</table>
EditMod Labels
2-device descriptor data definitions
6-file manager name

Description
Contains the name string of the file manager module to use.

Port Generic Default Value
“rbf”

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
2-device descriptor data definitions
7-driver name

Description
Contains the name string of the device driver module to use.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
**dd_class**

**Chapter 5: RBF Device Descriptors**

**EditMod Labels**

1-module header
2-device descriptor data definitions
8-device class (sequential or random)

**Description**

Used to identify the class of the device, whether it is random or sequential access.

**Port Generic Default Value**

Macro

```
DC_RND
```

EditMod

0x2

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

**Available Values**

Device class available values are defined in the header file, io.h, and in Table 5-10.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential access device</td>
<td>DC_SEQ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Random access device</td>
<td>DC_RND</td>
<td>0x0002</td>
</tr>
</tbody>
</table>

**RBF Path Option Fields**

The following section contains the RBF path option fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>pd_sid</td>
<td>SIDES</td>
</tr>
<tr>
<td>pd_vfy</td>
<td>VERIFY</td>
</tr>
<tr>
<td>pd_format</td>
<td>FORMAT</td>
</tr>
<tr>
<td>pd_cyl</td>
<td>CYLNDRS</td>
</tr>
<tr>
<td>pd_blk</td>
<td>BLKSTRK</td>
</tr>
<tr>
<td>pd_t0b</td>
<td>BLKSTRKO</td>
</tr>
<tr>
<td>pd_sas</td>
<td>SEGSIZE</td>
</tr>
<tr>
<td>pd_ilv</td>
<td>INTRLV</td>
</tr>
<tr>
<td>pd_toffs</td>
<td>TRKOFFS</td>
</tr>
<tr>
<td>pd_boffs</td>
<td>BLKOFFS</td>
</tr>
<tr>
<td>pd_trys</td>
<td>TRYS</td>
</tr>
<tr>
<td>pd_bsize</td>
<td>BLKSIZE</td>
</tr>
<tr>
<td>pd_cntl</td>
<td>CONTROL</td>
</tr>
<tr>
<td>pd_wpc</td>
<td>PRECOMP</td>
</tr>
<tr>
<td>pd_rwr</td>
<td>REDWRITE</td>
</tr>
<tr>
<td>pd_park</td>
<td>PARK</td>
</tr>
<tr>
<td>pd_lsnoffs</td>
<td>LSNOFFS</td>
</tr>
<tr>
<td>pd_xfersize</td>
<td>Xfersize</td>
</tr>
</tbody>
</table>
**EditMod Labels**

- **3-RBF path options**
- **1-number of surfaces**

**Description**

Indicates the number of surfaces (heads or sides) for a disk unit.

**Port Generic Default Value**

2

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

**Available Values**

-2147483648 to 2147483647
EditMod Labels

3-RBF path options
2-verify disk writes (0=verify)

Description
Indicates whether a write is verified by a re-read and compare. Write verify operations are generally performed on floppy disks but not hard disks because of the lower soft error rate of hard disks.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
Device verify values are defined in the header file, rbf.h, and in Table 5-12.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify disk write</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>No verification</td>
<td>1</td>
<td>0x01</td>
</tr>
</tbody>
</table>

Table 5-12. pd_vfy Available Values
EditMod Labels

3-RBF path options
3-device format

Description
Indicates whether a write is verified by a re-read and compare. Write verify operations are generally performed on floppy disks but not hard disks because of the lower soft error rate of hard disks.

Port Generic Default Value

Macro

\[
\text{FMT_STDFMT + FMT_DBLBITDNS + FMT_DBLTRKDNS + FMT_DBLSIDE}
\]

EditMod

0x200e

Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values

Device format values are defined in the header file, rbf.h, and in Table 5-13.

Table 5-13. pd_format Available Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track 0 is double density.</td>
<td>FMT_DBLTRK0</td>
<td>0x0001</td>
</tr>
<tr>
<td>Device is double bit density.</td>
<td>FMT_DBLBITDNS</td>
<td>0x0002</td>
</tr>
<tr>
<td>Device is double track density.</td>
<td>FMT_DBLTRKDNS</td>
<td>0x0004</td>
</tr>
<tr>
<td>Device is double sided.</td>
<td>FMT_DBLSIDE</td>
<td>0x0008</td>
</tr>
<tr>
<td>Drive is eight inch.</td>
<td>FMT_EIGHTINCH</td>
<td>0x0010</td>
</tr>
<tr>
<td>Drive is five inch.</td>
<td>FMT_FIVEINCH</td>
<td>0x0020</td>
</tr>
<tr>
<td>Drive is three inch.</td>
<td>FMT_THREEINCH</td>
<td>0x0040</td>
</tr>
<tr>
<td>Device is high density.</td>
<td>FMT_HIGHDENS</td>
<td>0x1000</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>Device is standard format.</td>
<td>FMT_STDFMT</td>
<td>0x2000</td>
</tr>
<tr>
<td>Media can be removed.</td>
<td>FMT_REMOVABLE</td>
<td>0x4000</td>
</tr>
<tr>
<td>Device is a hard disk.</td>
<td>FMT_HARDISK</td>
<td>0x8000</td>
</tr>
</tbody>
</table>
EditMod Labels

3-RBF path options
4-number of cylinders

Description

Indicates the number of cylinders per disk.

Port Generic Default Value

80

Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values

-2147483648 to 2147483647
EditMod Labels
3-RBF path options
5-default blocks/track

Description
Indicates the number of blocks per track on the disk for all tracks except track 0. (See pd_t0b for track 0 information.)

Port Generic Default Value
2048

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-RBF path options
6-default blocks/track for trk0

Description
Indicates the number of blocks per track 0 on the disk. Depending on the device, this may be a different number for track 0 than the other tracks on the disk.

Port Generic Default Value
10

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-2147483648 to 2147483647
EditMod Labels

3-RBF path options
7-segment allocation size

Description
This value specifies the default minimum number of sectors to be allocated when a file is expanded.

Port Generic Default Value
1

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-2147483648 to 2147483647
<table>
<thead>
<tr>
<th>EditMod Labels</th>
<th>Description</th>
<th>Port Generic Default Value</th>
<th>Port Specific Override Value</th>
<th>Available Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-RBF path options</td>
<td>This value determines the sector interleave factor. Sectors are arranged on a disk in a certain sequential order (1, 2, 3, ... or 1, 3, 5, ...). The interleave factor determines the arrangement. For example, if the interleave factor is 2, the sectors would be arranged by twos, (1,3,5,...) starting at the base sector. See <code>pd_boffs</code> for base sector information.</td>
<td>3</td>
<td>Refer to RBF/&lt;DEVICE&gt;/DESC/config.des (Figure 5-3).</td>
<td>-2147483648 to 2147483647</td>
</tr>
</tbody>
</table>
EditMod Labels
3-RBF path options
9-track base offset

Description
This is the offset to the first accessible track number. Because Track 0 is often a different density, Track 0 is sometimes not used as the base track.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-RBF path options
10-block base offset

Description
This is the offset to the first accessible sector number. Because Sector 0 is not always the base sector.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-2147483648 to 2147483647
EditMod Labels

3-RBF path options

11-# tries

Description

This is the number of times a device tries to access a disk before returning an error.

Port Generic Default Value

7

Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values

-2147483648 to 2147483647
EditMod Labels
3-RBF path options
12-size of block in bytes

Description
This is the logical block size in bytes.

Port Generic Default Value
256 (256 characters)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-RBF path options
13-control word

Description
This is the device control word.

Port Generic Default Value
Macro
CTRL_AUTOSIZE

EditMod
0x2

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
Control word values are defined in the header file, rbf.h, and in Table 5-14.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable formatting of the device</td>
<td>CTRL_FMTDIS</td>
<td>0x0</td>
</tr>
<tr>
<td>Device is capable of multi-sector transfers</td>
<td>CTRL_MULTI</td>
<td>0x1</td>
</tr>
<tr>
<td>Device size can be obtained from device</td>
<td>CTRL_AUTOSIZE</td>
<td>0x2</td>
</tr>
<tr>
<td>Device requires only one format command</td>
<td>CTRL_FMTENTIRE</td>
<td>0x3</td>
</tr>
<tr>
<td>Device needs a full track buffer for format</td>
<td>CTRL_TRKWRITE</td>
<td>0x4</td>
</tr>
</tbody>
</table>
EditMod Labels

3-RBF path options
14-first write precomp cylinder

Description

This number indicates at which cylinder to begin write precompensation. Only older disk drives require this information, such as MFM or RLL drives.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values

-2147483648 to 2147483647
**pd_rwr**

**REDWRITE**

**EditMod Labels**

3-RBF path options

15-first reduced write current cylinder

**Description**

This number indicates at which cylinder to begin reduced write current. Only older disk drives require this information, such as MFM or RLL drives.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

**Available Values**

-2147483648 to 2147483647
EditMod Labels
3-RBF path options
16-park cylinder for hard disks

Description
This is the cylinder where the hard disk heads should be parked when the drive is shut down.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-2147483648 to 2147483647
EditMod Labels

3 - RBF path options
17- lsn offset for partition

Description
This is the offset to be used when accessing a partitioned drive.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-2147483648 to 2147483647
Chapter 5: RBF Device Descriptors

**EditMod Labels**

3-RBF path options
18-max transfer size in terms of bytes

**Description**

This is the maximum size of memory the controller can transfer at one time. The size is specified in bytes.

**Port Generic Default Value**

0xff00

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

**Available Values**

0 to 4294967295

**RBF Logical Unit Static Storage Fields**

The following section contains the RBF logical unit static storage fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>v_vector</td>
<td>VECTOR</td>
</tr>
<tr>
<td>v_irqlevel</td>
<td>IRQLEVEL</td>
</tr>
<tr>
<td>v_priority</td>
<td>PRIORITY</td>
</tr>
</tbody>
</table>
EditMod Labels
4-RBF logical unit static storage
1-interrupt vector

Description
This is the vector number of the device interrupt.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
0 to 255
EditMod Labels
4-RBF logical unit static storage
2-interrupt level

Description
This is the hardware priority of the device interrupt.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-128 to 127
**EditMod Labels**

4-RBF logical unit static storage
3-interrupt priority

**Description**

This is the software (polling) priority of the device interrupt.

**Port Generic Default Value**

5

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

**Available Values**

-128 to 127

**RBF Logical Unit Options**

The following section contains the RBF logical unit options fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>lu_stp</td>
<td>STEP</td>
</tr>
<tr>
<td>lu_tfm</td>
<td>DMAMODE</td>
</tr>
<tr>
<td>lu_lun</td>
<td>SCSILUN</td>
</tr>
<tr>
<td>lu_ctrlrid</td>
<td>CTRLRID</td>
</tr>
<tr>
<td>lu_totcyls</td>
<td>TOTCYLS</td>
</tr>
</tbody>
</table>
EditMod Labels

4-RBF logical unit static storage
4-RBF logical unit options
1-step rate

Description

This code sets the head stepping rate used with the drive. Set the step rate to the fastest value the drive is capable of to reduce access time.

Port Generic Default Value

Macro
STEP_30MS

EditMod
0x00

Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values

Step rate values are defined in the header file, rbf.h, and in Table 5-17.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 millisecond step rate</td>
<td>STEP_30MS</td>
<td>0x00</td>
</tr>
<tr>
<td>20 millisecond step rate</td>
<td>STEP_20MS</td>
<td>0x01</td>
</tr>
<tr>
<td>12 millisecond step rate</td>
<td>STEP_12MS</td>
<td>0x02</td>
</tr>
<tr>
<td>6 millisecond step rate</td>
<td>STEP_6MS</td>
<td>0x03</td>
</tr>
</tbody>
</table>
EditMod Labels

4-RBF logical unit static storage
4-RBF logical unit options
2-dma transfer mode

Description

This hardware specific byte can be set for use of DMA mode, if it is available. DMA requires only a single interrupt for each block of characters transferred in an I/O operation. It is much faster than methods that interrupt for each character transferred.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values

-128 to 127
EditMod Labels

4-RBF logical unit static storage
4-RBF logical unit options
3-drive logical unit number

Description
This number is used in the command block to identify the drive to the controller. The driver uses this number when specifying the device.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-128 to 127
EditMod Labels

4-RBF logical unit static storage
4-RBF logical unit options
4-controller ID

Description
This is the identification number of the controller attached to the drive. The drive uses this number when communicating with the controller.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values
-128 to 127
**EditMod Labels**

4-RBF logical unit static storage
4-RBF logical unit options
5-total number of cylinders

**Description**

This is the actual number of cylinders on a partitioned drive. The driver uses this value to correctly initialize the drive.

**Port Generic Default Value**

5

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

**Available Values**

-2147483648 to 2147483647
PCF device descriptors contain configuration data specific to one OS-9 format disk device on an OS-9 system. Values which can be configured in the descriptor include:

- Device interrupt vector and priority
- Device I/O address
- Device geometry
- Logical sector size

The next section in this chapter provides a detailed example of the configuration options you can use to change configuration values for PCF (PC-DOS file) devices.

The rest of this chapter provides a detailed list of all of the PCF device descriptor fields.

This chapter includes the following topics:

- PCF Field Configuration Options
- PCF Device Descriptor Field Reference
- Module Header Fields
- Device Descriptor Data Definition Fields
- PCF Path Option Fields
- PCF Logical Unit Static Storage Fields
- PCF Logical Unit Options
PCF Field Configuration Options

To change a PCF device descriptor module configuration field, you can use either of the following methods:

1. Use the EditMod utility to directly modify existing PCF device descriptor modules either as a stand-alone module or as part of a merged module group (such as a boot image).
2. Modify the description file for the PCF device descriptor module and rebuild it using the makefile provided.

Direct Modification Advantages

The direct modification method has the following advantages:

- **Fast**: No source configuration file rebuilds are necessary.
- **Temporary**: The original module or merged-module group configuration can be easily restored through the appropriate rebuild.
- **Contained**: Changes are limited to the individual boot image modified (merged-module option).

Description File/Rebuild Advantages

The advantage of the description file/rebuild method is that the changes are permanent and reproducible. Modifications apply to all subsequent module rebuilds and to all merged-module groups built containing the updated module.

Both methods are documented in this section. These procedures are used with the field descriptions starting with the Module Header Fields. For direct modification, use the EditMod LABELS data to navigate the EditMod menus. The DESCRIPTION FILE MACRO data identifies the macro you need to define/modify in the configuration sources to rebuild the PCF device descriptor module.

Direct Modification

Use the EditMod utility and the following procedures to directly modify fields in the existing PCF device descriptor module. The module can stand-alone or it may be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The field references later in this chapter contain a description of each configurable field, its supported values, and the sequence of menu options required by EditMod to modify that field.

Refer to the Utilities Reference for a full description of EditMod's capabilities.
Direct Modification Procedures

To modify the stand-alone module, complete the following steps:

1. Change to the CMDS/BOOTOBS/DESC/<DEVICE> directory (see Figure 6-1).
2. Use EditMod to edit the module:
   
   $EditMod -e <descriptor>

To modify the module as part of a merged module group, complete the following steps:

1. Change to the BOOTS/SYSTEMS/PORTBOOT directory (see Figure 6-2).
2. Use EditMod to edit the module:
   
   $EditMod -e <descriptor> -f=<boot image name>
3. Use the menu selections provided in the EditMod LABELS section of the field reference later in this chapter to locate the fields you want to edit.

4. Select a new value for the field from the AVAILABLE VALUES section of the field reference. Enter that value at the EditMod prompt to modify the field.

5. If you want to make additional modifications, use the p command (previous) to step backward through the EditMod menus. Repeat Steps 3 and 4 until you have made all desired modifications to the descriptor.

6. Select the w command (write) to save the changes.

7. Select the q command (quit) to exit EditMod.

⚠️ Unless you modified the PCF device descriptors in your boot image, you should rebuild your boot image to include the new descriptor.

**Example EditMod Session**

This example modifies a PCF device descriptor as part of the boot image `rom`:

```
$ EditMod -e mhs0 -f=rom
```

1. module header
2. device descriptor data definitions
3. PCF path options
4. PCF logical unit static storage

Which? [?/1-4/p/t/a/w/q] 4

1. interrupt vector : 0x0
2. interrupt level : 0
3. interrupt priority : 5
4. PCF logical unit options

Which? [?/1-4/p/t/a/w/q] 3

interrupt priority : 5
New value: 1

1. interrupt vector : 0x0
2. interrupt level : 0
3. interrupt priority : 1
4. PCF logical unit options

Which? [?/1-19/p/t/a/w/q] w

Which? [?/1-19/p/t/a/w/q] q
Description File Configuration

You can use these procedures to modify the appropriate description file and rebuild the PCF device descriptors for your port directory. The DESCRIPTION FILE MACROS section of the field reference specifies the name of the macro you modify/define in the description files to configure the field. The value used in the define is chosen from the AVAILABLE VALUES specified for the field.

Figure 6-3. Directory Location for Modifying PCF Description Files

Description File Configuration Procedures

1. Change to the PCF/<DEVICE> directory (see Figure 6-3).
2. Edit the file config.des and read the included comments for more information on using the specific description files provided in your software distribution. The config.des file contains a list of macro names that can be defined to override the global default values for the configuration fields.
3. Refer to the DESCRIPTION FILE MACRO section in the field reference later in this chapter to determine the macro name you define to configure the target field.
4. Read the comments in config.des to determine where to place the define for this macro.
5. Select the value you want to use to configure the field. See the AVAILABLE VALUES section of the field reference data for values or macros that can be used for the definition. Define the macro by entering a definition in the appropriate description files as follows:
   
   \#define <macro> <value>

6. Save the changes and rebuild the PCF device descriptors, entering the following command in the PCF/<DEVICE>/DESC directory:
   
   os9make

7. Rebuild your boot image to include the new descriptor.

PCF Device Descriptor Field Reference

This section contains a list of the most commonly configured fields in the PCF device descriptors. Each field entry contains the following information:

- <Field name> - The call name for each field that can be reconfigured in the module.
• **EditMod LABELS** - EditMod menu selections for navigating to the proper field in an EditMod session.

• **DESCRIPTION FILE MACRO** - The macro name you modify/define in the description file.

• **DESCRIPTION** - A brief description of the field’s purpose and use.

• **EXAMPLE** - An optional example of the description file entry showing how to change the value of this field.

• **PORT GENERIC DEFAULT VALUE** - The value set in the port generic description file for this field. This is the value the field is assigned when the module is built, unless the appropriate macro has been defined in the port specific description file to override this default value.

• **PORT SPECIFIC OVERRIDE VALUE** - The value set in the port specific description file for this field. If defined, this is the value the field is assigned when the module is built, overriding the port generic default value.

• **AVAILABLE VALUES** - Values to which the field can be set through EditMod or the description files. In many cases, this data is presented in a table that maps a description of the value to a numeric value appropriate for entry in EditMod, and to a pre-defined macro available for use in the description file.

### Module Header Fields

The following section contains the module header fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>_m_group</td>
<td>MH_GROUP</td>
</tr>
<tr>
<td>_m_user</td>
<td>MH_USER</td>
</tr>
<tr>
<td>mod_name</td>
<td>MH_NAME</td>
</tr>
<tr>
<td>m_access</td>
<td>MH_ACCESS</td>
</tr>
<tr>
<td>m_tylan</td>
<td>MH_TYLAN</td>
</tr>
<tr>
<td>m_attrev</td>
<td>MH_ATTREVS</td>
</tr>
<tr>
<td>m_edit</td>
<td>MH_EDITION</td>
</tr>
</tbody>
</table>
EditMod Labels

- module header
- module owner’s group number

Description

Group ID of the module’s owner. The group number allows people working in the same department or on the same project to share a common identification number.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values

0 to 65535
EditMod Labels
1-module header
2-module owner’s user number

Description
User ID of the module’s owner. The user number identifies a specific user.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
0 to 65535
EditMod Labels
1-module header
3-module name

Description
Contains the module name string.

Port Generic Default Value
String value (None)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels

1-module header
4-access permissions

Description
Defines the permissible module access by its owner or by other users.

Port Generic Default Value
Macro

```
MP_OWNER_READ | MP_OWNER_EXEC | MP_GROUP_READ |
MP_GROUP_EXEC | MP_WORLD_READ | MP_WORLD_EXEC
```

EditMod

```
0x555
```

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
Module access permission values are located in the header file, module.h, and are listed in Table 6-2.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read permission by owner</td>
<td>MP_OWNER_READ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Write permission by owner</td>
<td>MP_OWNER_WRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Execute permission by owner</td>
<td>MP_OWNER_EXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Owner permission mask</td>
<td>MP_OWNER_MASK</td>
<td>0x000f</td>
</tr>
<tr>
<td>Read permission by group</td>
<td>MP_GROUP_READ</td>
<td>0x0010</td>
</tr>
<tr>
<td>Write permission by group</td>
<td>MP_GROUP_WRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Execute permission by group</td>
<td>MP_GROUP_EXEC</td>
<td>0x0040</td>
</tr>
</tbody>
</table>
## Table 6-2. Available Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group permission mask</td>
<td>MP_GROUP_MASK</td>
<td>0x00f0</td>
</tr>
<tr>
<td>Read permission by world</td>
<td>MP_WORLD_READ</td>
<td>0x0100</td>
</tr>
<tr>
<td>Write permission by world</td>
<td>MP_WORLD_WRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Execute permission by world</td>
<td>MP_WORLD_EXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>World permission mask</td>
<td>MP_WORLD_MASK</td>
<td>0x0f00</td>
</tr>
<tr>
<td>All permissions for owner, group, and world</td>
<td>MP_WORLD_ACCESS</td>
<td>0x0777</td>
</tr>
<tr>
<td>System permission mask</td>
<td>MP_SYSTM_MASK</td>
<td>0xf000</td>
</tr>
</tbody>
</table>
EditMod Labels
1-module header
5-type/language

Description
Contains the module’s type (first byte) and language (second byte). The language codes indicate if the module is executable and which language the run-time system requires for execution, if any.

Port Generic Default Value
Macro
\[(MT\_DATA \ll 8) + ML\_OBJECT\]

EditMod
0x401

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
Module type values and language codes are located in the header file, module.h, and are listed in Table 6-3 and Table 6-4.

Table 6-3. m_tylan Available Module Type Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not used (wildcard value in system calls)</td>
<td>MT_ANY</td>
<td>0x0000</td>
</tr>
<tr>
<td>Program module</td>
<td>MT_PROGRAM</td>
<td>0x0001</td>
</tr>
<tr>
<td>Subroutine module</td>
<td>MT_SUBROUT</td>
<td>0x0002</td>
</tr>
<tr>
<td>Multi-module (reserved for future use)</td>
<td>MT_MULTI</td>
<td>0x0003</td>
</tr>
<tr>
<td>Data module</td>
<td>MT_DATA</td>
<td>0x0004</td>
</tr>
<tr>
<td>Configuration data block data module</td>
<td>MT_CDBDATA</td>
<td>0x0005</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0xb-0xa</td>
<td>0xb-0xa</td>
</tr>
</tbody>
</table>
### Table 6-3. `m_tylan` Available Module Type Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>User trap library</td>
<td>MT_TRPLIB</td>
<td>0x000b</td>
</tr>
<tr>
<td>System module</td>
<td>MT_SYSTEM</td>
<td>0x000c</td>
</tr>
<tr>
<td>File manager module</td>
<td>MT_FILEMAN</td>
<td>0x000d</td>
</tr>
<tr>
<td>Physical device driver</td>
<td>MT_DEVDVR</td>
<td>0x000e</td>
</tr>
<tr>
<td>Device descriptor module</td>
<td>MT_DEVDESC</td>
<td>0x000f</td>
</tr>
<tr>
<td>User definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module type mask</td>
<td>MT_MASK</td>
<td>0xff00</td>
</tr>
</tbody>
</table>

### Table 6-4. `m_tylan` Available Language Code Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unspecified language (wildcard in system calls)</td>
<td>ML_ANY</td>
<td>0x0</td>
</tr>
<tr>
<td>Machine language</td>
<td>ML_OBJECT</td>
<td>0x1</td>
</tr>
<tr>
<td>Basic I-code (reserved for future use)</td>
<td>ML_ICODE</td>
<td>0x2</td>
</tr>
<tr>
<td>Pascal P-code (reserved for future use)</td>
<td>ML_PCODE</td>
<td>0x3</td>
</tr>
<tr>
<td>C I-code (reserved for future use)</td>
<td>ML_CCODE</td>
<td>0x4</td>
</tr>
<tr>
<td>Cobol I-code (reserved for future use)</td>
<td>ML_CBLCODE</td>
<td>0x5</td>
</tr>
<tr>
<td>Fortran</td>
<td>ML_FRTNCODE</td>
<td>0x6</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0x7-0xf</td>
<td>0x7-0xf</td>
</tr>
<tr>
<td>User-definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module language mask</td>
<td>ML_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels

1. module header
6. revision/attributes

Description
Contains the module’s attributes (first byte) and revision (second byte).

Port Generic Default Value
Macro
MA_REENT<<8

EditMod
0x8000

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
Module attribute and revision codes are located in the header file module.h, and are listed in Table 6-5.

If two modules with the same name are found in the memory search or are loaded into the current module directory, only the module with the highest revision level is kept. This enables easy substitution of modules for update or correction.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is re-entrant (shareable by multiple tasks).</td>
<td>MA_REENT</td>
<td>0x80 (shifted left to first byte: 0x8000)</td>
</tr>
<tr>
<td></td>
<td>(shifted left to first byte: MA_REENT&lt;&lt;8)</td>
<td></td>
</tr>
<tr>
<td>The module is sticky. A sticky module is not removed from memory until its link count becomes -1 or memory is required for another use.</td>
<td>MA_GHOST</td>
<td>0x40 (shifted left to first byte: 0x4000)</td>
</tr>
<tr>
<td></td>
<td>(shifted left to first byte: MA_GHOST&lt;&lt;8)</td>
<td></td>
</tr>
</tbody>
</table>
### Table 6-5. m_attrev Available Attribute and Revision Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is a system-state module.</td>
<td>MA_SUPER</td>
<td>0x20 (shifted left to first byte: 0x2000)</td>
</tr>
<tr>
<td></td>
<td>(shifted left to first byte: MA_SUPER&lt;&lt;8)</td>
<td></td>
</tr>
<tr>
<td>User-definable revision number</td>
<td>0x0-0xfe</td>
<td>0x0-0xfe</td>
</tr>
<tr>
<td>Module attribute mask</td>
<td>MA_MASK</td>
<td>0xff00</td>
</tr>
<tr>
<td>Module revision mask</td>
<td>MR_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels
1-module header
7-edition

Description
Indicates the software release level for maintenance. OS-9 does not use this field. Whenever a program is revised (even for a small change), increase this number. We recommend internal documentation within the source program be keyed to this system.

Port Generic Default Value
1

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
0 to 65535

Device Descriptor Data Definition Fields
The following section contains the device descriptor data definition fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd_port</td>
<td>PORTADDR</td>
</tr>
<tr>
<td>dd_lun</td>
<td>LUN</td>
</tr>
<tr>
<td>dd_pd_size</td>
<td>PD_SIZE</td>
</tr>
<tr>
<td>dd_type</td>
<td>DD_TYPE</td>
</tr>
<tr>
<td>dd_mode</td>
<td>DD_MODE</td>
</tr>
<tr>
<td>dd_port</td>
<td>MFGR_NAME</td>
</tr>
<tr>
<td>Field</td>
<td>Description File Macro</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------</td>
</tr>
<tr>
<td><code>drvr_name</code></td>
<td>DRVR_NAME</td>
</tr>
<tr>
<td><code>dd_class</code></td>
<td>DD_CLASS</td>
</tr>
</tbody>
</table>
EditMod Labels
2-device descriptor data definitions
1-device port address

Description
Absolute physical address of the hardware controller. This is the address of the device on the bus. This is the lowest address the device has mapped. Port address is hardware dependent.

Macro Example
#define PORTADDR 0xffffe4000

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
0 to 4294967295
Chapter 6: PCF Device Descriptors

dd_lun

LUN

EditMod Labels

2-device descriptor data definitions
2-logical unit number

Description

Distinguishes between the different devices driven from a unique controller. Each unique number represents a different logical unit static storage area.

Macro Example

#define LUN 2

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values

0 to 65535
**dd_pd_size**

<table>
<thead>
<tr>
<th>EditMod Labels</th>
<th>Description</th>
<th>Port Generic Default Value</th>
<th>Port Specific Override Value</th>
<th>Available Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-device descriptor data definitions</td>
<td>Size of the path descriptor. IOMAN uses this value when it allocates a path descriptor.</td>
<td>360</td>
<td>Refer to PCF/&lt;DEVICE&gt;/DESC/config.des (Figure 6-3).</td>
<td>0 to 65535</td>
</tr>
</tbody>
</table>
Chapter 6: PCF Device Descriptors

**dd_type**

**DD_TYPE**

**EditMod Labels**

2-device descriptor data definitions
4-device type

**Description**

Identifies the I/O class of the device.

**Port Generic Default Value**

Macro

```
DT_PCF
```

**EditMod**

```
0xa
```

**Port Specific Override Value**

Refer to `PCF/<DEVICE>/DESC/config.des` (Figure 6-3).

**Available Values**

Device type values are defined in the header file `io.h`, and are listed in Table 6-7.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Character File Type</td>
<td>DT_SCF</td>
<td>0x0</td>
</tr>
<tr>
<td>Random Block File Type</td>
<td>DT_RBF</td>
<td>0x1</td>
</tr>
<tr>
<td>Pipe File Type</td>
<td>DTPIPE</td>
<td>0x2</td>
</tr>
<tr>
<td>Sequential Block File Type</td>
<td>DT_SBF</td>
<td>0x3</td>
</tr>
<tr>
<td>Network File Type</td>
<td>DT_NFM</td>
<td>0x4</td>
</tr>
<tr>
<td>Compact Disc File Type</td>
<td>DT_CDFM</td>
<td>0x5</td>
</tr>
<tr>
<td>User Communication Manager</td>
<td>DT_UCM</td>
<td>0x6</td>
</tr>
<tr>
<td>Socket Communication Manager</td>
<td>DT_SOCK</td>
<td>0x7</td>
</tr>
</tbody>
</table>
### Table 6-7. dd_type Available Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo-Keyboard Manager</td>
<td>DT_PTTY</td>
<td>0x8</td>
</tr>
<tr>
<td>Graphics File Manager</td>
<td>DT_GFM</td>
<td>0x9</td>
</tr>
<tr>
<td>PC-DOS File Manager</td>
<td>DT_PCF</td>
<td>0xa</td>
</tr>
<tr>
<td>Non-volatile RAM File Manager</td>
<td>DT_NRF</td>
<td>0xb</td>
</tr>
<tr>
<td>ISDN File Manager</td>
<td>DT_ISDN</td>
<td>0xc</td>
</tr>
<tr>
<td>MPFM File Manager</td>
<td>DT_MPFM</td>
<td>0xd</td>
</tr>
<tr>
<td>Real-Time Network File Manager</td>
<td>DT_RTNFM</td>
<td>0xe</td>
</tr>
<tr>
<td>Serial Protocol File Manager</td>
<td>DT_SPF</td>
<td>0xf</td>
</tr>
<tr>
<td>Inet File Manager</td>
<td>DT_INET</td>
<td>0xa0</td>
</tr>
<tr>
<td>Reserved for Microware Use Only</td>
<td>17-127</td>
<td>0xa1-0x7f</td>
</tr>
</tbody>
</table>
EditMod Labels
2-device descriptor data definitions
5-device mode capabilities

Description
Used to check the validity of a caller’s access mode byte in I_CREATE or I_OPEN system calls. If a bit is set, the device can perform the corresponding function. The S_ISIZE bit is usually set, because it is handled by the file manager or ignored. If the S_ISHARE bit is set, the device is non-sharable. A printer is an example of a non-sharable device.

Port Generic Default Value
Macro
S_IPRM

EditMod
0xFFFF

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
The file access modes are defined in the header file, modes.h, and located in Table 6-8. The file access permission values are defined in the header file modes.h and in Table 6-9.

Table 6-8. dd_mode Available Values for File Access Modes

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate on open</td>
<td>S_ITRUNC</td>
<td>0x0100</td>
</tr>
<tr>
<td>Ensure contiguous file</td>
<td>S_ICONTIG</td>
<td>0x0400</td>
</tr>
<tr>
<td>Error if file exists on create</td>
<td>S_IEXCL</td>
<td>0x0400</td>
</tr>
<tr>
<td>Create file</td>
<td>S_ICREAT</td>
<td>0x0800</td>
</tr>
</tbody>
</table>
### Table 6-8. `dd_mode` Available Values for File Access Modes (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append to file</td>
<td>S_IAPPEND</td>
<td>0x1000</td>
</tr>
<tr>
<td>Non-sharable</td>
<td>S_ISHARE</td>
<td>0x4000</td>
</tr>
</tbody>
</table>

### Table 6-9. `dd_mode` Available Values for File Access Permissions

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask for permission bits</td>
<td>S_IPRM</td>
<td>0xffffffff</td>
</tr>
<tr>
<td>Owner read</td>
<td>S_IREAD</td>
<td>0x00001</td>
</tr>
<tr>
<td>Owner write</td>
<td>S_IWRITE</td>
<td>0x00002</td>
</tr>
<tr>
<td>Owner execute</td>
<td>S_IEXEC</td>
<td>0x00004</td>
</tr>
<tr>
<td>Search permission</td>
<td>S_ISEARCH</td>
<td>0x00004</td>
</tr>
<tr>
<td>Group read</td>
<td>S_IGREAD</td>
<td>0x00010</td>
</tr>
<tr>
<td>Group write</td>
<td>S_IGWRITE</td>
<td>0x00020</td>
</tr>
<tr>
<td>Group execute</td>
<td>S_IGEXEC</td>
<td>0x00040</td>
</tr>
<tr>
<td>Group search</td>
<td>S_IGSEARCH</td>
<td>0x00040</td>
</tr>
<tr>
<td>Public read</td>
<td>S_IOREAD</td>
<td>0x0100</td>
</tr>
<tr>
<td>Public write</td>
<td>S_IOWRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Public execute</td>
<td>S_IOEXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>Public search</td>
<td>S_IOSEARCH</td>
<td>0x0400</td>
</tr>
</tbody>
</table>
EditMod Labels
2-device descriptor data definitions
6-file manager name

Description
Contains the name string of the file manager module to use.

Port Generic Default Value
“pcf”

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
Any ASCII character string. The string may contain C-style character escapes (such as \n and \012).
EditMod Labels
2-device descriptor data definitions
7-driver name

Description
Contains the name string of the device driver module to use.

Port Generic Default Value
NULL

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
**dd_class**

**DD_CLASS**

**EditMod Labels**
1. module header
2. device descriptor data definitions
3. device class (sequential or random)

**Description**
Used to identify the class of the device, whether it is random or sequential access.

**Port Generic Default Value**
Macro

```
DC_RND
```

**EditMod**

```
0x2
```

**Port Specific Override Value**
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

**Available Values**
Device class available values are defined in the header file, io.h, and in Table 6-10.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential access device</td>
<td>DC_SEQ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Random access device</td>
<td>DC_RND</td>
<td>0x0002</td>
</tr>
</tbody>
</table>

**PCF Path Option Fields**
The following section contains the PCF path option fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.
### Table 6-11. PCF Path Option Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>pd_sid</td>
<td>SIDES</td>
</tr>
<tr>
<td>pd_vfy</td>
<td>VERIFY</td>
</tr>
<tr>
<td>pd_format</td>
<td>FORMAT</td>
</tr>
<tr>
<td>pd_cyl</td>
<td>CYLNDRS</td>
</tr>
<tr>
<td>pd_blk</td>
<td>BLKSTRK</td>
</tr>
<tr>
<td>pd_t0b</td>
<td>BLKSTRK0</td>
</tr>
<tr>
<td>pd_sas</td>
<td>SEGSIZE</td>
</tr>
<tr>
<td>pd_ilv</td>
<td>INTRLV</td>
</tr>
<tr>
<td>pd_toffs</td>
<td>TRKOFFS</td>
</tr>
<tr>
<td>pd_boffs</td>
<td>BLKOFFS</td>
</tr>
<tr>
<td>pd_trys</td>
<td>TRYS</td>
</tr>
<tr>
<td>pd_bsize</td>
<td>BLKSIZE</td>
</tr>
<tr>
<td>pd_cntl</td>
<td>CONTROL</td>
</tr>
<tr>
<td>pd_wpc</td>
<td>PRECOMP</td>
</tr>
<tr>
<td>pd_rwr</td>
<td>REDWRITE</td>
</tr>
<tr>
<td>pd_park</td>
<td>PARK</td>
</tr>
<tr>
<td>pd_lsnoffs</td>
<td>LSNOFFS</td>
</tr>
<tr>
<td>pd_xfersize</td>
<td>XFERSIZE</td>
</tr>
</tbody>
</table>
EditMod Labels

3-PCF path options
1-number of surfaces

Description

Indicates the number of surfaces (heads or sides) for a disk unit.

Port Generic Default Value

2

Port Specific Override Value

Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values

-2147483648 to 2147483647
EditMod Labels

3-PCF path options
2-verify disk writes (0=verify)

Description
Indicates whether a write is verified by a re-read and compare. Write verify operations are generally performed on floppy disks but not hard disks because of the lower soft error rate of hard disks.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
Device verify values are defined in the header file, PCF.h, and in Table 6-12.

Table 6-12. pd_vfy Available Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify disk write</td>
<td>0</td>
<td>0x0</td>
</tr>
<tr>
<td>No verification</td>
<td>1</td>
<td>0x01</td>
</tr>
</tbody>
</table>
**EditMod Labels**

3-PCF path options
3-device format

**Description**

Indicates whether a write is verified by a re-read and compare. Write verify operations are generally performed on floppy disks but not hard disks because of the lower soft error rate of hard disks.

**Port Generic Default Value**

Macro

\[
\text{FMT\_STDFMT + FMT\_DBLBIDTDNS + FMT\_DBLTRKDNS + FMT\_DBLSIDE}
\]

EditMod

0x200e

**Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

**Available Values**

Device format values are defined in the header file, `PCF.h`, and in **Table 6-13**.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track 0 is double density.</td>
<td>FMT_DBLTRK0</td>
<td>0x0001</td>
</tr>
<tr>
<td>Device is double bit density.</td>
<td>FMT_DBLBIDTDNS</td>
<td>0x0002</td>
</tr>
<tr>
<td>Device is double track density.</td>
<td>FMT_DBLTRKDNS</td>
<td>0x0004</td>
</tr>
<tr>
<td>Device is double sided.</td>
<td>FMT_DBLSIDE</td>
<td>0x0008</td>
</tr>
<tr>
<td>Drive is eight inch.</td>
<td>FMT_EIGHTINCH</td>
<td>0x0010</td>
</tr>
<tr>
<td>Drive is five inch.</td>
<td>FMT_FIVEINCH</td>
<td>0x0020</td>
</tr>
<tr>
<td>Drive is three inch.</td>
<td>FMT_THREEINCH</td>
<td>0x0040</td>
</tr>
<tr>
<td>Device is high density.</td>
<td>FMT_HIGHDENS</td>
<td>0x1000</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>Device is standard format.</td>
<td>FMT_STDFMT</td>
<td>0x2000</td>
</tr>
<tr>
<td>Media can be removed.</td>
<td>FMT_REMOVABLE</td>
<td>0x4000</td>
</tr>
<tr>
<td>Device is a hard disk.</td>
<td>FMT_HARDISK</td>
<td>0x8000</td>
</tr>
</tbody>
</table>
EditMod Labels
3-PCF path options
4-number of cylinders

Description
Indicates the number of cylinders per disk.

Port Generic Default Value
80

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
5-default blocks/track

Description
Indicates the number of blocks per track on the disk for all tracks except track 0. (See pd_t0b for track 0 information.)

Port Generic Default Value
16

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
Chapter 6: PCF Device Descriptors

pd_t0b
BLKSTRK0

EditMod Labels
3-PCF path options
6-default blocks/track for trk0

Description
Indicates the number of blocks per track 0 on the disk. Depending on the device, this can be a different number for track 0 than the other tracks on the disk.

Port Generic Default Value
10

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
7-segment allocation size

Description
This value specifies the default minimum number of sectors to be allocated when a file is expanded.

Port Generic Default Value
1

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
**EditMod Labels**

3-PCF path options

8-block interleave offset

**Description**

This value determines the sector interleave factor. Sectors are arranged on a disk in a certain sequential order (1, 2, 3, ... or 1, 3, 5, ...). The interleave factor determines the arrangement. For example, if the interleave factor is 2, the sectors would be arranged by twos, (1,3,5,...) starting at the base sector. (See `pd_boffs` for base sector information.)

**Port Generic Default Value**

3

**Port Specific Override Value**

Refer to `PCF/<DEVICE>/DESC/config.des` (Figure 6-3).

**Available Values**

-2147483648 to 2147483647
EditMod Labels
3-PCF path options
9-track base offset

Description
This is the offset to the first accessible track number. Because Track 0 is often a different density, Track 0 is sometimes not used as the base track.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
10-block base offset

Description
This is the offset to the first accessible sector number. Because Sector 0 is not always the base sector.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
11-# tries

Description
This is the number of times a device tries to access a disk before returning an error.

Port Generic Default Value
7

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
12-size of block in bytes

Description
This is the logical block size in bytes.

Port Generic Default Value
256 (256 characters)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
13-control word

Description
This is the device control word.

Port Generic Default Value
Macro
CTRL_MULTI

EditMod
0x1

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
Control word values are defined in the header file, PCF.h, and in Table 6-14.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable formatting of the device</td>
<td>CTRL_FMTDIS</td>
<td>0x0</td>
</tr>
<tr>
<td>Device is capable of multi-sector transfers</td>
<td>CTRL_MULTI</td>
<td>0x1</td>
</tr>
<tr>
<td>Device size can be obtained from device</td>
<td>CTRL_AUTOSIZE</td>
<td>0x2</td>
</tr>
<tr>
<td>Device requires only one format command</td>
<td>CTRL_FMTENTIRE</td>
<td>0x3</td>
</tr>
<tr>
<td>Device needs a full track buffer for format</td>
<td>CTRL_TRKWRITE</td>
<td>0x4</td>
</tr>
</tbody>
</table>
EditMod Labels
3-PCF path options
14-first write precomp cylinder

Description
This number indicates at which cylinder to begin write precompensation. Only older disk drives require this information, such as MFM or RLL drives.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
15-first reduced write current cylinder

Description
This number indicates at which cylinder to begin reduced write current. Only older disk drives require this information, such as MFM or RLL drives.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
16-park cylinder for hard disks

Description
This is the cylinder where the hard disk heads should be parked when the drive is shut down.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
17- lsn offset for partition

Description
This is the offset to be used when accessing a partitioned drive.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
EditMod Labels
3-PCF path options
max transfer size in terms of bytes

Description
This is the maximum size of memory the controller can transfer at one time. The size is specified in bytes.

Port Generic Default Value
0xff00

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
0 to 4294967295

PCF Logical Unit Static Storage Fields
The following section contains the PCF logical unit static storage fields in the order they appear during an interactive EditMod session. Defined fields may appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>v_vector</td>
<td>VECTOR</td>
</tr>
<tr>
<td>v_irqlevel</td>
<td>IRQLEVEL</td>
</tr>
<tr>
<td>v_priority</td>
<td>PRIORITY</td>
</tr>
</tbody>
</table>
### v_vector

#### EditMod Labels

- 4-PCF logical unit static storage
- 1-interrupt vector

#### Description

This is the vector number of the device interrupt.

#### Port Generic Default Value

80

#### Port Specific Override Value

Refer to `PCF/<DEVICE>/DESC/config.des` (Figure 6-3).

#### Available Values

0 to 255
EditMod Labels
4-PCF logical unit static storage
2-interrupt level

Description
This is the hardware priority of the device interrupt.

Port Generic Default Value
3

Port Specific Override Value
Refer to PCF/\<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-128 to 127
EditMod Labels

4-PCF logical unit static storage
3-interrupt priority

Description
This is the software (polling) priority of the device interrupt.

Port Generic Default Value
10

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-128 to 127

PCF Logical Unit Options
The following section contains the PCF logical unit options fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>lu_stp</td>
<td>STEP</td>
</tr>
<tr>
<td>lu_tfm</td>
<td>DMAMODE</td>
</tr>
<tr>
<td>lu_lun</td>
<td>SCSILUN</td>
</tr>
<tr>
<td>lu_ctrlrid</td>
<td>CTRLRID</td>
</tr>
<tr>
<td>lu_totcyls</td>
<td>TOTCYLS</td>
</tr>
</tbody>
</table>
EditMod Labels
4-PCF logical unit static storage
4-PCF logical unit options
1-step rate

Description
This code sets the head stepping rate used with the drive. Set the step rate to the fastest value the drive is capable of to reduce access time.

Port Generic Default Value
Macro
STEP_30MS

EditMod
0x00

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
Step rate values are defined in the header file, PCF.h, and in Table 6-17.

Table 6-17. lu_stp Available Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 millisecond step rate</td>
<td>STEP_30MS</td>
<td>0x00</td>
</tr>
<tr>
<td>20 millisecond step rate</td>
<td>STEP_20MS</td>
<td>0x01</td>
</tr>
<tr>
<td>12 millisecond step rate</td>
<td>STEP_12MS</td>
<td>0x02</td>
</tr>
<tr>
<td>6 millisecond step rate</td>
<td>STEP_6MS</td>
<td>0x03</td>
</tr>
</tbody>
</table>
EditMod Labels

4-PCF logical unit static storage
4-PCF logical unit options
2-dma transfer mode

Description
This hardware specific byte can be set for use of DMA mode, if it is available. DMA requires only a single interrupt for each block of characters transferred in an I/O operation. It is much faster than methods that interrupt for each character transferred.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-128 to 127
EditMod Labels

4-PCF logical unit static storage
4-PCF logical unit options
3-drive logical unit number

Description

This number is used in the command block to identify the drive to the controller. The driver uses this number when specifying the device.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to \texttt{PCF/<DEVICE>/DESC/config.des} (Figure 6-3).

Available Values

-128 to 127
EditMod Labels
4-PCF logical unit static storage
4-PCF logical unit options
4-controller ID

Description
This is the identification number of the controller attached to the drive. The drive uses this number when communicating with the controller.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-128 to 127
EditMod Labels
4-PCF logical unit static storage
4-PCF logical unit options
5-total number of cylinders

Description
This is the actual number of cylinders on a partitioned drive. The driver uses this value to correctly initialize the drive.

Port Generic Default Value
80

Port Specific Override Value
Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values
-2147483648 to 2147483647
Pipe Device Descriptors

Pipe device descriptors contain configuration data for the pipe pseudo-device used on OS-9. The most common value configured in the pipe device descriptor is the default pipe size.

The next section in this chapter provides a detailed example of the two configuration options you can use to change configuration values in pipe device descriptors.

The rest of this chapter provides a detailed list of all of the pipe device descriptor fields, including field descriptions and available values.

This chapter includes the following topics:

Pipe Device Descriptor Field Configuration Options
Pipe Device Descriptor Field Reference
Module Header Fields
Device Descriptor Data Definition Fields
Pipeman Logical Unit Static Storage
Pipe Device Descriptor Field Configuration Options

To change a pipe device descriptor module configuration field, you can use either of the following methods:

1. Use the `EditMod` utility to directly modify existing pipe device descriptor modules either as a stand-alone module or as part of a merged module group (such as a boot image).
2. Modify the description file for the pipe device descriptor module and rebuild it using the makefile provided.

Direct Modification Advantages

The direct modification method has the following advantages:

- **Fast**  
  No source configuration file rebuilds are necessary.
- **Temporary**  
  The original module or merged-module group configuration can be easily restored through the appropriate rebuild.
- **Contained**  
  Changes are limited to the individual boot image modified (merged-module option).

Description File/Rebuild Advantages

The advantage of the description file/rebuild method is that the changes are permanent and reproducible. Modifications apply to all subsequent module rebuilds and to all merged-module groups built containing the updated module.

Both methods are documented in this section. These procedures are used with the field descriptions starting with the Module Header Fields. For direct modification, use the `EditMod` LABELS data to navigate the `EditMod` menus. The DESCRIPTION FILE MACRO data identifies the macro you need to define/modify in the configuration sources to rebuild the pipe device descriptor module.

Direct Modification

Use the `EditMod` utility and the following procedures to directly modify fields in the existing pipe device descriptor module. The module can stand-alone or it can be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The field references later in this chapter contain a description of each configurable field, its supported values, and the sequence of menu options required by `EditMod` to modify that field.

Refer to the Utilities Reference for a full description of `EditMod`'s capabilities.
Refer to your board guide for information about how to modify the module lists and remake the boot images, and for specified boot image names.

**Direct Modification Procedures**

To modify the stand-alone module, complete the following steps:

1. Change to the `CMDS/BOOTOJBS/DESC/<DEVICE>` directory (see Figure 7-1).

2. Use `EditMod` to edit the module:
   
   `$EditMod -e <descriptor>`

To modify the module as part of a merged module group, complete the following steps:

1. Change to the `BOOTS/SYSTEMS/PORTBOOT` directory (see Figure 7-2).

2. Use `EditMod` to edit the module:
   
   `$EditMod -e <descriptor> -f=<boot image name>`

3. Use the menu selections provided in the `EditMod LABELS` section of the field reference later in this chapter to locate the fields you want to edit.

4. Select a new value for the field from the AVAILABLE VALUES section of the field reference. Enter that value at the `EditMod` prompt to modify the field.
5. If you want to make additional modifications, use the \texttt{p} command (previous) to step backward through the EditMod menus. Repeat Steps 3 and 4 until you have made all desired modifications to the descriptor.

6. Select the \texttt{w} command (write) to save the changes.

7. Select the \texttt{q} command (quit) to exit EditMod.

Unless you modified the pipe device descriptors in your boot image, you should rebuild your boot image to include the new descriptor.

**Example EditMod Session**

This example modifies an pipe device descriptor as part of the boot image \texttt{rom}:

```bash
$ EditMod -e pipe
```

1. module header
2. device descriptor data definitions
3. pipeman logical unit static storage

Which? [?/1-3/p/t/a/w/q] \texttt{3}

1. pipe FIFO buffer size : 0x100

Which? [?/1-6/p/t/a/w/q] \texttt{1}

pipe FIFO buffer size : 0x100
New value: 0x200

1. pipe FIFO buffer size : 0x200

Which? [?/1-19/p/t/a/w/q] \texttt{w}

Which? [?/1-19/p/t/a/w/q] \texttt{q}

**Description File Modification**

You can use these procedures to modify the appropriate description file and rebuild the pipe device descriptors for your port directory. The DESCRIPTION FILE MACROS section of the field reference specifies the name of the macro you modify/define in the description files to configure the field. The value used in the define is chosen from the AVAILABLE VALUES specified for the field.
Description File Modification Procedures

1. Change to the PIPE/<DEVICE> directory (see Figure 7-3).
2. Edit the file config.des and read the included comments for more specific information on using the specific description files provided in your software distribution. The config.des file contains a list of macro names that can be defined to override the global default values for the configuration fields.
3. Refer to the DESCRIPTION FILE MACRO section in the field reference later in this chapter to determine the macro name you define to configure the target field.
4. Read the comments in config.des to determine where to place the define for this macro.
5. Select the value you want to use to configure the field. See the AVAILABLE VALUES section of the field reference data for values or macros that can be used for the definition. Define the macro by entering a definition in the appropriate description files as follows:
   
   `#define <macro> <value>`

6. Save the changes and rebuild the pipe device descriptors, entering the following command in the PIPE/<DEVICE>/DESC directory:
   
   `os9make`

7. Rebuild your boot image to include the new descriptor.

Pipe Device Descriptor Field Reference

This section contains a list of the most commonly configured fields in the pipe device descriptors. Each field entry contains the following information:

- `<Field name>` - The call name for each field that can be reconfigured in the module.
- EditMod LABELS - EditMod menu selections for navigating to the proper field in an EditMod session.
- DESCRIPTION FILE MACRO - The macro name you modify/define in the description file.
- DESCRIPTION - A brief description of the field's purpose and use.
- EXAMPLE - An optional example of the description file entry showing how to change the value of this field.
• PORT GENERIC DEFAULT VALUE - The value set in the port generic description file for this field. This is the value the field is assigned when the module is built, unless the appropriate macro has been defined in the port specific description file to override this default value.

• PORT SPECIFIC OVERRIDE VALUE - The value set in the port specific description file for this field. If defined, this is the value the field is assigned when the module is built, overriding the port generic default value.

• AVAILABLE VALUES - Values to which the field can be set through EditMod or the description files. In many cases, this data is presented in a table that maps a description of the value to a numeric value appropriate for entry in EditMod, and to a pre-defined macro available for use in the description file.

### Module Header Fields

The following section contains the module header fields in the order they appear in the EditMod utility. Defined fields can appear in a different order in the description files.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>_m_group</td>
<td>MH_GROUP</td>
</tr>
<tr>
<td>_m_user</td>
<td>MH_USER</td>
</tr>
<tr>
<td>mod_name</td>
<td>MH_NAME</td>
</tr>
<tr>
<td>m_access</td>
<td>MH_ACCESS</td>
</tr>
<tr>
<td>m_tylan</td>
<td>MH_TYLAN</td>
</tr>
<tr>
<td>m_attrev</td>
<td>MH_ATTREV</td>
</tr>
<tr>
<td>m_edit</td>
<td>MH_EDITION</td>
</tr>
</tbody>
</table>
Chapter 7: Pipe Device Descriptors

EditMod Labels
1-module header
1-module owner’s group number

Description
Group ID of the module’s owner. The group number allows people working in the same department or on the same project to share a common identification number.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
0 to 65535
EditMod Labels
1-module header
2-module owner’s user number

Description
User ID of the module’s owner. The user number identifies a specific user.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
0 to 65535
**EditMod Labels**

1 - module header
2 - module name

**Description**

Contains the module name string.

**Port Generic Default Value**

NULL

**Port Specific Override Value**

Refer to PIPE/config.des (Figure 7-3).

**Available Values**

Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels

1-module header
4-access permissions

Description
Defines the permissible module access by its owner or by other users.

Port Generic Default Value
Macro

MP_OWNER_READ | MP_OWNER_EXEC | MP_GROUP_READ |
MP_GROUP_EXEC | MP_WORLD_READ | MP_WORLD_EXEC

EditMod
0x555

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
Module access permission values are located in the header file, module.h, and are listed in Table 7-2.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read permission by owner</td>
<td>MP_OWNER_READ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Write permission by owner</td>
<td>MP_OWNER_WRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Execute permission by owner</td>
<td>MP_OWNER_EXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Owner permission mask</td>
<td>MP_OWNER_MASK</td>
<td>0x000f</td>
</tr>
<tr>
<td>Read permission by group</td>
<td>MP_GROUP_READ</td>
<td>0x0010</td>
</tr>
<tr>
<td>Write permission by group</td>
<td>MP_GROUP_WRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Execute permission by group</td>
<td>MP_GROUP_EXEC</td>
<td>0x0040</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>--------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Group permission mask</td>
<td>MP_GROUP_MASK</td>
<td>0x00f0</td>
</tr>
<tr>
<td>Read permission by world</td>
<td>MP_WORLD_READ</td>
<td>0x0100</td>
</tr>
<tr>
<td>Write permission by world</td>
<td>MP_WORLD_WRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Execute permission by world</td>
<td>MP_WORLD_EXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>World permission mask</td>
<td>MP_WORLD_MASK</td>
<td>0x0f00</td>
</tr>
<tr>
<td>All permissions for owner, group, and world</td>
<td>MP_WORLD_ACCESS</td>
<td>0x0777</td>
</tr>
<tr>
<td>System permission mask</td>
<td>MP_SYSTM_MASK</td>
<td>0x0000</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
5-type/language

Description
Contains the module’s type (first byte) and language (second byte). The language codes indicate if the module is executable and which language the run-time system requires for execution, if any.

Port Generic Default Value
Macro
\[ (MT\_DATA<<8) + ML\_OBJECT \]

EditMod
0x401

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
Module type values and language codes are located in the header file, module.h, and are listed in Table 7-3 and Table 7-4.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not used ( wildcard value in system calls )</td>
<td>MT_ANY</td>
<td>0x0000</td>
</tr>
<tr>
<td>Program module</td>
<td>MT_PROGRAM</td>
<td>0x0001</td>
</tr>
<tr>
<td>Subroutine module</td>
<td>MT_SUBROUT</td>
<td>0x0002</td>
</tr>
<tr>
<td>Multi-module ( reserved for future use )</td>
<td>MT_MULTI</td>
<td>0x0003</td>
</tr>
<tr>
<td>Data module</td>
<td>MT_DATA</td>
<td>0x0004</td>
</tr>
<tr>
<td>Configuration data block data module</td>
<td>MT_CDBDATA</td>
<td>0x0005</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0xb-0xa</td>
<td>0xb-0xa</td>
</tr>
</tbody>
</table>
### Table 7-3. \texttt{m_tylan} Available Module Type Values (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>User trap library</td>
<td>MT_TRAPLIB</td>
<td>0x000b</td>
</tr>
<tr>
<td>System module</td>
<td>MT_SYSTEM</td>
<td>0x000c</td>
</tr>
<tr>
<td>File manager module</td>
<td>MT_FILEMAN</td>
<td>0x000d</td>
</tr>
<tr>
<td>Physical device driver</td>
<td>MT_DEVDRVR</td>
<td>0x000e</td>
</tr>
<tr>
<td>Device descriptor module</td>
<td>MT_DEVDESC</td>
<td>0x000f</td>
</tr>
<tr>
<td>User definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module type mask</td>
<td>MT_MASK</td>
<td>0xff00</td>
</tr>
</tbody>
</table>

### Table 7-4. \texttt{m_tylan} Available Language Code Values

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unspecified language (wildcard in system calls)</td>
<td>ML_ANY</td>
<td>0x0</td>
</tr>
<tr>
<td>Machine language</td>
<td>ML_OBJECT</td>
<td>0x1</td>
</tr>
<tr>
<td>Basic I-code (reserved for future use)</td>
<td>ML_ICODE</td>
<td>0x2</td>
</tr>
<tr>
<td>Pascal P-code (reserved for future use)</td>
<td>ML_PCODE</td>
<td>0x3</td>
</tr>
<tr>
<td>C I-code (reserved for future use)</td>
<td>ML_CCODE</td>
<td>0x4</td>
</tr>
<tr>
<td>Cobol I-code (reserved for future use)</td>
<td>ML_CBLCODE</td>
<td>0x5</td>
</tr>
<tr>
<td>Fortran</td>
<td>ML_FRTNCODE</td>
<td>0x6</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>0x7-0xf</td>
<td>0x7-0xf</td>
</tr>
<tr>
<td>User-definable</td>
<td>0x10-0xfe</td>
<td>0x10-0xfe</td>
</tr>
<tr>
<td>Module language mask</td>
<td>ML_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
6-revision/attributes

Description
Contains the module’s attributes (first byte) and revision (second byte).

Port Generic Default Value

Macro
MA_REENT<<8

EditMod
0x8000

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values

Module attribute and revision codes are located in the header file module.h., and are listed in Table 7-5.

If two modules with the same name are found in the memory search or are loaded into the current module directory, only the module with the highest revision level is kept. This enables easy substitution of modules for update or correction.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is re-entrant (shareable by multiple tasks).</td>
<td>MA_REENT (shifted left to first byte: MA_REENT&lt;&lt;8)</td>
<td>0x80 (shifted left to first byte: 0x8000)</td>
</tr>
<tr>
<td>The module is sticky. A sticky module is not removed from memory until its link count becomes -1 or memory is required for another use.</td>
<td>MA_GHOST (shifted left to first byte: MA_GHOST&lt;&lt;8)</td>
<td>0x40 (shifted left to first byte: 0x4000)</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>The module is a system-state module.</td>
<td>MA_SUPER</td>
<td>0x20 (shifted left to first byte: 0x2000)</td>
</tr>
<tr>
<td>(shifted left to first byte: MA_SUPER&lt;&lt;8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User-definable revision number</td>
<td>0x0-0xfe</td>
<td>0x0-0xfe</td>
</tr>
<tr>
<td>Module attribute mask</td>
<td>MA_MASK</td>
<td>0xff00</td>
</tr>
<tr>
<td>Module revision mask</td>
<td>MR_MASK</td>
<td>0x00ff</td>
</tr>
</tbody>
</table>
EditMod Labels

1-module header
7-edition

Description
Indicates the software release level for maintenance. OS-9 does not use this field. Whenever a program is revised (even for a small change), increase this number. We recommend internal documentation within the source program be keyed to this system.

Port Generic Default Value
1

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
0 to 65535

Device Descriptor Data Definition Fields

The following section contains the device descriptor data definition fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd_port</td>
<td>PORTADDR</td>
</tr>
<tr>
<td>dd_lun</td>
<td>LUN</td>
</tr>
<tr>
<td>dd_pd_size</td>
<td>PD_SIZE</td>
</tr>
<tr>
<td>dd_type</td>
<td>DD_TYPE</td>
</tr>
<tr>
<td>dd_mode</td>
<td>DD_MODE</td>
</tr>
<tr>
<td>fmgr_name</td>
<td>FMGR_NAME</td>
</tr>
</tbody>
</table>
Table 7-6. Device Descriptor Data Definition Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>drvr_name</code></td>
<td><code>DRVR_NAME</code></td>
</tr>
<tr>
<td><code>dd_class</code></td>
<td><code>DD_CLASS</code></td>
</tr>
</tbody>
</table>
EditMod Labels
2-device descriptor data definitions
1-device port address

Description
Absolute physical address of the hardware controller. This is the address of the device on the bus. This is the lowest address the device has mapped. Port address is hardware dependent.

Macro Example
#define PORTADDR 0xfffe4000

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
0 to 4294967295
EditMod Labels

2-device descriptor data definitions
2-logical unit number

Description

Distinguishes between the different devices driven from a unique controller. Each unique number represents a different logical unit static storage area.

Macro Example

```c
#define LUN 2
```

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to PIPE/config.des (Figure 7-3).

Available Values

0 to 65535
dd_pd_size
PD_SIZE

EditMod Labels
2-device descriptor data definitions
3-path descriptor size

Description
Size of the path descriptor. IOMAN uses this value when it allocates a path descriptor.

Port Generic Default Value
108

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
0 to 65535
Chapter 7: Pipe Device Descriptors

EditMod Labels
2-device descriptor data definitions
4-device type

Description
Identifies the I/O class of the device.

Port Generic Default Value
Macro
DT_PIPE

EditMod
0x2

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
Device type values are defined in the header file io.h, and are listed in Table 7-7.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Character File Type</td>
<td>DT_SCF</td>
<td>0x0</td>
</tr>
<tr>
<td>Random Block File Type</td>
<td>DT_RBF</td>
<td>0x1</td>
</tr>
<tr>
<td>Pipe File Type</td>
<td>DT_PIPE</td>
<td>0x2</td>
</tr>
<tr>
<td>Sequential Block File Type</td>
<td>DT_SBF</td>
<td>0x3</td>
</tr>
<tr>
<td>Network File Type</td>
<td>DT_NFM</td>
<td>0x4</td>
</tr>
<tr>
<td>Compact Disc File Type</td>
<td>DT_CDFM</td>
<td>0x5</td>
</tr>
<tr>
<td>User Communication Manager</td>
<td>DT_UCM</td>
<td>0x6</td>
</tr>
<tr>
<td>Socket Communication Manager</td>
<td>DT_SOCK</td>
<td>0x7</td>
</tr>
<tr>
<td>Description</td>
<td>Macro</td>
<td>EditMod</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Pseudo-Keyboard Manager</td>
<td>DT_PTTY</td>
<td>0x8</td>
</tr>
<tr>
<td>Graphics File Manager</td>
<td>DT_GFM</td>
<td>0x9</td>
</tr>
<tr>
<td>PC-DOS File Manager</td>
<td>DT_PCF</td>
<td>0xa</td>
</tr>
<tr>
<td>Non-volatile RAM File Manager</td>
<td>DT_NRF</td>
<td>0xb</td>
</tr>
<tr>
<td>ISDN File Manager</td>
<td>DT_ISDN</td>
<td>0xc</td>
</tr>
<tr>
<td>MPFM File Manager</td>
<td>DT_MPFM</td>
<td>0xd</td>
</tr>
<tr>
<td>Real-Time Network File Manager</td>
<td>DT_RTNFM</td>
<td>0xe</td>
</tr>
<tr>
<td>Serial Protocol File Manager</td>
<td>DT_SPF</td>
<td>0xf</td>
</tr>
<tr>
<td>Inet File Manager</td>
<td>DT_INET</td>
<td>0xa0</td>
</tr>
<tr>
<td>Reserved for Microware Use Only</td>
<td>17-127</td>
<td>0xa1-0x7f</td>
</tr>
</tbody>
</table>
EditMod Labels

2-device descriptor data definitions  
5-device mode capabilities

Description

Used to check the validity of a caller’s access mode byte in I_CREATE or I_OPEN system calls. If a bit is set, the device can perform the corresponding function. The S_ISIZE bit is usually set, because it is handled by the file manager or ignored. If the S_ISHARE bit is set, the device is non-sharable. A printer is an example of a non-sharable device.

Port Generic Default Value

Macro

\[
S_IREAD \ | \ S_IWRITE
\]

EditMod

0x3

Port Specific Override Value

Refer to PIPE/config.des (Figure 7-3).

Available Values

The file access modes are defined in the header file, modes.h, and located in Table 7-8. The file access permission values are defined in the header file modes.h and in Table 7-9.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate on open</td>
<td>S_ITRUNC</td>
<td>0x0100</td>
</tr>
<tr>
<td>Ensure contiguous file</td>
<td>S_ICONTIG</td>
<td>0x0400</td>
</tr>
<tr>
<td>Error if file exists on create</td>
<td>S_IEXCL</td>
<td>0x0400</td>
</tr>
<tr>
<td>Create file</td>
<td>S_ICREAT</td>
<td>0x0800</td>
</tr>
</tbody>
</table>
### Table 7-8. `dd_mode` Available Values for File Access Modes (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append to file</td>
<td>S_IAPPEND</td>
<td>0x1000</td>
</tr>
<tr>
<td>Non-sharable</td>
<td>S_ISHARE</td>
<td>0x4000</td>
</tr>
</tbody>
</table>

### Table 7-9. `dd_mode` Available Values for File Access Permissions

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask for permission bits</td>
<td>S_IPRM</td>
<td>0xffff</td>
</tr>
<tr>
<td>Owner read</td>
<td>S_IREAD</td>
<td>0x0001</td>
</tr>
<tr>
<td>Owner write</td>
<td>S_IWRITE</td>
<td>0x0002</td>
</tr>
<tr>
<td>Owner execute</td>
<td>S_IEXEC</td>
<td>0x0004</td>
</tr>
<tr>
<td>Search permission</td>
<td>S_ISEARCH</td>
<td>0x0004</td>
</tr>
<tr>
<td>Group read</td>
<td>S_IGREAD</td>
<td>0x0010</td>
</tr>
<tr>
<td>Group write</td>
<td>S_IGWRITE</td>
<td>0x0020</td>
</tr>
<tr>
<td>Group execute</td>
<td>S_IGEXEC</td>
<td>0x0040</td>
</tr>
<tr>
<td>Group search</td>
<td>S_IGSEARCH</td>
<td>0x0040</td>
</tr>
<tr>
<td>Public read</td>
<td>S_IOREAD</td>
<td>0x0100</td>
</tr>
<tr>
<td>Public write</td>
<td>S_IOWRITE</td>
<td>0x0200</td>
</tr>
<tr>
<td>Public execute</td>
<td>S_IOEXEC</td>
<td>0x0400</td>
</tr>
<tr>
<td>Public search</td>
<td>S_IOSEARCH</td>
<td>0x0400</td>
</tr>
</tbody>
</table>
EditMod Labels
1-module header
2-device descriptor data definitions
6-file manager name

Description
Contains the name string of the file manager module to use.

Port Generic Default Value
"pipe"

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
EditMod Labels
1-module header
2-device descriptor data definitions
7-driver name

Description
Contains the name string of the device driver module to use.

Port Generic Default Value
0 (zero)

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
Any ASCII character string. The string can contain C-style character escapes (such as \n and \012).
Chapter 7: Pipe Device Descriptors

dd_class
DD_CLASS

EditMod Labels
2-device descriptor data definitions
8-device class (sequential or random)

Description
Used to identify the class of the device, whether it is random or sequential access.

Port Generic Default Value
Macro
DC_SEQ

EditMod
0x1

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
Device class available values are defined in the header file, io.h, and in Table 7-10.

<table>
<thead>
<tr>
<th>Description</th>
<th>Macro</th>
<th>EditMod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential access device</td>
<td>DC_SEQ</td>
<td>0x0001</td>
</tr>
<tr>
<td>Random access device</td>
<td>DC_RND</td>
<td>0x0002</td>
</tr>
</tbody>
</table>

Pipeman Logical Unit Static Storage

The following section contains the Pipeman logical unit static storage fields in the order they appear during an interactive EditMod session. Defined fields can appear in a different order in config.des.
Table 7-11. Pipeman Logical Unit Static Storage Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description File Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>bufsz</td>
<td>BUFSZ</td>
</tr>
</tbody>
</table>
EditMod Labels
3-pipeman logical unit static storage
1-pipe FIFO buffer size

Description
Used to define the buffer size of the pipe.

Port Generic Default Value
256

Port Specific Override Value
Refer to PIPE/config.des (Figure 7-3).

Available Values
0 to 4294967295
Index

A
access
  changing for init 140
access permissions
  setting for cnfgdata 25
  for init 97
  for PCF 460
  for pipe 516
  for RBF 404
  for SBF 366
  for SCF 159
acct_name
  changing for init 118
attributes
  of module
    setting for cnfgdata 29
    setting for init 101
    setting for PCF 464
    setting for pipe 520
    setting for RBF 408
    setting for SBF 370
    setting for SCF 163
autoboot_delay
  changing for cnfgdata 85
AUTOECHO
  changing for SCF 350
AUTOLF
  changing for SCF 351

B
B_NVRAM 140
B_PARITY 140
B_ROM 140
B_SHARED 140
B_USERRAM 140
back space character
  changing 348
baud rate
  changing for SCF device 198, 200
BAUDRATE
  changing for SCF 198
  bell character
    changing 345
BELLCH
  changing for SCF 345
  bits per character,
    changing the 202
  blk_beg
    changing for init 146
  blk_end
    changing for init 147
BLKOFFS
  changing
    for PCF 489
    for RBF 433
blksz
  changing for init 141
BLKSIZE
  changing 335, 491
    for SBF 385
BLKSTRK
  changing
    for PCF 484
    for RBF 428
BLKSTRK0
  changing
    for PCF 485
    for RBF 429
block offset
  changing 433, 434, 435, 436, 489, 490, 491, 492
block size 385
blocks per track
  number of on disk
    changing 428, 484
blocks per track 0
  number of on disk
    changing 429, 485
boot data
  boot_abname
    configuration 80
  boot_automenu

537
configuration 83
  boot_delay
  configuration 85
Boot_newab
  configuration 81
boot_newname
  configuration 82
boot_params
  configuration 84
boot_abname
  boot data
  configuration 80
changing for cnfgdata 80
boot_automenu
  boot data
  configuration 83
changing for cnfgdata 83
BOOT_CMDSIZE
  changing 79
boot_cmdsize
  changing for cnfgdata 79
BOOT_COUNT
  changing 78
boot_count
  changing for cnfgdata 78
boot_delay
  boot data
  configuration 85
boot_newab
  boot data
  configuration 81
changing for cnfgdata 81
boot_newname
  boot data
  configuration 82
changing for cnfgdata 82
boot_params
  boot data
  configuration 84
changing for cnfgdata 84
brdcst_address
  changing for cnfgdata 67
interface data
  configuration 67
BSB
changing
got SCF 348
BSPCH
  changing for SCF 346
buffer size
  changing for pipe 535
BUFSZ
  changing for pipe 535
bufsz
  changing for pipe 535
C
C I-code (reserved)
  module header
    language code 28, 100
    language code for PCF 463
    language code for pipe 519
    language code for RBF 407
    language code for SBF 369
    language code for SCF 162
chd utility 112
chx utility 112
cinit
  changing
    m_attrev (attributes/revision) field 101
cnfgdata
  changing
    autoboot_delay field 85
    boot_abname field 80
    boot_automenu field 83
    boot_cmdsize field 79
    boot_count field 78
    boot_newab field 81
    boot_newname field 82
    boot_params field 84
    brdcst_address field 67
communication device cons_baudrate field 51
communication device cons_flow field 55
communication device cons_level field 48
communication device cons_parity field 50
communication device cons_priority field 47
communication device cons_stopbits field 54
communication device cons_timeout field 49
communication device cons_vector field 46
communication device cons_wordsize field 53
cons_name field 33
console device cons_baudrate field 39
console device cons_flow field 43
console device cons_level field 36
console device cons_parity field 38
console device cons_priority field 35
console device cons_stopbits field 42
console device cons_timeout field 37
console device cons_vector field 34
console device cons_wordsize field 41
debug_call_at_cold field 58
debugger_name field 57
gw_address field 68
hwtype field 70
if_flags field 71
if_level field 76
if_name field 72
if_priority field 75
if_vector field 74
ip_address field 65
lpm_count field 63
m_access field 25
m_attrev (attributes/revision) field 29
m_edit field 31
m_group field 22
m_tylan (type/language) field 27
m_user field 23
mac_Address field 69
max_notifiers field 86
maxlpmconns field 62
maxlpmprotos field 60
maxrcmbuffs field 61
port_address field 73
subnet_mask field 66
changing mod_name field 24
MH_ACCESS
  changing 25
MH_EDITION
  changing 31
MH_GROUP
  changing 22
MH_NAME
  changing 24
MH_TYLAN
  changing 27
MH_USER
  changing 23
module header
group ID 22
user ID
  module header 23
COBOL I-code (reserved)
module header
  language code 28, 100
  language code for PCF 463
  language code for pipe 519
  language code for RBF 407
  language code for SBF 369
  language code for SCF 162
COMM_BAUDRATE
  changing 51
COMM_FLOW
  changing 55
COMM_PARITY
  changing 50
COMM_PRIORITY
  changing 47
COMM_STOPBITS
  changing 47
COMM_TIMEOUT
  changing 49
COMM_VECTOR
  changing 46
COMM_WORDSIZE
  changing 53
COMPAT
  changing
    for init 134
configuration data block
  module header
    type code for cnfgdata 27
    type code for configdata 99
    type code for PCF 462
    type code for pipe 518
    type code for RBF 406
    type code for SBF 368
    type code for SCF 161
configuration module 92
CONS_BAUDRATE
  changing 39
  cons_baudrate
    changing
      for cnfgdata communication device 51
      for cnfgdata console device 39
CONS_FLOW
  changing 43
  cons_flow
    changing
      for cnfgdata communication device 55
      for cnfgdata console device 43
CONS_LEVEL
  changing 36
  cons_level
    changing
      for cnfgdata communication device 48
      for cnfgdata console device 36
CONS_NAME
  changing 33
  for init 113
CONS_PARITY
changing 38
cons_parity
changing
for cnfgdata communication device 50
for cnfgdata console device 38
CONS_PRIORITY
changing 35
cons_priority
changing
for cnfgdata communication device 47
for cnfgdata console device 35
CONS_STOPBITS
changing 42
cons_stopbits
changing
for cnfgdata communication device 54
for cnfgdata console device 42
CONS_TIMEOUT
changing 37
cons_timeout
changing
for cnfgdata communication device 49
for cnfgdata console device 37
CONS_VECTOR
changing 34
cons_vector
changing
for cnfgdata communication device 46
for cnfgdata console device 34
CONS_WORDSIZE
changing 41
cons_wordsize
changing
for cnfgdata communication device 53
for cnfgdata console device 41
console device
setting vector number 34, 46
console device name
setting 33
console_name
changing
for cnfgdata 33
for init 113
CONTROL
changing 436, 492
controller ID number
setting 448, 504, 505
CPUCOMPAT
changing
for init 128
CTRLRID
changing 448, 504
for PCF 505
cylinder
starting reduced write
changing 438, 494
cylinders
number of disk
changing 427, 483
CYLNDRS
changing 427, 483
D
data module
module header
type code for cnfgdata 27
type code for init 99
type code for PCF 462
type code for pipe 518
type code for RBF 406
type code for SBF 368
type code for SCF 161
DC_RND
dd_class
available value 176, 383, 421, 477, 533
DC_SEQ
dd_class
available value 176, 383, 421, 477, 533
DD_CLASS
changing
for PCF 477
for pipe 533
for RBF 421
for SBF 383
for SCF 176
dd_class
available value
DC_RND 176, 383, 421, 477, 533
DC_SEQ 176, 383, 421, 477, 533
changing
for PCF 477
for pipe 533
for RBF 421
for SBF 383
for SCF 176
dd_lun
changing
for PCF 469
for pipe 525
for RBF 413
for SBF 375
for SCF 168
DD_MODE
changing
for PCF 473
for pipe 529
for RBF 417
for SBF 379
for SCF 172

```
for pipe 527
for RBF 415
for SBF 377
for SCF 170
```

`dd_mode`

available value
- `S_IAPPEND` 173, 380, 418, 474, 530
- `S_ICONTIG` 172, 379, 417, 473, 529
- `S_ICREATE` 172, 379, 417, 473, 529
- `S_IEXCL` 172, 379, 417, 473, 529
- `S_IEXEC` 173, 380, 418, 474, 530
- `S_IWRITE` 173, 380, 418, 474, 530
- `S_IREAD` 173, 380, 418, 474, 530
- `S_ISEARCH` 173, 380, 418, 474, 530
- `S_IEXEC` 173, 380, 418, 474, 530
- `S_IWRITE` 173, 380, 418, 474, 530

changing
- for PCF 473
- for pipe 529
- for RBF 417
- for SBF 379
- for SCF 172

```
for pipe 526
for RBF 414
for SBF 376
for SCF 169
```

`dd_pd_size`

changing
- for PCF 470
- for pipe 526
- for RBF 414
- for SBF 376
- for SCF 169

```
for pipe 524
for RBF 412
for SBF 374
for SCF 167
```

`dd_port`

changing
- for PCF 468
- for pipe 524
- for RBF 412
- for SBF 374
- for SCF 167

```
for pipe 527
for RBF 415
for SBF 377
for SCF 170
```

`DD_TYPE`

changing
- for PCF 471
- for pipe 527
- for RBF 415
- for SBF 377
- for SCF 170

```
for pipe 527
for RBF 415
for SBF 377
for SCF 170
```

`dd_type`

available values
- for RBF 417
- for SBF 379
- for SCF 172

```
for RBF 415
for SBF 377
for SCF 170
```

`debug_call_at_cold`

changing
- for PCF 471
- for pipe 527
- for RBF 415
- for SBF 377
- for SCF 170

```
for pipe 527
for RBF 415
for SBF 377
for SCF 170
```

`debug_name`

changing
- for cnfgdata 58

```
for cnfgdata 57
```

`DBUS_RBF_8FLG`

changing
- for SBF 380
- for pipe 524
- for RBF 412
- for SBF 374
- for SCF 167

```
for pipe 524
for RBF 412
for SBF 374
for SCF 167
```

`device`

identify unique
- for controller 168, 375, 413, 469, 525

Device Controller SCSI ID
changing 388

device descriptor

module header
type code for cnfgdata 28
type code for init 100
type code for PCF 463
type code for pipe 519
type code for RBF 407
type code for SBF 369
type code for SCF 162
device driver
module header
type code for cnfgdata 28
type code for init 100
type code for PCF 463
type code for pipe 519
type code for RBF 407
type code for SBF 369
type code for SCF 162

name string
selecting for PCF 476
selecting for pipe 532
selecting for RBF 420
selecting for SBF 382
selecting for SCF 175
port address
   for hardware controller 167, 374, 412, 468, 524
device interrupt
   changing 391, 443, 499
device mode
   I/O class of 172, 379, 417, 473, 529
device state
   changing 393
device type
   I/O class of 170, 377, 415, 471, 527
DFLG_DRIVEBUSY 393
DFLG_OFFLAG 393
DFLG_READFLAG 393
DFLG_WRITEFLAG 393
disk cylinders
   changing 427, 483
disk format
   changing 425, 481
disk sides
   changing 423, 479
disk type
   changing 425, 481
disk write verification
   changing 424, 480
DMA mode
   changing 387
DMA transfer mode
   setting 446, 502
dma_addr
   changing
      for init 145
DMAMODE
   changing
      for PCF 502
      for RBF 446
      for SBF 387
drive unit number
   setting 447, 503
DRIVE_FLAG
   changing 393
drive_name
   changing
      for init 112
Driver Compatibility Flags
   changing 386
DRVR_NAME
   changing
      for PCF 476
      for pipe 532
      for RBF 420
      for SBF 382
EOFCH  changing  for SCF  343
EOLNULLS  changing  for SCF  354
EOR character  changing  340
EORCH  changing  for SCF  340
EVENTS  changing  for init  121
executable
module header
type code for cnfgdata  27
type code for init  99
type code for PCF  462
type code for pipe  518
type code for RBF  406
type code for SBF  368
type code for SCF  161
extens_list
changing  for init  114
EXTENSIONS  changing  for init  114
F
file
sysboot  92
file manager
module header
type code for cnfgdata  28
type code for init  100
type code for PCF  463
type code for pipe  519
type code for RBF  407
type code for SBF  369
type code for SCF  162
name string
selecting for PCF  475
selecting for pipe  531
selecting for RBF  419
selecting for SBF  381
selecting for SCF  174
FLAGS  changing  386
FMGR_NAME  changing  for PCF  475
FUNC0x07
  changing for SCF 241
FUNC0x08
  changing for SCF 245
FUNC0x09
  changing for SCF 249
FUNC0x0a
  changing for SCF 253
FUNC0x0b
  changing for SCF 257
FUNC0x0c
  changing for SCF 261
FUNC0x0d
  changing for SCF 265
FUNC0x0e
  changing for SCF 269
FUNC0x0f
  changing for SCF 273
FUNC0x10
  changing for SCF 277
FUNC0x11
  changing for SCF 281
FUNC0x12
  changing for SCF 285
FUNC0x13
  changing for SCF 289
FUNC0x14
  changing for SCF 293
FUNC0x15
  changing for SCF 297
FUNC0x16
  changing for SCF 301
FUNC0x17
  changing for SCF 305
FUNC0x18
  changing

G
  ghost module
    setting value for cnfgdata 29
    setting value for init 101
    setting value for PCF 464
    setting value for pipe 520
    setting value for RBF 408
    setting value for SBF 370
    setting value for SCF 163
Greenwich Mean Time (GMT) 129
  group ID
    module header
      cnfgdata 22
      init 94
      PCF 457
      pipe 513
      RBF 401
      SBF 363
      SCF 156
  gw_address
    changing
      for cnfgdata 68
  interface data

H
  hardware controller
    absolute physical address
for dd_port 167, 374, 412, 468, 524
hardware_vector
    changing
        for SCF 180
head step rate
    changing 445, 501
header files
    io.h
        for available device types 170, 377, 415, 471, 527
    modes.h
        setting dd_mode 172, 379, 417, 473, 529
hilim
    changing
        for init 143
hwtype
    changing
        for cnfgdata 70
interface data
    configuration 70
I
I/O class of
    device mode
        changing 172, 379, 417, 473, 529
    device type
        changing 170, 377, 415, 471, 527
I_CREATE
    setting access mode 172, 379, 417, 473, 529
I_OPEN
    setting access mode 172, 379, 417, 473, 529
if_flags
    changing
        for cnfgdata 71
        interface data
            configuration 71
if_level
    changing
        for cnfgdata 76
        interface data
            configuration 76
if_name
    changing
        for cnfgdata 72
if_name_ether
    interface data
        configuration 72
if_name_slip
    interface data
        configuration 72
if_priority
    changing
        for cnfgdata 75
interface data
    configuration 75
if_vector
    changing
        for cnfgdata 74
        interface data
            configuration 74
init
    changing
        access field 140
        acct_name field 118
        blk_beg field 146
        blk_end field 147
        blksz field 141
        COMPAT 134
        CONS_NAME 113
        console_name field 113
        CPUCOMPAT 128
        desc field 144
        dma_addr field 145
        drive_name field 112
        DSPTBLSZ 127
        EVENTS 121
        extens_list field 114
        EXTENSIONS 114
        hilim field 143
        install_name field 108
        ioman_name field 117
        lolim field 142
        m_access field 97
        m_compat field 134
        m_cpucompat field 128
        m_cputyp field 107
        m_dsptbl field 127
        m_edit field 103
        m_edition field 133
        m_events field 121
        m_group field 94
        m_level field 130
        m_major field 131
        m_maxage field 126
        m_maxsigs field 135
        m_minor field 132
        m_minpty field 125
        m_paths field 120
        m_procs field 119
        m_site field 106
        m_slice field 123
        m_syspri field 124
        m_tylan (type/language) field 99
        m_user field 95
        MAXPTY 126
**OS-9® Device Descriptor and Configuration Module Reference**

N
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXSIGS</td>
<td>135</td>
</tr>
<tr>
<td>MINTPTY</td>
<td>125</td>
</tr>
<tr>
<td>MPUCHIP</td>
<td>107</td>
</tr>
<tr>
<td>OS_EDITION</td>
<td>133</td>
</tr>
<tr>
<td>OS_LEVEL</td>
<td>130</td>
</tr>
<tr>
<td>OS_REVISION</td>
<td>132</td>
</tr>
<tr>
<td>OS_VERSION</td>
<td>131</td>
</tr>
<tr>
<td>OS9K_REVSTR</td>
<td>109</td>
</tr>
<tr>
<td>os9rev_name</td>
<td>109</td>
</tr>
<tr>
<td>PATHS</td>
<td>120</td>
</tr>
<tr>
<td>preio_name</td>
<td>136</td>
</tr>
<tr>
<td>PREIOS</td>
<td>136</td>
</tr>
<tr>
<td>prior</td>
<td>139</td>
</tr>
<tr>
<td>PROC5</td>
<td>119</td>
</tr>
<tr>
<td>RTC_NAME</td>
<td>116</td>
</tr>
<tr>
<td>rtc_name</td>
<td>116</td>
</tr>
<tr>
<td>SITE</td>
<td>106</td>
</tr>
<tr>
<td>SLICE</td>
<td>123</td>
</tr>
<tr>
<td>sparam_string</td>
<td>111</td>
</tr>
<tr>
<td>SYS_DEVICE</td>
<td>112</td>
</tr>
<tr>
<td>SYS_PARAMS</td>
<td>111</td>
</tr>
<tr>
<td>SYS_PRIOR</td>
<td>124</td>
</tr>
<tr>
<td>SYS_START</td>
<td>110</td>
</tr>
<tr>
<td>SYS_TMZONE</td>
<td>129</td>
</tr>
<tr>
<td>sysgo_name</td>
<td>110</td>
</tr>
<tr>
<td>TICK_NAME</td>
<td>115</td>
</tr>
<tr>
<td>TICK_SEC</td>
<td>122</td>
</tr>
<tr>
<td>ticker_name</td>
<td>115</td>
</tr>
<tr>
<td>ticsec</td>
<td>122</td>
</tr>
<tr>
<td>type</td>
<td>138</td>
</tr>
<tr>
<td>USRACCT_NAME</td>
<td>118</td>
</tr>
<tr>
<td>changing IOMAN_NAME</td>
<td>117</td>
</tr>
<tr>
<td>changing mod_name</td>
<td>96</td>
</tr>
<tr>
<td>INSTALNAME</td>
<td>108</td>
</tr>
<tr>
<td>changing</td>
<td>108</td>
</tr>
<tr>
<td>MH_ACCESS</td>
<td>97</td>
</tr>
<tr>
<td>changing</td>
<td>97</td>
</tr>
<tr>
<td>MH_EDITION</td>
<td>103</td>
</tr>
<tr>
<td>changing</td>
<td>103</td>
</tr>
<tr>
<td>MH_GROUP</td>
<td>94</td>
</tr>
<tr>
<td>changing</td>
<td>94</td>
</tr>
<tr>
<td>MH_NAME</td>
<td>96</td>
</tr>
<tr>
<td>changing</td>
<td>96</td>
</tr>
<tr>
<td>MH_TYLAN</td>
<td>99</td>
</tr>
<tr>
<td>changing</td>
<td>99</td>
</tr>
<tr>
<td>MH_USER</td>
<td>95</td>
</tr>
<tr>
<td>changing</td>
<td>95</td>
</tr>
<tr>
<td>module header</td>
<td></td>
</tr>
<tr>
<td>group ID</td>
<td>94</td>
</tr>
<tr>
<td>user ID</td>
<td></td>
</tr>
<tr>
<td>changing</td>
<td>95</td>
</tr>
</tbody>
</table>

**INPUT_TYPE**
changing for SCF 183
insert mode character changing 353
INSERTMODE changing for SCF 353
INSIZE changing for SCF 188
install_name changing for init 108
INSTALNAME changing for init 108
interface data
   brdct_address
   configuration 67
gw_address
   configuration 68
hwtype
   configuration 70
   if_flags
   configuration 71
   if_level
   configuration 76
   if_name_ether
   configuration 72
   if_name_slip
   configuration 72
   if_priority
   configuration 75
   if_vector
   configuration 74
   ip_address
   configuration 65
   mac_address
   configuration 69
   port_address
   configuration 73
   subnet_mask
   configuration 66
interleave factor changing 431, 487
intermediate code module header
   language code for cnfgdata 28
   language code for init 100
   language code for PCF 463
   language code for pipe 519
   language code for RBF 407

Init module 92
init.h 92
language code for SBF 369
language code for SCF 162
interrupt control key
  changing for keyboard 191
interrupt levels
  supported number
  changing 181
interrupt vector
  setting
  for console device 34, 46
INTRLV
  changing
  for PCF 487
  for RBF 431
io.h
  available device types 170, 377, 415, 471, 527
IOMAN_NAME
  changing
  for init 117
iomant_name
  changing
  for init 117
ip_address
  changing
  for cnfgdata 65
  interface data
  configuration 65
IRQ_MASK
  changing
  for SCF 186
IRQLEVEL
  changing 391, 443, 499
  for SCF 181

K
keyboard interrupt
  changing control key 191
keyboard pause
  changing control key 195
keyboard quit
  changing control key 194
keyboard X-OFF
  changing control key 197
keyboard X-ON
  changing control key 196
KYBDINTR
  changing
  for SCF 191
KYBPDPAUSE
  changing
  for SCF 195
KYBDQUIT
  changing
  for SCF 191

L
language
  required for running
    setting for cnfgdata 27
    setting for init 99
    setting for PCF 462
    setting for pipe 518
    setting for RBF 406
    setting for SBF 368
    setting for SCF 161
language code
  available values
    MT_CBLCODE 28, 100
    MT_CCODE 28, 100
    MT_FRTNCODE 28, 100
    MT_MASK 28, 100
  available values for cnfgdata
    MT_ANY 28
    MT_ICODE 28
    MT_OBJECT 28
    MT_PCODE 28
  available values for init
    MT_ANY 100
    MT_ICODE 100
    MT_OBJECT 100
    MT_PCODE 100
  available values for PCF
    MT_ANY 463
    MT_CBLCODE 463
    MT_CCODE 463
    MT_FRTNCODE 463
    MT_ICODE 463
    MT_MASK 463
    MT_OBJECT 463
    MT_PCODE 463
  available values for pipe
    MT_ANY 519
    MT_CBLCODE 519
    MT_CCODE 519
    MT_FRTNCODE 519
    MT_ICODE 519
    MT_MASK 519
    MT_OBJECT 519
    MT_PCODE 519
  available values for RBF
    MT_ANY 407
    MT_CBLCODE 407
    MT_CCODE 407
    MT_FRTNCODE 407
    MT_ICODE 407
    MT_MASK 407
MT_OBJECT 407
MT_PCODE 407
available values for SBF
MT_ANY 369
MT_CBLCODE 369
MT_CCODE 369
MT_FRTNCODE 369
MT_ICODE 369
MT_MASK 369
MT_OBJECT 369
MT_PCODE 369
available values for SCF
ML_ANY 162
ML_CBLCODE 162
ML_CCODE 162
ML_FRTNCODE 162
ML_ICODE 162
ML_MASK 162
ML_OBJECT 162
ML_PCODE 162
line delete
changing 349
line feed character
changing 351, 352
LINEDEL
changing for SCF 349
lines per page
setting 190
lines per screen
setting 190
LLPM_COUNT
changing 63
llpm_count
changing for cnfgdata 63
LLPM_MAXCONNS
changing 62
LLPM_MAXPROTOS
changing 60
LLPM_MAXRCVMBUFS
changing 61
logical sector offset
changing 440, 496
logical unit static storage
changing 168, 375, 413, 469, 525
device identifier 168, 375, 413, 469, 525
lolim
changing for init 142
LSNOFFS
changing for PCF 496
lu_ctrlrid
changing for RBF 440
for PCF 504
for RBF 448
PCF 505
lu_lun
changing for PCF 503
for RBF 447
lu_stp
changing for PCF 501
for RBF 445
lu_tfm
changing for PCF 502
for RBF 446
lu_totcyls
changing for PCF 469
for pipe 525
for RBF 413
for SBF 375
for SCF 168, 185
LUPARITY
changing for SCF 200
M
m_access
changing for cnfgdata 25
for init 97
for PCF 460
for pipe 516
for RBF 404
for SBF 366
for SCF 159
m_attrev (attributes/revision)
changing for cnfgdata 29
for init 101
for PCF 464
for pipe 520
for RBF 408
for SBF 370
for SCF 163
m_compat
changing
m_cpucompat
  changing
  init 128
m_cputyp
  changing
  for init 107
m_dsptbl
  changing
  for init 127
m_edit
  changing
  for cnfgdata 31
  for init 103
  for PCF 466
  for pipe 522
  for RBF 410
  for SBF 372
  for SCF 165
m_edition
  changing
  for init 133
m_events
  changing
  for init 121
m_exec 92
m_group
  changing
  for cnfgdata 22
  for init 94
  for PCF 457
  for pipe 513
  for RBF 401
  for SBF 363
  for SCF 156
m_level
  changing
  for init 130
m_major
  changing
  for init 131
m_maxage
  changing
  for init 126
m_maxsigs
  changing
  for init 135
m_minor
  changing
  for init 132
m_minpty
  changing
  for init 125
m_paths
  changing
  for init 120
m_procs
  changing
  for init 119
m_site
  changing
  for init 106
m_slice
  changing
  init 123
m_syspri
  changing
  for init 124
m_ticsec
  changing
  for init 122
m_tmzone
  changing
  for init 129
m_tylan
  available values for cnfgdata
    MT_ANY 27
    MT_CDBDATA 27
    MT_DATA 27
    MT_DEVDESC 28
    MT_DEVDRVR 28
    MT_FILEMAN 28
    MT_MASK 28
    MT_MULTI (reserved) 27
    MT_PROGRAM 27
    MT_SUBROUT 27
    MT_SYSTEM 28
    MT_TRAPLIB 27
  available values for init
    MT_ANY 99
    MT_CDBDATA 99
    MT_DATA 99
    MT_DEVDESC 100
    MT_DEVDRVR 100
    MT_FILEMAN 100
    MT_MASK 100
    MT_MULTI (reserved) 99
    MT_PROGRAM 99
    MT_SUBROUT 99
    MT_SYSTEM 100
    MT_TRAPLIB 100
  available values for PCF
    MT_ANY 462
    MT_CDBDATA 462
    MT_DATA 462
    MT_DEVDESC 463
| MT_DEVDRVR | 463 |
| MT_FILEMAN | 463 |
| MT_MASK | 463 |
| MT_MULTI (reserved) | 462 |
| MT_PROGRAM | 462 |
| MT_SUBROUT | 462 |
| MT_SYSTEM | 463 |
| MT_TRAPLIB | 463 |

available values for pipe
| MT_ANY | 518 |
| MT_CDBDATA | 518 |
| MT_DATA | 518 |
| MT_DEVDESC | 519 |
| MT_DEVDRVR | 519 |
| MT_FILEMAN | 519 |
| MT_MASK | 519 |
| MT_MULTI (reserved) | 518 |
| MT_PROGRAM | 518 |
| MT_SUBROUT | 518 |
| MT_SYSTEM | 519 |
| MT_TRAPLIB | 519 |

available values for RBF
| MT_ANY | 406 |
| MT_CDBDATA | 406 |
| MT_DATA | 406 |
| MT_DEVDESC | 407 |
| MT_DEVDRVR | 407 |
| MT_FILEMAN | 407 |
| MT_MASK | 407 |
| MT_MULTI (reserved) | 406 |
| MT_PROGRAM | 406 |
| MT_SUBROUT | 406 |
| MT_SYSTEM | 407 |
| MT_TRAPLIB | 407 |

available values for SBF
| MT_ANY | 368 |
| MT_CDBDATA | 368 |
| MT_DATA | 368 |
| MT_DEVDESC | 369 |
| MT_DEVDRVR | 369 |
| MT_FILEMAN | 369 |
| MT_MASK | 369 |
| MT_MULTI (reserved) | 368 |
| MT_PROGRAM | 368 |
| MT_SUBROUT | 368 |
| MT_SYSTEM | 369 |
| MT_TRAPLIB | 369 |

available values for SCF
| MT_ANY | 161 |
| MT_CDBDATA | 161 |
| MT_DATA | 161 |
| MT_DEVDESC | 162 |
| MT_DEVDRVR | 162 |

m_tylan (type/language)
changing
for cnfgdata | 27 |
for init | 99 |
for PCF | 462 |
for pipe | 518 |
for RBF | 406 |
for SBF | 368 |
for SCF | 161 |

m_user
changing
for cnfgdata | 23 |
for init | 95 |
for PCF | 458 |
for pipe | 514 |
for RBF | 402 |
for SBF | 364 |
for SCF | 157 |

MA_GHOST
module attribute for cnfgdata
ghost | 29 |
module attribute for init
ghost | 101 |
module attribute for PCF
ghost | 464 |
module attribute for pipe
ghost | 520 |
module attribute for RBF
ghost | 408 |
module attribute for SBF
ghost | 370 |
module attribute for SCF
ghost | 163 |

MA_MASK | 371 |

MA_REENT
module attribute for cnfgdata
re-entrant | 29 |
module attribute for init
re-entrant | 101 |
module attribute for PCF
re-entrant | 464 |
module attribute for pipe
re-entrant | 520 |
module attribute for RBF
re-entrant | 408 |
module attribute for SBF
re-entrant 370
module attribute for SCF
re-entrant 163
MA_SUPER
module attribute for cnfgdata
system-state 30
module attribute for init
system-state 102
module attribute for PCF
system-state 465
module attribute for pipe
system-state 521
module attribute for RBF
system-state 409
module attribute for SBF
system-state 371
module attribute for SCF
system-state 164
mac_address
changing for cnfgdata 69
interface data
configuration 69
mask
module header
type code for cnfgdata 28
type code for init 100
type code for PCF 463
type code for pipe 519
type code for RBF 407
type code for SBF 369
type code for SCF 162
MAX_NOTIFIERS
changing 86
max_notifiers
changing for cnfgdata 86
MAXBUFF
changing for SCF 187
maxlpmconns
changing for cnfgdata 62
maxlpmprotos
changing for cnfgdata 60
MAXPTY
changing for init 126
maxrcvmbufs
changing for cnfgdata 61
MAXSIGS
changing
for init 135
MEM_SHARED 138
MEM_SYS 138
memory transfer size
changing 497
memory.h 138
MH_ACCESS
changing
for cnfgdata 25
for init 97
for PCF 460
for pipe 516
for RBF 404
for SBF 366
for SCF 159
MH_ATTREV
changing
for cnfgdata 29
for init 101
for PCF 464
for pipe 520
for RBF 408
for SBF 370
for SCF 163
MH_EDITION
changing
for cnfgdata 31
for init 103
for PCF 466
for pipe 522
for RBF 410
for SBF 372
for SCF 165
MH_GROUP
changing
for cnfgdata 22
for init 94
for PCF 457
for pipe 513
for RBF 401
for SBF 363
for SCF 156
MH_NAME
changing
for cnfgdata 24
for init 96
for PCF 459
for pipe 515
for RBF 403
for SBF 365
for SCF 158
MH_TYLAN
changing for cnfgdata 27
   for init 99
   for PCF 462
   for pipe 518
   for RBF 406
   for SBF 368
   for SCF 161

MH_USER
   changing for cnfgdata 23
   for init 95
   for PCF 458
   for pipe 514
   for RBF 402
   for SBF 364
   for SCF 157

MINPTY
   changing for init 125

ML_ANY
   language code for SCF wildcard value 162

ML_CBLCODE
   language code
      COBOL I-code (reserved) 28, 100
      language code for PCF
         COBOL I-code (reserved) 463
      language code for pipe
         COBOL I-code (reserved) 519
      language code for RBF
         COBOL I-code (reserved) 407
      language code for SBF
         COBOL I-code (reserved) 369
      language code for SCF
         COBOL I-code (reserved) 162

ML_CCODE
   language code
      C I-code (reserved) 28, 100
      language code for PCF
         C I-code (reserved) 463
      language code for pipe
         C I-code (reserved) 519
      language code for RBF
         C I-code (reserved) 407
      language code for SBF
         C I-code (reserved) 369
      language code for SCF
         C I-code (reserved) 162

ML_FRTNCODE
   language code
      Fortran 28, 100
      language code for PCF

ML_ICODE
   language code for cnfgdata intermediate code 28
   language code for init intermediate code 100
   language code for PCF intermediate code 463
   language code for pipe intermediate code 519
   language code for RBF intermediate code 407
   language code for SBF intermediate code 369
   language code for SCF intermediate code 162

ML_MASK
   language code
      mask 28, 100
      language code for PCF
         mask 463
      language code for pipe
         mask 519
      language code for RBF
         mask 407
      language code for SBF
         mask 369
      language code for SCF
         mask 162

ML_OBJECT
   language code for cnfgdata machine language 28
   language code for init machine language 100
   language code for PCF machine language 463
   language code for pipe machine language 519
   language code for RBF machine language 407
   language code for SBF machine language 369
   language code for SCF machine language 162

ML_PCODE
language code for cnfgdata
  PASCAL 28
language code for init
  PASCAL 100
language code for PCF
  PASCAL 463
language code for pipe
  PASCAL 519
language code for RBF
  PASCAL 407
language code for SBF
  PASCAL 369
language code for SCF
  PASCAL 162
mod_name
  changing
    for cnfgdata 24
    for init 96
    PCF descriptor name 459
    pipe descriptor name 515
    RBF descriptor name 403
    SBF descriptor name 365
    SCF descriptor name 158
modes.h
  setting dd_mode 172, 379, 417, 473, 529
module
  header 92
module header
  attributes/revision
    setting for cnfgdata 29
    setting for init 101
    setting for PCF 464
    setting for pipe 520
    setting for RBF 408
    setting for SBF 370
    setting for SCF 163
cnfgdata
  access permissions
    setting 25
edition number
  setting for cnfgdata 31
  setting for init 103
  setting for PCF 466
  setting for pipe 522
  setting for RBF 410
  setting for SBF 372
  setting for SCF 165
init
  access permissions
    setting 97
PCF
  access permissions
    setting 460
pipe
  access permissions
    setting 516
RBF
  access permissions
    setting 404
SBF
  access permissions
    setting 366
SCF
  access permissions
    setting 159
type and language
  setting for cnfgdata 27
  setting for init 99
  setting for PCF 462
  setting for pipe 518
  setting for RBF 406
  setting for SBF 368
  setting for SCF 161
user ID
  cnfgdata 23
  init 95
  PCF 458
  pipe 514
  RBF 402
  SBF 364
  SCF 157
module.h
  access permissions
    available values for cnfgdata 25, 97
    available values for PCF 460
    available values for pipe 516
    available values for RBF 404
    available values for SBF 366
    available values for SCF 159
attribute/revision
    available values for cnfgdata 29
    available values for init 101
    available values for PCF 464
    available values for pipe 520
    available values for RBF 408
    available values for SBF 370
    available values for SCF 163
type/language
    available values for cnfgdata 27
    available values for init 99
    available values for PCF 462
    available values for pipe 518
    available values for RBF 406
    available values for SBF 368
    available values for SCF 161

MP_GROUP_EXEC
setting module permissions
  for cnfgdata 25
  for init 97
  for PCF 460
  for pipe 516
  for RBF 404
  for SBF 366
  for SCF 159
MP_GROUP_MASK
  setting module permissions
  for cnfgdata 25
  for init 98
  for PCF 461
  for pipe 517
  for RBF 405
  for SBF 367
  for SCF 160
MP_GROUP_READ
  setting module permissions
  for cnfgdata 25
  for init 97
  for PCF 460
  for pipe 516
  for RBF 404
  for SBF 366
  for SCF 159
MP_GROUP_WRITE
  setting module permissions
  for cnfgdata 25
  for init 97
  for PCF 460
  for pipe 516
  for RBF 404
  for SBF 366
  for SCF 159
MP_OWNER_EXEC
  setting module permissions
  for cnfgdata 25
  for init 97
  for PCF 460
  for pipe 516
  for RBF 404
  for SBF 366
  for SCF 159
MP_OWNER_MASK
  setting module permissions
  for cnfgdata 25
  for init 97
  for PCF 460
  for pipe 516
  for RBF 404
  for SBF 366
  for SCF 159
MP_OWNER_READ
  setting module permissions
  for cnfgdata 26
  for init 97
  for PCF 460
  for pipe 516
  for RBF 405
  for SBF 367
  for SCF 159
MP_OWNER_WRITE
  setting module permissions
  for cnfgdata 26
  for init 98
  for PCF 461
  for pipe 517
  for RBF 405
  for SBF 367
  for SCF 160
MP_SYSTM_ACCESS
  setting module permissions
  for cnfgdata 26
  for init 98
  for PCF 461
  for pipe 517
  for RBF 405
  for SBF 367
  for SCF 160
MP_WORL_ACCESS
  setting module permissions
  for cnfgdata 26
  for init 97
  for PCF 460
  for pipe 516
  for RBF 404
  for SBF 366
  for SCF 159
MP_WORL.Exec
  setting module permissions
  for cnfgdata 26
  for init 97
  for PCF 460
  for pipe 516
  for RBF 404
  for SBF 366
  for SCF 159
MP_WORL_MASK
  setting module permissions
  for cnfgdata 26
  for init 98
  for PCF 461
  for pipe 517
  for RBF 405
  for SBF 367
for SCF 160
MP_WORLD_READ
  setting module permissions
  for cnfgdata 26
  for init 98
  for PCF 461
  for pipe 517
  for RBF 405
  for SBF 367
  for SCF 160
MP_WORLD_WRITE
  setting module permissions
  for cnfgdata 26
  for init 98
  for PCF 461
  for pipe 517
  for RBF 405
  for SBF 367
  for SCF 160
MPUCHIP
  changing
  for init 107
MR_MASK 371
MT_ANY
  language code for cnfgdata
    wildcard value 28
  language code for init
    wildcard value 100
  language code for PCF
    wildcard value 463
  language code for pipe
    wildcard value 519
  language code for RBF
    wildcard value 407
  language code for SBF
    wildcard value 369
  m_tylan field for cnfgdata
    wildcard value 27
  m_tylan field for init
    wildcard value 99
  m_tylan field for PCF
    wildcard value 462
  m_tylan field for pipe
    wildcard value 518
  m_tylan field for RBF
    wildcard value 406
  m_tylan field for SBF
    wildcard value 368
  m_tylan field for SCF
    wildcard value 161
MT_DATA
  m_tylan field for cnfgdata
    data value 27
  m_tylan field for init
    data value 99
  m_tylan field for PCF
    data value 462
  m_tylan field for pipe
    data value 518
  m_tylan field for RBF
    data value 406
  m_tylan field for SBF
    data value 368
  m_tylan field for SCF
    data value 161
MT_DEVDESC
  m_tylan field for cnfgdata
    device descriptor value 28
  m_tylan field for init
    device descriptor value 100
  m_tylan field for PCF
    device descriptor value 463
  m_tylan field for pipe
    device descriptor value 519
  m_tylan field for RBF
    device descriptor value 407
  m_tylan field for SBF
    device descriptor value 369
  m_tylan field for SCF
    device descriptor value 162
MT_DEVDRVR
  m_tylan field for cnfgdata
    physical device driver value 28
  m_tylan field for init
    physical device driver value 100
  m_tylan field for PCF
    physical device driver value 463
  m_tylan field for pipe
    physical device driver value 519
  m_tylan field for RBF
    physical device driver value 407
m_tylan field for SBF
  physical device driver value 369
m_tylan field for SCF
  physical device driver value 162

MT_FILEMAN
  m_tylan field for cnfgdata
    file manager value 28
  m_tylan field for init
    file manager value 100
  m_tylan field for PCF
    file manager value 463
  m_tylan field for pipe
    file manager value 519
  m_tylan field for RBF
    file manager value 407
  m_tylan field for SBF
    file manager value 369
  m_tylan field for SCF
    file manager value 162

MT_MASK
  m_tylan field for cnfgdata 28
  m_tylan field for init 100
  m_tylan field for PCF 463
  m_tylan field for pipe 519
  m_tylan field for RBF 407
  m_tylan field for SBF 369
  m_tylan field for SCF 162

MT_MULTI
  m_tylan field for cnfgdata
    reserved value 27
  m_tylan field for init
    reserved value 99
  m_tylan field for PCF
    reserved value 462
  m_tylan field for pipe
    reserved value 518
  m_tylan field for RBF
    reserved value 406
  m_tylan field for SBF
    reserved value 368
  m_tylan field for SCF
    reserved value 161

MT_PROGRAM
  m_tylan field for cnfgdata
    executable value 27
  m_tylan field for init
    executable value 99
  m_tylan field for PCF
    executable value 462
  m_tylan field for pipe
    executable value 518
  m_tylan field for RBF
    executable value 406

N
  name string
  device driver
selecting for PCF 476
selecting for pipe 532
selecting for RBF 420
selecting for SBF 382
selecting for SCF 175

file manager
selecting for PCF 475
selecting for pipe 531
selecting for RBF 419
selecting for SBF 381
selecting for SCF 174

O
object code
module header
language code for cnfgdata 28
language code for init 100
language code for PCF 463
language code for pipe 519
language code for RBF 407
language code for SBF 369
language code for SCF 162

OS_EDITION
changing for init 133

OS_LEVEL
changing for init 130

OS_REVISION
changing 132

OS_VERSION
changing for init 131

OS9K_REVSTR
changing for init 109

os9rev_name
changing for init 109

OUTPUT_TYPE
changing for SCF 184

OUTSIZE
changing for SCF 189

P
PAGE_SIZE
changing for SCF 190

PAGEPAUSE
changing

for SCF 352

PAGESIZE
changing for SCF 355

PARK
changing for PCF 495
for RBF 439

parking disk heads,
changing cylinder location for 439, 495

PASCAL
module header
language code for cnfgdata 28
language code for init 100
language code for PCF 463
language code for pipe 519
language code for RBF 407
language code for SBF 369
language code for SCF 162

path descriptor size
dd_pd_size 169, 376, 414, 470, 526
for IOMAN 169, 376, 414, 470, 526

PATHS
changing for init 120
pause control key
changing for keyboard 195

PCF
BLKSIZE
changing 491
changing
BLKOFFS 489
BLKSTRK 484
BLKSTRK0 485
CONTROL 492
CTRLRID 504, 505
CYLINDRS 483
dd_class field 477
dd_lun field 469
dd_mode field 473
dd_pd_size field 470
dd_port field 468
dd_type field 471
dvr_name field 476
fmgr_name field 475
FORMAT 481
INTRLV 487
IRQLEVEL 499
LSNOFFS 496
lu_ctrlrid field 504, 505
lu_lun field 503
lu_stp field 501
lu_tfm field 502
m_access field 460
m_attrev (attributes/revision) field 464
m_edit field 466
m_group field 457
m_tylan (type/language) field 462
m_user field 458
PARK 495
pd_blk field 484
pd_boffs field 489
pd_bsize field 491
pd_cntl field 492
pd_cyl field 483
pd_format field 481
pd_ivl field 487
pd_lsnoffs field 496
pd_park field 495
pd_rwr field 494
pd_sas field 486
pd_sid field 479
pd_tob field 485
pd_tofts field 488
pd_trys field 490
pd_vfy field 480
pd_wpc field 493
pd_xfersize field 497
PRECOMP 493
REDWRITE 494
SCSILUN 503
SEGSIZE 486
STEP 501
TRKOFFS 488
TRYS 490
v_irqlevel field 499
v_priority field 500
v_vector field 498
VECTOR 498
XFERSIZE 497
DD_CLASS
changing 477
DD_MODE
changing 473
DD_TYPE
changing 471
descriptor name
changing mod_name field 459
device driver
port address offset 468
DMAMODE
changing 502
DRVR_NAME
changing 476
FMGR_NAME
changing 475
LUN
changing 469
MH_ACCESS
changing 460
MH_ATTREV
changing 464
MH_EDITION
changing 466
MH_GROUP
changing 457
MH_NAME
changing 459
MH_TYLAN
changing 462
MH_USER
changing 458
module header
group ID 457
PD_SIZE
changing 470
PORTADDR
changing 468
user ID
module header 458
pd_alf
changing
for SCF 351
pd_backsp
changing
for SCF 348
pd_bellch
changing
for SCF 345
pd_blk
changing
for PCF 484
for RBF 428
pd_blksz
changing
for SBF 385
pd_boffs
changing
for PCF 489
for RBF 433
pd_bsize
changing
for PCF 491
for RBF 435
pd_bspch
changing
for SCF 346
pd_case  changing  for SCF 347
pd_cntl  changing  for PCF 492  for RBF 436
pd_cyl  changing  for PCF 483  for RBF 427
pd_delete  changing  for SCF 349
pd_dmamode  changing  for SBF 387
pd_echo  changing  for SCF 350
pd_eofch  changing  for SCF 343
pd_eorch  changing  for SCF 340
pd_flags  changing  for SBF 386
pd_format  changing  for PCF 481  for RBF 425
pd_ilv  changing  for PCF 487  for RBF 431
pd_inmap0func_code  changing  for SCF 212
pd_inmap0size  changing  for SCF 214
pd_inmap0string  changing  for SCF 215
pd_inmap0type  changing  for SCF 211
pd_inmap10func_code  changing  for SCF 253
pd_inmap10size  changing  for SCF 254
pd_inmap10string  changing  for SCF 255
pd_inmap10type  changing  for SCF 252
pd_inmap11func_code  changing  for SCF 257
pd_inmap11size  changing  for SCF 258
pd_inmap11string  changing  for SCF 259
pd_inmap11type  changing  for SCF 256
pd_inmap12func_code  changing  for SCF 261
pd_inmap12size  changing  for SCF 262
pd_inmap12string  changing  for SCF 263
pd_inmap12type  changing  for SCF 260
pd_inmap13func_code  changing  for SCF 265
pd_inmap13size  changing  for SCF 266
pd_inmap13string  changing  for SCF 267
pd_inmap13type  changing  for SCF 264
pd_inmap14func_code  changing  for SCF 269
pd_inmap14size  changing  for SCF 270
pd_inmap14string  changing  for SCF 271
pd_inmap14type
changing
for SCF 268

pd_inmap15func_code
changing
for SCF 273

pd_inmap15size
changing
for SCF 274

pd_inmap15string
changing
for SCF 275

pd_inmap15type
changing
for SCF 272

pd_inmap16func_code
changing
for SCF 277

pd_inmap16size
changing
for SCF 278

pd_inmap16string
changing
for SCF 279

pd_inmap16type
changing
for SCF 276

pd_inmap17func_code
changing
for SCF 281

pd_inmap17size
changing
for SCF 282

pd_inmap17string
changing
for SCF 283

pd_inmap17type
changing
for SCF 280

pd_inmap18func_code
changing
for SCF 285

pd_inmap18size
changing
for SCF 286

pd_inmap18string
changing
for SCF 287

pd_inmap18type
changing
for SCF 284

pd_inmap19func_code
changing
for SCF 289

pd_inmap19size
changing
for SCF 290

pd_inmap19string
changing
for SCF 291

pd_inmap19type
changing
for SCF 288

pd_inmap1func_code
changing
for SCF 217

pd_inmap1size
changing
for SCF 218

pd_inmap1string
changing
for SCF 219

pd_inmap1type
changing
for SCF 216

pd_inmap20func_code
changing
for SCF 293

pd_inmap20size
changing
for SCF 294

pd_inmap20string
changing
for SCF 295

pd_inmap20type
changing
for SCF 292

pd_inmap21func_code
changing
for SCF 297

pd_inmap21size
changing
for SCF 298

pd_inmap21string
changing
for SCF 299

pd_inmap21type
changing
for SCF 296

pd_inmap22func_code
changing
for SCF 301

pd_inmap22size
changing
for SCF 302

pd_inmap22string
changing for SCF 303
pd_inmap22type changing for SCF 300
pd_inmap23func_code changing for SCF 305
pd_inmap23size changing for SCF 306
pd_inmap23string changing for SCF 307
pd_inmap23type changing for SCF 304
pd_inmap24func_code changing for SCF 309
pd_inmap24size changing for SCF 310
pd_inmap24string changing for SCF 311
pd_inmap24type changing for SCF 308
pd_inmap25func_code changing for SCF 313
pd_inmap25size changing for SCF 314
pd_inmap25string changing for SCF 315
pd_inmap25type changing for SCF 312
pd_inmap26func_code changing for SCF 317
pd_inmap26size changing for SCF 318
pd_inmap26string changing for SCF 319
pd_inmap26type changing for SCF 316

pd_inmap27func_code changing for SCF 321
pd_inmap27size changing for SCF 322
pd_inmap27string changing for SCF 323
pd_inmap27type changing for SCF 320
pd_inmap28func_code changing for SCF 325
pd_inmap28size changing for SCF 326
pd_inmap28string changing for SCF 327
pd_inmap28type changing for SCF 324
pd_inmap29func_code changing for SCF 329
pd_inmap29size changing for SCF 330
pd_inmap29string changing for SCF 331
pd_inmap29type changing for SCF 328
pd_inmap30func_code changing for SCF 221
pd_inmap30size changing for SCF 222
pd_inmap30string changing for SCF 223
pd_inmap2type changing for SCF 220
pd_inmap30func_code changing for SCF 333
pd_inmap30size changing
pd_inmap30string changing for SCF 334
pd_inmap30type changing for SCF 335
pd_inmap31func_code changing for SCF 337
pd_inmap31size changing for SCF 338
pd_inmap31string changing for SCF 339
pd_inmap31type changing for SCF 336
pd_inmap3func_code changing for SCF 225
pd_inmap3size changing for SCF 226
pd_inmap3string changing for SCF 227
pd_inmap3type changing for SCF 224
pd_inmap4func_code changing for SCF 229
pd_inmap4size changing for SCF 230
pd_inmap4string changing for SCF 231
pd_inmap4type changing for SCF 228
pd_inmap5func_code changing for SCF 233
pd_inmap5size changing for SCF 234
pd_inmap5string changing for SCF 235, 239
pd_inmap5type changing for SCF 228
pd_inmap6func_code changing for SCF 237
pd_inmap6size changing for SCF 238
pd_inmap6type changing for SCF 236
pd_inmap7func_code changing for SCF 241
pd_inmap7size changing for SCF 242
pd_inmap7string changing for SCF 243
pd_inmap7type changing for SCF 240
pd_inmap8func_code changing for SCF 245
pd_inmap8size changing for SCF 246
pd_inmap8string changing for SCF 247
pd_inmap8type changing for SCF 244
pd_inmap9func_code changing for SCF 249
pd_inmap9size changing for SCF 250
pd_inmap9string changing for SCF 251
pd_inmap9type changing for SCF 248
pd_insm changing for SCF 353
pd_lsnoffs changing for PCF 496
pd_nulls
   changing
      for SCF 354

pd_page
   changing
      for SCF 355

pd_park
   changing
      PCF 495
      RBF 439

pd_pause
   changing
      for SCF 352

pd_rwr
   changing
      PCF 494
      RBF 438

pd_sas
   changing
      PCF 486
      RBF 430

pd_sci_id
   changing
      SBF 388

pd_scsilun
   changing
      SBF 389

pd_sid
   changing
      for PCF 479
      for RBF 423

PD_SIZE
   changing
      for PCF 470
      for pipe 526
      for RBF 414
      for SBF 376
      for SCF 169

pd_t0b
   changing
      PCF 485
      RBF 429

pd_tabch
   changing
      for SCF 344

pd_tabsiz
   changing
      for SCF 356

pd_toffs
   changing
      PCF 488
      RBF 432

pd_trys
   changing
      PCF 490
      RBF 434

pd_vfy
   changing
      for PCF 480
      for RBF 424

pd_wpc
   changing
      for PCF 493
      for RBF 437

pd_xfersize
   changing
      for PCF 497
      for RBF 441

pipe
   changing
      BUFSZ 535
      bufsz field 535
      dd_class field 533
      dd_lun field 525
      dd_mode field 529
      dd_pd_size field 526
      dd_port field 524
      dd_type field 527
      drvr_name field 532
      fmgr_name field 531
      m_access field 516
      m_attrev (attributes/revision) field 520
      m_edit field 522
      m_group field 513
      m_tylan (type/language) field 518
      m_user field 514

DD_CLASS
   changing 533

DD_MODE
   changing 529

DD_TYPE
   changing 527
descriptor name
   changing mod_name field 515
device driver
   port address offset 524

DRVR_NAME
   changing 532

FMGR_NAME
   changing 531

LUN
   changing 525

MH_ACCESS
   changing 516

MH_ATTREV
changing 520
MH_EDITION
  changing 522
MH_GROUP
  changing 513
MH_NAME
  changing 515
MH_TYLAN
  changing 518
MH_USER
  changing 514
  module header
    group ID 513
PD_SIZE
  changing 526
PORTADDR
  changing 524
  user ID
    module header 514
polling interrupt
  changing 392, 444, 500
port address
  changing, for
    PCF device 468
    pipe device 524
    RBF device 412
    SBF device 374
    SCF device 167
port_address
  changing
    for cnfgdata 73
interface data
  configuration 73
PORTADDR
  changing
    for PCF 468
    for pipe 524
    for RBF 412
    for SBF 374
    for SCF 167
PRECOMP
  changing
    for PCF 493
    for RBF 437
precompensation
  starting point for writing
    changing 437, 493
preio_name
  changing
    for init 136
PREIOS
  changing
    for init 136
prior
  changing
    for init 139
PRIORITY
  changing 392, 444, 500
    for SCF 182
PROCS
  changing
    for init 119
Q
  quit control key
    changing for keyboard 194
R
RBF
  BLKSIZE
    changing 435
    changing
      BLKOFFS 433
      BLKSTRK 428
      BLKSTRK0 429
      CONTROL 436
      CTRLRID 448
      CYLNDRS 427
      dd_class field 421
      dd_lun field 413
      dd_mode field 417
      dd_pd_size field 414
      dd_port field 412
      dd_type field 415
      drv_name field 420
      fmgr_name field 419
      FORMAT 425
      INTRLV 431
      IRQLEVEL 443
      LSNOFFS 440
      lu_ctrlrid field 448
      lu_lun field 447
      lu_stp field 445
      lu_tfm field 446
      lu_totcyls field 449
      m_access field 404
      m_attrev (attributes/revision) field 408
      m_edit field 410
      m_group field 401
      m_tylan (type/language) field 406
      m_user field 402
      PARK 439
      pd_blk field 428
      pd_boffs field 433
      pd_bsize field 435
pd_cntl field 436
pd_cyl field 427
pd_format field 425
pd_lv field 431
pd_lsnoffs field 440
pd_park field 439
pd_rwr field 438
pd_sas field 430
pd_sid field 423
pd_tob field 429
pd_tooffs field 432
pdtrys field 434
pd_vfy field 424
pd_wpc field 437
pd_xfersize field 441
PRECOMP 437
REDWRITE 438
SCSILUN 447
SEGSIZE 430
STEP 445
TOTCYLS 449
TRKOFFS 432
TRYS 434
v_irqlevel field 443
v_priority field 444
v_vector field 442
VECTORSIZE 441
DD_CLASS
  changing 421
DD_MODE
  changing 417
DD_TYPE
  changing 415
descriptor name
  changing mod_name field 403
device driver
  port address offset 412
DMAMODE
  changing 446
DRVR_NAME
  changing 420
FMGR_NAME
  changing 419
LUN
  changing 413
MH_ACCESS
  changing 404
MH_ATTREV
  changing 408
MH_EDITION
  changing 410
MH_GROUP
  changing 410
MH_NAME
  changing 403
MH_TYLAN
  changing 406
MH_USER
  changing 402
module header
group ID 401
PD_SIZE
  changing 414
PORTADDR
  changing 412
user ID
  module header 402
reduced write cylinder
  starting point
  changing 438, 494
REDWRITE
  changing
    for PCF 494
    for RBF 438
re-entrant
  module
    setting value for cnfgdata 29
    setting value for init 101
    setting value for PCF 464
    setting value for pipe 520
    setting value for RBF 408
    setting value for SBF 370
    setting value for SCF 163
request to send flag
  changing for SCF device 203
revision
  of module
    setting for cnfgdata 29
    setting for init 101
    setting for PCF 464
    setting for pipe 520
    setting for RBF 408
    setting for SBF 370
    setting for SCF 163
RTC_NAME
  changing
    for init 116
rtc_name
  changing
    for init 116
RTSSTATE
  changing
    for SCF 203
S
S_IAPPEND
dd_mode
  available value 173, 380, 418, 474, 530
S_ICONTIG
dd_mode
  available value 173, 380, 417, 473, 529
S_ICREAT
dd_mode
  available value 172, 379, 417, 473, 529
S_IEXEC
dd_mode
  available value 173, 380, 418, 474, 530
S_IGEXEC
dd_mode
  available value 172, 379, 417, 473, 529
S_IGWRITE
dd_mode
  available value 173, 380, 418, 474, 530
S_IOEXEC
dd_mode
  available value 173, 380, 418, 474, 530
S_IOWRITE
dd_mode
  available value 173, 380, 418, 474, 530
S_IEXEC
dd_mode
  available value 173, 380, 418, 474, 530
S_IlPORT
dd_mode
  available value 173, 380, 418, 474, 530
S_IREAD
dd_mode
  available value 173, 380, 418, 474, 530
S_IWRITE
dd_mode
  available value 173, 380, 418, 474, 530
SBF
  changing
dd_class field 383
dd_lun field 375
dd_mode field 379
dd_pd_size field 376
dd_port field 374
dd_type field 377
DMAMODE 387
drvr_name field 382
FLAGS 386
fmrngr_name field 381
IRQLEVEL 391
m_access field 366
m_attrev (attributes/revision) field 370
m_edit field 372
m_group field 363
m_tylan (type/language) field 368
m_user field 364
pd_blksz field 385
pd_dmamode field 387
pd_flags field 386
pd_scilun field 389
sbf_dflag field 393
sbf_irqlevel field 391
sbf_priority field 392
sbf_vector field 390
SCSID 388
SCSiLUN 389
VECT 390
DD_CLASS
  changing 383
DD_MODE
  changing 379
DD_TYPE
  changing 377
descriptor name
  changing mod_name field 365
device driver
  port address offset 374
DRVR_NAME
  changing 382
FMGR_NAME
  changing 381
LUN
  changing 375
MH_ACCESS
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>pd_inmap18func_code field</td>
<td>285</td>
</tr>
<tr>
<td>pd_inmap18size field</td>
<td>286</td>
</tr>
<tr>
<td>pd_inmap18string field</td>
<td>287</td>
</tr>
<tr>
<td>pd_inmap18type field</td>
<td>288</td>
</tr>
<tr>
<td>pd_inmap19func_code field</td>
<td>289</td>
</tr>
<tr>
<td>pd_inmap19size field</td>
<td>290</td>
</tr>
<tr>
<td>pd_inmap19string field</td>
<td>291</td>
</tr>
<tr>
<td>pd_inmap19type field</td>
<td>292</td>
</tr>
<tr>
<td>pd_inmap20func_code field</td>
<td>293</td>
</tr>
<tr>
<td>pd_inmap20size field</td>
<td>294</td>
</tr>
<tr>
<td>pd_inmap20string field</td>
<td>295</td>
</tr>
<tr>
<td>pd_inmap20type field</td>
<td>296</td>
</tr>
<tr>
<td>pd_inmap21func_code field</td>
<td>297</td>
</tr>
<tr>
<td>pd_inmap21size field</td>
<td>298</td>
</tr>
<tr>
<td>pd_inmap21string field</td>
<td>299</td>
</tr>
<tr>
<td>pd_inmap21type field</td>
<td>300</td>
</tr>
<tr>
<td>pd_inmap22func_code field</td>
<td>301</td>
</tr>
<tr>
<td>pd_inmap22size field</td>
<td>302</td>
</tr>
<tr>
<td>pd_inmap22string field</td>
<td>303</td>
</tr>
<tr>
<td>pd_inmap22type field</td>
<td>304</td>
</tr>
<tr>
<td>pd_inmap23func_code field</td>
<td>305</td>
</tr>
<tr>
<td>pd_inmap23size field</td>
<td>306</td>
</tr>
<tr>
<td>pd_inmap23string field</td>
<td>307</td>
</tr>
<tr>
<td>pd_inmap23type field</td>
<td>308</td>
</tr>
<tr>
<td>pd_inmap24func_code field</td>
<td>309</td>
</tr>
<tr>
<td>pd_inmap24size field</td>
<td>310</td>
</tr>
<tr>
<td>pd_inmap24string field</td>
<td>311</td>
</tr>
<tr>
<td>pd_inmap24type field</td>
<td>312</td>
</tr>
<tr>
<td>pd_inmap25func_code field</td>
<td>313</td>
</tr>
<tr>
<td>pd_inmap25size field</td>
<td>314</td>
</tr>
<tr>
<td>pd_inmap25string field</td>
<td>315</td>
</tr>
<tr>
<td>pd_inmap25type field</td>
<td>316</td>
</tr>
<tr>
<td>pd_inmap26func_code field</td>
<td>317</td>
</tr>
<tr>
<td>pd_inmap26size field</td>
<td>318</td>
</tr>
<tr>
<td>pd_inmap26string field</td>
<td>319</td>
</tr>
<tr>
<td>pd_inmap26type field</td>
<td>320</td>
</tr>
<tr>
<td>pd_inmap27func_code field</td>
<td>321</td>
</tr>
<tr>
<td>pd_inmap27size field</td>
<td>322</td>
</tr>
<tr>
<td>pd_inmap27string field</td>
<td>323</td>
</tr>
<tr>
<td>pd_inmap27type field</td>
<td>324</td>
</tr>
<tr>
<td>pd_inmap28func_code field</td>
<td>325</td>
</tr>
<tr>
<td>pd_inmap28size field</td>
<td>326</td>
</tr>
<tr>
<td>pd_inmap28string field</td>
<td>327</td>
</tr>
<tr>
<td>pd_inmap28type field</td>
<td>328</td>
</tr>
<tr>
<td>pd_inmap29func_code field</td>
<td>329</td>
</tr>
<tr>
<td>pd_inmap29size field</td>
<td>330</td>
</tr>
<tr>
<td>pd_inmap29string field</td>
<td>331</td>
</tr>
<tr>
<td>pd_inmap29type field</td>
<td>332</td>
</tr>
<tr>
<td>pd_inmap2func_code field</td>
<td>221</td>
</tr>
<tr>
<td>pd_inmap2size field</td>
<td>222</td>
</tr>
<tr>
<td>pd_inmap2string field</td>
<td>223</td>
</tr>
<tr>
<td>pd_inmap2type field</td>
<td>224</td>
</tr>
<tr>
<td>pd_inmap30func_code field</td>
<td>233</td>
</tr>
<tr>
<td>pd_inmap30size field</td>
<td>234</td>
</tr>
<tr>
<td>pd_inmap30string field</td>
<td>235</td>
</tr>
<tr>
<td>pd_inmap30type field</td>
<td>236</td>
</tr>
<tr>
<td>pd_inmap31func_code field</td>
<td>237</td>
</tr>
<tr>
<td>pd_inmap31size field</td>
<td>238</td>
</tr>
<tr>
<td>pd_inmap31string field</td>
<td>239</td>
</tr>
<tr>
<td>pd_inmap31type field</td>
<td>240</td>
</tr>
<tr>
<td>pd_inmap4func_code field</td>
<td>229</td>
</tr>
<tr>
<td>pd_inmap4size field</td>
<td>230</td>
</tr>
<tr>
<td>pd_inmap4string field</td>
<td>231</td>
</tr>
<tr>
<td>pd_inmap4type field</td>
<td>228</td>
</tr>
<tr>
<td>pd_inmap5func_code field</td>
<td>233</td>
</tr>
<tr>
<td>pd_inmap5size field</td>
<td>234</td>
</tr>
<tr>
<td>pd_inmap5string field</td>
<td>235</td>
</tr>
<tr>
<td>pd_inmap5type field</td>
<td>241</td>
</tr>
<tr>
<td>pd_inmap6func_code field</td>
<td>237</td>
</tr>
<tr>
<td>pd_inmap6size field</td>
<td>238</td>
</tr>
<tr>
<td>pd_inmap6type field</td>
<td>236</td>
</tr>
<tr>
<td>pd_inmap7func_code field</td>
<td>241</td>
</tr>
<tr>
<td>pd_inmap7size field</td>
<td>242</td>
</tr>
<tr>
<td>pd_inmap7string field</td>
<td>243</td>
</tr>
<tr>
<td>pd_inmap7type field</td>
<td>240</td>
</tr>
<tr>
<td>pd_inmap8func_code field</td>
<td>245</td>
</tr>
<tr>
<td>pd_inmap8size field</td>
<td>246</td>
</tr>
<tr>
<td>pd_inmap8string field</td>
<td>247</td>
</tr>
<tr>
<td>pd_inmap8type field</td>
<td>244</td>
</tr>
<tr>
<td>pd_inmap9func_code field</td>
<td>249</td>
</tr>
<tr>
<td>pd_inmap9size field</td>
<td>250</td>
</tr>
<tr>
<td>pd_inmap9string field</td>
<td>251</td>
</tr>
<tr>
<td>pd_inmap9type field</td>
<td>248</td>
</tr>
<tr>
<td>pd_insrm field</td>
<td>353</td>
</tr>
<tr>
<td>pd_nulls field</td>
<td>354</td>
</tr>
<tr>
<td>pd_page field</td>
<td>355</td>
</tr>
<tr>
<td>pd_pause field</td>
<td>352</td>
</tr>
<tr>
<td>pd_tabch field</td>
<td>344</td>
</tr>
<tr>
<td>pd_tabsz field</td>
<td>356</td>
</tr>
<tr>
<td>v_baud field</td>
<td>198</td>
</tr>
<tr>
<td>v_devspec field</td>
<td>204</td>
</tr>
<tr>
<td>v_insize field</td>
<td>188</td>
</tr>
<tr>
<td>v_intr field</td>
<td>191</td>
</tr>
<tr>
<td>v_irqlevel field</td>
<td>181</td>
</tr>
<tr>
<td>v_irqmask field</td>
<td>186</td>
</tr>
<tr>
<td>v_line field</td>
<td>190</td>
</tr>
<tr>
<td>v_lun field</td>
<td>185</td>
</tr>
<tr>
<td>v_maxbuff field</td>
<td>187</td>
</tr>
</tbody>
</table>
Index

Index

N A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

v_outsize field 189
v_parity field 200
v_pollin field 183
v_pollout field 184
v_priority field 182
v_psch field 195
v_quit field 194
v_rtsstate field 203
v_stopbits field 201
v_wordsize field 202
v_xoff field 197
v_xon field 196
DD_CLASS changing 176
DD_MODE changing 172
DD_TYPE changing 170
descriptor name
  changing mod_name field 158
device driver
  port address offset 167
DRV_R_NAME changing 175
EOLCH changing 343
EOLNULLS changing 354
EOCH changing 340
FMGR_NAME changing 174
FUNC0x01 changing 217
FUNC0x02 changing 221
FUNC0x03 changing 225
FUNC0x04 changing 229
FUNC0x05 changing 233
FUNC0x06 changing 237
FUNC0x07 changing 241
FUNC0x08 changing 245
FUNC0x09 changing 249
FUNC0x0a changing 253
FUNC0x0b changing 257
FUNC0x0c changing 261
FUNC0x0d changing 265
FUNC0x0e changing 269
FUNC0x0f changing 273
FUNC0x10 changing 277
FUNC0x11 changing 281
FUNC0x12 changing 285
FUNC0x13 changing 289
FUNC0x14 changing 293
FUNC0x15 changing 297
FUNC0x16 changing 301
FUNC0x17 changing 305
FUNC0x18 changing 309, 313
FUNC0x1a changing 317
FUNC0x1b changing 321
FUNC0x1c changing 325
FUNC0x1d changing 329
FUNC0x1e changing 333
FUNC0x1f changing 337
FUNC0x7f changing 212
INPUT_TYPE changing 183
INSERTMODE changing 353
INSIZE changing 188
IRQ_MASK changing 186
IRQLEVEL changing 181
KYBDINTR changing 191
TYPE0x11
changing 280

TYPE0x12
changing 284

TYPE0x13
changing 288

TYPE0x14
changing 292

TYPE0x15
changing 296

TYPE0x16
changing 300

TYPE0x17
changing 304

TYPE0x18
changing 308

TYPE0x19
changing 312

TYPE0x1a
changing 316

TYPE0x1b
changing 320

TYPE0x1c
changing 324

TYPE0x1d
changing 328

TYPE0x1e
changing 332

TYPE0x1f
changing 336

TYPE0x7f
changing 347

UPC_LICK
changing 351

user ID
module header 157

VECTOR
changing 164

WORDSIZE
changing 202

XOFF
changing 197

XON
changing 196

SCF baud rate,
changing the 198, 200

SCF request to send flag,
changing the 203

SCF stop bits,
changing the 201

SCSIID
changing for SBF 388

SCSILUN
changing 389
for PCF 503
for RBF 447

segment allocation size of
changing 430, 486

SEGSIZE
changing 430, 486

setting module permissions

configdata 25
init 97
PCF 460
pipe 516
RBF 404
SBF 366
SCF 159

shell 92

SIDES
changing 423, 479

sides
number of disk
changing 423, 479

SITE
changing
for init 106

size of
path descriptor
changing 169, 376, 414, 470, 526

SIZE0x01
changing
for SCF 218

SIZE0x02
changing
for SCF 222

SIZE0x03
changing
for SCF 226

SIZE0x04
changing
for SCF 230

SIZE0x05
changing
for SCF 234

SIZE0x06
changing
for SCF 238

SIZE0x07
changing
for SCF 242

SIZE0x08
changing
for SCF 246
SIZE0x09
changing for SCF 250

SIZE0x0a
changing for SCF 254

SIZE0x0b
changing for SCF 258

SIZE0x0c
changing for SCF 262

SIZE0x0d
changing for SCF 266

SIZE0x0e
changing for SCF 270

SIZE0x0f
changing for SCF 274

SIZE0x10
changing for SCF 278

SIZE0x11
changing for SCF 282

SIZE0x12
changing for SCF 286

SIZE0x13
changing for SCF 290

SIZE0x14
changing for SCF 294

SIZE0x15
changing for SCF 298

SIZE0x16
changing for SCF 302

SIZE0x17
changing for SCF 306

SIZE0x18
changing for SCF 310

SIZE0x19
changing for SCF 314

SIZE0x1a
changing

for SCF 318

SIZE0x1b
changing

for SCF 322

SIZE0x1c
changing

for SCF 326

SIZE0x1d
changing

for SCF 330

SIZE0x1e
changing

for SCF 334

SIZE0x1f
changing

for SCF 338

SIZE0x2f
changing

for SCF 214

SLICE
changing

for init 123

software interrupt
changing 392, 444, 500

sparam_string
changing

for init 111

startup 92

STEP
changing

for PCF 501
for RBF 445

step rate
of drive heads
changing 445, 501

sticky
module
setting value for cnfgdata 29
setting value for init 101
setting value for PCF 464
setting value for pipe 520
setting value for RBF 408
setting value for SBF 370
setting value for SCF 163

stop bits
changing for SCF device 201

STOPBITS
changing

for SCF 201

STRING0x01
changing

for SCF 201

STRING0x02

for SCF 219
changing
for SCF 223

STRING0x03
changing
for SCF 227

STRING0x04
changing
for SCF 231

STRING0x05
changing
for SCF 235

STRING0x06
changing
for SCF 239

STRING0x07
changing
for SCF 243

STRING0x08
changing
for SCF 247

STRING0x09
changing
for SCF 251

STRING0x0a
changing
for SCF 255

STRING0x0b
changing
for SCF 259

STRING0x0c
changing
for SCF 263

STRING0x0d
changing
for SCF 267

STRING0x0e
changing
for SCF 271

STRING0x0f
changing
for SCF 275

STRING0x10
changing
for SCF 279

STRING0x11
changing
for SCF 283

STRING0x12
changing
for SCF 287

STRING0x13
changing
for SCF 291

STRING0x14
changing
for SCF 295

STRING0x15
changing
for SCF 299

STRING0x16
changing
for SCF 303

STRING0x17
changing
for SCF 307

STRING0x18
changing
for SCF 311

STRING0x19
changing
for SCF 315

STRING0x1a
changing
for SCF 319

STRING0x1b
changing
for SCF 323

STRING0x1c
changing
for SCF 327

STRING0x1d
changing
for SCF 331

STRING0x1e
changing
for SCF 335

STRING0x1f
changing
for SCF 339

STRING0x7f
changing
for cnfgdata 66

subnet_mask
changing
for cnfgdata 66

interface data
configuration 66

subroutine
module header
type code for cnfgdata 27
type code for init 99
type code for PCF 462
type code for pipe 518
type code for RBF 406
type code for SBF 368
type code for SCF 161
super user only
module
  setting value for cnfgdata 30
  setting value for init 102
  setting value for PCF 465
  setting value for pipe 521
  setting value for RBF 409
  setting value for SBF 371
  setting value for SCF 164
SYS_DEVICE
  changing for init 112
SYS_PARAMS
  changing for init 111
SYS_PRIOR
  changing for init 124
SYS_START
  changing for init 110
SYS_TMZONE
  changing for init 129
sysboot file 92
sysgo 92
sysgo_name
  changing for init 110
system
time zone 129
system module
module header
  type code for cnfgdata 28
  type code for init 100
  type code for PCF 463
  type code for pipe 519
  type code for RBF 407
  type code for SBF 369
  type code for SCF 162
system-state
module
  setting value for cnfgdata 30
  setting value for init 102
  setting value for PCF 465
  setting value for pipe 521
  setting value for RBF 409
  setting value for SBF 371
  setting value for SCF 164
TABCH
  changing for SCF 344
TABSIZE
  changing for SCF 356
  Tape Device Logical Unit Number
  changing 389
TICK_NAME
  changing for init 115
TICK_SEC
  changing for init 122
ticker_name
  changing for init 115
total number of cylinders
  setting 449
TOTCYLS
  changing for RBF 449
track
  number of blocks per
  changing 428, 484
track 0
  number of blocks per
  changing 429, 485
track offset
  changing 432, 488
  transfer memory size
  changing 441
trap library
module header
  type code for cnfgdata 27
  type code for init 100
  type code for PCF 463
  type code for pipe 519
  type code for RBF 407
  type code for SBF 369
  type code for SCF 162
TRKOFFS
  changing for PCF 488
  for RBF 432
TRYS
  changing 434, 490
type
  changing for init 138
  of module
  setting for cnfgdata 27
  setting for init 99

T
  tab character
  changing 344
setting for PCF 462
setting for pipe 518
setting for RBF 406
setting for SBF 368
setting for SCF 161

TYPE0x01
changing for SCF 216

TYPE0x02
changing for SCF 220

TYPE0x03
changing for SCF 224

TYPE0x04
changing for SCF 228

TYPE0x05
changing for SCF 232

TYPE0x06
changing for SCF 236

TYPE0x07
changing for SCF 240

TYPE0x08
changing for SCF 244

TYPE0x09
changing for SCF 248

TYPE0x0a
changing for SCF 252

TYPE0x0b
changing for SCF 256

TYPE0x0c
changing for SCF 260

TYPE0x0d
changing for SCF 264

TYPE0x0e
changing for SCF 268

TYPE0x0f
changing for SCF 272

TYPE0x10
changing for SCF 276

TYPE0x11
changing for SCF 280

TYPE0x12
changing for SCF 284

TYPE0x13
changing for SCF 288

TYPE0x14
changing for SCF 292

TYPE0x15
changing for SCF 296

TYPE0x16
changing for SCF 300

TYPE0x17
changing for SCF 304

TYPE0x18
changing for SCF 308

TYPE0x19
changing for SCF 312

TYPE0x1a
changing for SCF 316

TYPE0x1b
changing for SCF 320

TYPE0x1c
changing for SCF 324

TYPE0x1d
changing for SCF 328

TYPE0x1e
changing for SCF 332

TYPE0x1f
changing for SCF 336

TYPE0x2f
changing for SCF 336

U

UPC_LOCK
changing for SCF 347
upper-case lock character
changing 347

USRACCT_NAME
changing
for init 118

utility
chd 112
chx 112

V

v_baud
changing
for SCF 198

v_devspec
changing
for SCF 204

v_insize
changing
for SCF 188

v_intr
changing
for SCF 191

v_irqlevel
changing
for PCF 499
for RBF 443
for SCF 181

v_irqmask
changing
for SCF 186

v_line
changing
for SCF 190

v_lun
changing
for SCF 185

v_maxbuff
changing
for SCF 187

v_outsize
changing
for SCF 189

v_parity
changing
for SCF 200

v_pollin
changing
for SCF 183

v_pollout field
changing
for SCF 184

v_priority
changing
for PCF 500
for RBF 444
for SCF 182

v_psch
changing
for SCF 195

v_quit
changing
for SCF 194

v_rtsstate
changing
for SCF 203

v_stopbits
changing
for SCF 201

v_vector
changing
for PCF 498
for RBF 442

v_wordsize
changing
for SCF 202

v_xoff
changing
for SCF 197

v_xon
changing
for SCF 196

VECTOR
changing
for PCF 498
for RBF 442
for SBF 390
for SCF 180

vector interrupt
changing 390, 442, 498

vector number
setting
for console device 34, 46

VERIFY
changing
for PCF 480
for RBF 424

verify
write operation
changing 424, 480

W

wildcard
module header
language code for cnfgdata 28
language code for init 100
language code for PCF 463
language code for pipe 519
language code for RBF 407
language code for SBF 369
language code for SCF 162
type code for cnfgdata 27
type code for int 99
type code for PCF 462
type code for pipe 518
type code for RBF 406
type code for SBF 368
type code for SCF 161

WORDSIZE
changing for SCF 202

wordsize
changing for SCF device 202
write precompensation
starting point
changing 437, 493

X
XFERSIZE
changing 441, 497
XOFF
changing for SCF 197
X-OFF control key
changing for keyboard 197
XON
changing for SCF 196
X-ON control key
changing for keyboard 196